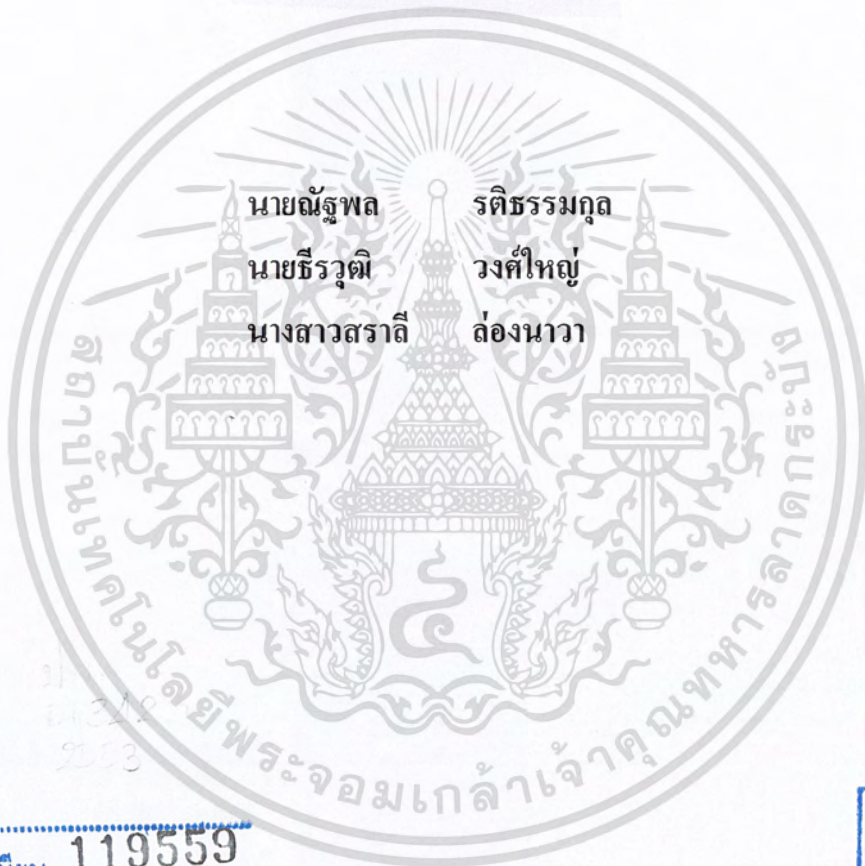


หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุล โดยใช้หลักการ อินเวอร์ทเพนดูลัม
TWO WHEELS BALANCING ROBOT BY INVERTED PENDULUM



T119559



เลขหมู่.....
เลขทะเบียน..... 119559
วัน,เดือน,ปี..... - ๘ ส.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น 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
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุค โดยใช้หลักการ อินเวอร์ทเพนดูลัม
TWO WHEELS BALANCING ROBOT BY INVERTED PENDULUM

นักศึกษาผู้จัดทำ นายณัฐพล รติธรรมกุล รหัสนักศึกษา 50010491
 นายธีรวุฒิ วงศ์ใหญ่ รหัสนักศึกษา 50010714
 นางสาวสรลติ ล่องนาวา รหัสนักศึกษา 50011648

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2553

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
ผศ.ดร.ธนิตย์ ตรีสุวรรณวัฒน์	
รศ.ประภาส อุดคกิมานันท์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุล โดยใช้หลักการ อินเวอร์ทเพนดูลัม TWO WHEELS BALANCING ROBOT BY INVERTED PENDULUM		
นักศึกษาผู้จัดทำ	นายณัฐพล	รติธรรมกุล	รหัสประจำตัว 50010491
	นายธีรวุฒิ	วงศ์ใหญ่	รหัสประจำตัว 50010714
	นางสาวสรลณี	ล่องนาวา	รหัสประจำตัว 50011648
อาจารย์ที่ปรึกษา	ผศ.ดร.ธนิตย์	ตรีสุวรรณวัฒน์	
	ร.ศ.ประภาส	อุคคกิมพันธ์	
ปีการศึกษา	2553		

บทคัดย่อ

เทคโนโลยีที่เกี่ยวกับการทรงตัวของวัตถุนั้นได้มีการคิดค้น — และพัฒนาให้ดีขึ้นอยู่ตลอดเวลาซึ่ง หลักการและทฤษฎีต่าง ๆ นั้นถูกนำมาใช้ประโยชน์มากมายไม่ว่าจะเป็นสิ่งปลูกสร้าง อาคาร บ้านเรือน, หุ่นยนต์ในอุตสาหกรรม หรืออาจเป็นยานพาหนะต่างๆ เป็นต้น ระบบอินเวอร์ทเพนดูลัม ถือว่าเป็นระบบหนึ่งที่เกี่ยวข้องกับการทรงตัวของวัตถุ ซึ่งได้ถูกนำมาประยุกต์ใช้ในรูปแบบต่างๆ มากมาย ล่าสุดได้มีการนำเอาระบบนี้มาใช้กับยานพาหนะที่เรียกว่า เสกเวย์ ซึ่งหลักการของระบบอินเวอร์ทเพนดูลัม ก็คือหลักการที่เกี่ยวกับการทรงตัว โดยการรักษาสมดุลของก้านเพนดูลัม ให้ตั้งฉากกับแกน หรือฐานของเพนดูลัม ซึ่งหลักการนี้ได้ถูกนำมาประยุกต์ใช้ร่วมกับหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุล เพื่อรักษาสมดุลของตัวรถให้ทรงตัวอยู่ได้ตลอด โดยการทรงตัวของก้านนี้จะต้องสัมพันธ์กับส่วนของการทำงานระบบขับเคลื่อนของตัวรถด้วย ซึ่งการควบคุมการทำงานของมอเตอร์นั้น จะใช้ระบบควบคุม ออปติมอลคอนโทรลแบบไม่ต่อเนื่อง และต่อเนื่องทางเวลา ผ่านตัวประมวลผลไมโครคอนโทรลเลอร์ แบบ dsPIC ซึ่งเป็นตัวไมโครคอนโทรลเลอร์ความเร็วสูง ความสัมพันธ์ทั้งหมดนี้จะต้องถูกออกแบบอย่างดีที่สุด เพื่อให้หุ่นยนต์สามารถที่จะทรงตัวอยู่ได้ แม้จะถูกรบกวนจากแรงภายนอก

Thesis Title	Two Wheels Balancing Robot by Inverted Pendulum	
Authors	Mr. Nattapon	Ratitammakul
	Mr. Theerawut	Wongyai
	Miss Saralee	Longnava
Thesis Advisors	Assoc. Prof. Dr. Thanit	Trisuwannawat
	Assoc. Prof. Prapat	Ukkakimapan
Year	2010	

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 Discrete and Continuous Time 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 even though the car is disturbed by unexpected force.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดีจาก ผศ.ดร.ธนิตย์ ตรีสุวรรณวัฒน์ และรศ.ประภาส อุคคกิม่าพันธุ์ ซึ่งเป็นอาจารย์ที่ปรึกษา และอาจารย์ที่ปรึกษาร่วมโครงการ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปการณและเครื่องมือต่างๆในการทำปริญญาบัตรนี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน อาจารย์กฤษณ์ เสมอพิทักษ์ ที่ได้คำแนะนำอันเป็นประโยชน์และให้ความช่วยเหลือตลอดมา

ขอบคุณเพื่อนๆ ทุกคนที่ให้ความสนใจ สนับสนุนอุปการณที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยตามไถ่ความคืบหน้าของโครงการอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ.....	1
1.1 กล่าวนำ.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	2
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 รายละเอียดของปริญญาโท.....	3
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง.....	4
2.1 กล่าวนำ.....	4
2.1.1 หลักการทำงานและทฤษฎีต่างๆ.....	4
2.1.2 หลักการออกแบบ.....	5
2.2 ความรู้พื้นฐานเกี่ยวกับ ดี.ซี.มอเตอร์.....	5
2.3 หลักการของระบบควบคุมที่เหมาะสมที่สุด.....	7
2.3.1 หลักการออกแบบระบบควบคุม.....	7
2.3.2 การออกแบบตัวควบคุมแอสคิวอาร์.....	10
2.3.3 การออกแบบตัวควบคุมแอสคิวอี.....	13
2.3.4 ระบบคู่.....	22
2.4 หลักการการทำงานของเซนเซอร์.....	27
2.4.1 เซนเซอร์ไจโรสโคป.....	27
2.4.2 เซนเซอร์แอกเซลเลอโรมิเตอร์.....	28
2.5 การควบคุมการหมุนของดีซีมอเตอร์.....	29

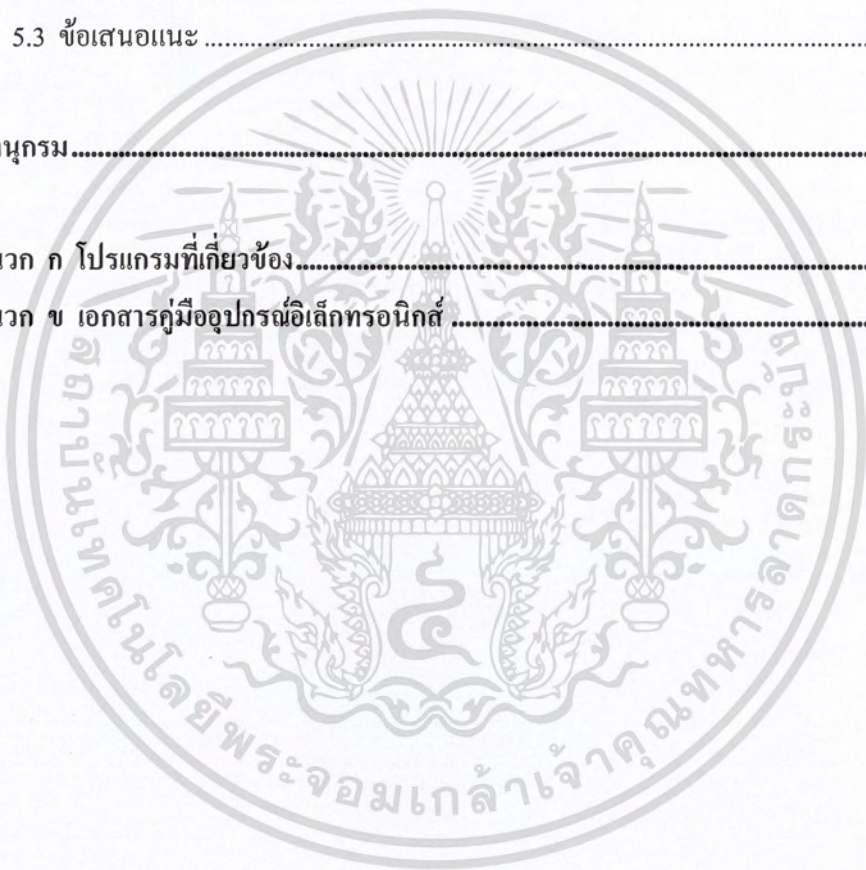
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6 การออกแบบไมโครคอนโทรลเลอร์	31
2.6.1 ข้อมูลเบื้องต้นของไมโครคอนโทรลเลอร์แบบ dsPIC.....	31
2.6.2 หน่วยกำเนิดสัญญาณพัลส์วิดท์มอดูเลตชั่น	31
2.7 การแปลงสัญญาณอนาล็อก - ดิจิตอล.....	33
2.8 พัลส์วิดท์มอดูเลตชั่น.....	35
บทที่ 3 การออกแบบระบบควบคุม ไมโครคอนโทรลเลอร์ วงจรขับมอเตอร์ แบบจำลองทาง	
คณิตศาสตร์ และ การออกแบบตัวหุ่นยนต์	37
3.1 แบบจำลองทางคณิตศาสตร์ของหุ่นยนต์.....	37
3.1.1 แบบจำลองทางคณิตศาสตร์ของดีซีมอเตอร์	37
3.1.2 พฤติกรรมของหุ่นยนต์สมดุค	39
3.1.3 หลักการของอินเวอร์ทเพนดูลัม	41
3.2 การออกแบบระบบควบคุม	44
3.2.1 การออกแบบแอสคิวอาร์	45
3.2.2 การออกแบบแอสคิวอี	46
3.3 การออกแบบโปรแกรมบนไมโครคอนโทรลเลอร์	49
3.4 การออกแบบวงจรขับมอเตอร์.....	50
3.5 การออกแบบจำลองการทำงานของหุ่นยนต์ขับเคลื่อนสองล้อ	51
3.5.1 การใช้งานโปรแกรมแลปวิว	51
3.5.2 แผงควบคุมด้านหน้า.....	52
3.5.3 บล็อกไดอะแกรม	55
3.5.4 ไอคอน และคอนเนคเตอร์	57
3.6 การออกแบบแคลมาลฟิตเตอร์	58
บทที่ 4 การทดลองและผลการทดลอง	59
4.1 การทดลองทำงานของไจโรสโคป	59
4.2 การทดลองทำงานของแอกเซลเลอโรมิเตอร์	60
4.3 การทดลองการรวมสัญญาณเซนเซอร์ด้วยแคลมาลฟิตเตอร์	61

สารบัญ (ต่อ)

	หน้า
4.4 การทดลองการจำลองการควบคุม	62
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	64
5.1 สรุปผลการทดลอง	64
5.2 ปัญหา	64
5.3 ข้อเสนอแนะ	64
บรรณานุกรม	65
ภาคผนวก ก โปรแกรมที่เกี่ยวข้อง	67
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	88



สารบัญตาราง

ตารางที่	หน้า
3.1 ค่าพารามิเตอร์ต่างๆ ของตัวหุ่นยนต์.....	44
3.2 ศัพท์ที่ใช้ในโปรแกรมแลปวิว.....	57



สารบัญรูป

รูปที่	หน้า
1.1 แสดงระบบลูกคู้มกลับหัว	1
2.1 แสดงระบบการทำงานของหุ่นยนต์ขับเคลื่อนแบบสองล้อ	4
2.2 แสดงดีซีมอเตอร์แบ่งตามลักษณะของการกระตุ้น	5
2.3 วงจรสมมูลของดีซีมอเตอร์	6
2.4 แสดงการออกแบบตัวควบคุมในแนวดั้งเดิม	7
2.5 แสดงถึงสิ่งรบกวนที่เกิดขึ้นในวงควบคุม	7
2.6 แสดงการออกแบบตัวควบคุมในแนวใหม่	8
2.7 แสดงถึงความสัมพันธ์ระหว่างตัวแปรและสัญญาณ	8
2.8 แสดงถึงลักษณะของเจเนอรัลโร้สเพลนที่	8
2.9 แสดงถึงลักษณะของเพลนที่	9
2.10 แสดงถึงลักษณะของฟังก์ชันถ่วงน้ำหนัก	9
2.11 แสดงถึงลักษณะของระบบ LQR	11
2.12 แสดงถึงลักษณะของเจเนอรัลโร้สเพลนที่	12
2.13 แสดงถึงลักษณะของเพลนที่	12
2.14 แสดงถึงลักษณะของฟังก์ชันถ่วงน้ำหนัก	12
2.15 แสดงถึงลักษณะของระบบ LQR ร่วมกับ LQE	13
2.16 แสดงถึงลักษณะของระบบ LQE	17
2.17 แสดงถึงลักษณะของระบบ LQG	18
2.18 แสดงถึงลักษณะของระบบ LQR	21
2.19 แสดงถึงลักษณะของตัวปรับแรงดัน	26
2.20 เซนเซอร์ใจโรสโคป	28
2.21 เซนเซอร์แอคเชลเลอโรมิเตอร์	28
2.22 แสดงวงจร H-Bridge Switching Transistor	29
2.23 แสดงวงจร H-Bridge Switching Transistor เมื่อ Q1 และ Q3 ทำงาน	29
2.24 แสดงวงจร H-Bridge Switching Transistor เมื่อ Q2 และ Q4 ทำงาน	30
2.25 ไมโครคอนโทรลเลอร์ dsPIC30F6010A	31
2.26 แผนภาพของโมดูล MCPWM	32
2.27 แสดงลำดับการทำงานตัวแปลงสัญญาณอนาล็อก – ดิจิตอล	34

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.28 แสดงลักษณะความกว้างของพัลส์จาก PWM.....	35
2.29 แสดงการควบคุมลักษณะพัลส์วิดท์มอดูเลชัน	36
3.1 แบบจำลองโครงสร้างของมอเตอร์	37
3.2 แผนภาพวัตถุอิสระของล้อทั้ง 2 ข้าง	39
3.3 แผนภาพวัตถุอิสระของ โครงรถส่วนที่เป็นก้านเพนดูลัม	41
3.4 หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุคโดยใช้หลักการอินเวอร์ทเพนดูลัม.....	44
3.5 ผลตอบสนองของระบบเมื่อกำหนด $q_{11} = 1$ กับ $q_{33} = 1$ และ $R = 1$	47
3.6 ผลตอบสนองเมื่อทำการเปลี่ยนแปลงค่า $q_{11} = 1000$ กับ $q_{33} = 1$ และ $R = 1$	48
3.7 ผลตอบสนองเมื่อกำหนด $q_{11} = 1000$ กับ $q_{33} = 1000$ และ $R = 1/100$	48
3.8 แผนภาพโดยรวมของการเขียน โปรแกรม	49
3.9 แสดงแผนภาพควบคุมด้านหน้าของ โปรแกรมแลปวิว	53
3.10 แสดงส่วนควบคุมของ โปรแกรมแลปวิว	54
3.11 แสดงตัวแสดงผลของ โปรแกรมแลปวิว	54
3.12 แสดงบล็อกไดอะแกรมของ โปรแกรมแลปวิว	55
3.13 แสดงถึงลักษณะของ โหนดที่บรรจุอยู่ในบล็อก ไดอะแกรมของแลปวิว.....	56
4.1 ผลการทดสอบใจโรสโคปขณะอยู่กับที่.....	60
4.2 ผลการทดสอบใจโรสโคปขณะเคลื่อนที่ไปมา	60
4.3 ผลการทดสอบแอกเซลเลอโรมิเตอร์เมื่อเซนเซอร์เอียง	61
4.4 ผลการทดสอบแอกเซลเลอโรมิเตอร์เมื่อเซนเซอร์ตั้งฉากกับพื้นโลก	61
4.5 ทดลองการรวมสัญญาณของใจโรสโคป และแอกเซลเลอโรมิเตอร์ด้วยแคลคูลัสพีดี	62
4.6 ทดลองการจำลองระบบอินเวอร์ทเพนดูลัมโดยใช้โปรแกรมแลปวิว.....	63
4.7 แสดงบล็อกไดอะแกรมของตัวหุ่นยนต์	64

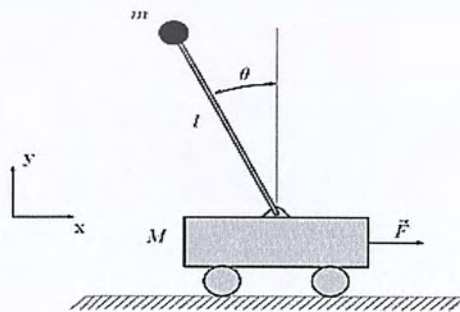
บทที่ 1

บทนำ

1.1 กล่าวนำ

หุ่นยนต์ควบคุมการทรงตัวอัตโนมัติเป็นหุ่นยนต์ที่สามารถทรงตัวได้ด้วยตัวเอง โดยใช้ล้อเพียงสองล้อ อาศัยหลักการของลูกตุ้มกลับหัวหรืออินเวิร์ทเพนดูลัม (Invert Pendulum) ซึ่งเราสามารถที่จะเอาหลักการนี้มาใช้ให้เกิดประโยชน์ได้หลากหลาย เช่น ทำเป็นสกู๊ตเตอร์ (Segway) หรือทำเป็นรถเข็นที่ทรงตัวได้เอง เพื่อใช้กับคนพิการได้ ในโครงการนี้ได้ศึกษาและทำเป็นหุ่นยนต์สองล้อที่ทรงตัวได้โดยไม่ล้ม โดยแบ่งการทำงานออกเป็นสองส่วนใหญ่ๆ คือฮาร์ดแวร์และซอฟต์แวร์โดยหัวใจหลักของโครงการนี้ ก็คือการใช้งานร่วมกันระหว่างไจโรสโคป (Gyroscope) และแอกเซลเลอโรมิเตอร์ (Accelerometer) โดยเราจะต้องนำค่าสัญญาณเอาต์พุต ของเซนเซอร์ (Sensor) ทั้ง 2 นี้มาใช้งานร่วมกันเพื่อให้เกิดการตอบสนองที่ดีของระบบตามที่เรากำลังต้องการ

ดังนั้นในโครงการนี้จึงเป็นการพัฒนาหุ่นยนต์ที่สามารถรักษาสมดุลด้วยตัวเองได้อัตโนมัติ โดยใช้ผลการจำลองเสมือน (Simulation) ของระบบควบคุม เพื่อดูแนวโน้มของตัวหุ่นยนต์ที่สร้างขึ้นว่าสามารถรักษาสมดุลด้วยตัวเองได้และสามารถควบคุมตำแหน่งของล้อทั้งสองข้าง ให้เป็นอิสระต่อกัน เพื่อให้หุ่นยนต์สามารถเคลื่อนที่ได้ซึ่งควบคุมด้วย ไมโครคอนโทรลเลอร์ dsPIC30f6010A ที่มีความเร็วในการประมวลผลสูงพอสมควร ใช้ไจโรสโคป เบอร์ ADIS16100 ทำหน้าที่ในการวัดความเร็วเชิงมุมและใช้แอกเซลเลอโรมิเตอร์เบอร์ ADXL202 ในการวัดความเร่ง ในส่วนของการออกแบบตัวควบคุมที่เหมาะสมที่สุดหรือออปติมอลคอนโทรล (Optimal Control) จะใช้โปรแกรมแมทแลป (MATLAB) ในการคำนวณหาค่าพารามิเตอร์ (Parameter) ส่วนการทำงานของหุ่นยนต์จะจำลองการทำงานในโปรแกรมแลปวิว (LabVIEW) และเขียนโปรแกรมไมโครคอนโทรลเลอร์ด้วยภาษาซี (C Language) ในโปรแกรม เอ็มพีแลป (MPLAB)



รูปที่ 1.1 แสดงระบบลูกตุ้มกลับหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาเกี่ยวกับการออกแบบควบคุม หุ่นยนต์ขับเคลื่อนสองล้อ แบบสมดุลง โดยให้หลักการอินเวอร์ทเพนดูลัม เพื่อให้หุ่นรักษาสมดุลอยู่ได้โดยไม่ล้ม
2. ออกแบบตัวควบคุมที่เหมาะสมที่สุดและนำไปควบคุมระบบ อินเวอร์ทเพนดูลัม ได้ตามวัตถุประสงค์
3. ศึกษาหลักการพื้นฐานของหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุลง โดยใช้หลักการของอินเวอร์ทเพนดูลัม
4. ศึกษาเพื่อพัฒนารักษาสมดุลและเคลื่อนที่ได้อย่างรวดเร็วของหุ่นยนต์

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะกล่าวถึงการออกแบบ ตัวควบคุมออปติมอลคอนโทรล ทั้งแบบไม่ต่อเนื่องทางเวลา (Discrete Time) และแบบต่อเนื่องทางเวลา (Continuous Time) โดยอาศัยหลักการของระบบอินเวอร์ทเพนดูลัมและทฤษฎีของมอเตอร์ (Motor) เพื่อใช้ในการพิจารณาออกแบบชุดไมโครคอนโทรลเลอร์ dsPIC (Digital Signal Processing PIC) โดยใช้ไมโครคอนโทรลเลอร์ dsPIC (Digital Signal Processing PIC) โดยใช้ไมโครคอนโทรลเลอร์ dsPIC (Digital Signal Processing PIC) โดยอาศัยหลักการของเพียโซอิเล็กทริก (Piezoelectric) มาประยุกต์ใช้เป็นตัวเซนเซอร์บอกความเอียงซึ่งจะใช้ร่วมกับแอกเซลเลอโรมิเตอร์

1.4 ขั้นตอนการศึกษาและการจัดทำโครงการ

1. ศึกษากระบวนการควบคุมอินเวอร์ทเพนดูลัม แบบจำลองทางคณิตศาสตร์และอุปกรณ์ต่างๆ ที่เกี่ยวข้องในการจัดทำโครงการ เช่น เซนเซอร์วัดความเร็วเชิงมุม เซนเซอร์วัดความเร่ง เป็นต้น
2. ศึกษารูปแบบและวงจรที่ใช้ในโครงการ
3. ศึกษาเทคนิคใหม่ๆ ที่เข้ามาช่วยปรับปรุงประสิทธิภาพของระบบ อินเวอร์ทเพนดูลัม เช่น คาลมานฟิลเตอร์ (Kalman Filter) เป็นต้น
4. สร้างรูปแบบการหาแบบจำลองทางคณิตศาสตร์
5. ออกแบบระบบควบคุมที่เหมาะสมที่สุด โดยใช้โปรแกรมเมทแลป
6. ออกแบบโครงสร้างและการเลือกใช้อุปกรณ์ โดยใช้โปรแกรมโซลิดเวิร์ค (SolidWorks)
7. ออกแบบวงจรปรับแต่งสัญญาณ เพื่อที่จะทำการเปลี่ยนแปลงระดับแรงดัน ที่ออกมาจากไมโครคอนโทรลเลอร์ ให้มีขนาดเหมาะสมกับอินพุต (Input) ของไมโครคอนโทรลเลอร์
8. เขียนโปรแกรมไมโครคอนโทรลเลอร์ ควบคุมการทำงานของหุ่นยนต์โดยใช้โปรแกรมเอ็มทีแลป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. แปลงสัญญาณอนาล็อกที่อ่านได้จาก เซนเซอร์ใจโรสโคปเป็นดิจิทัล เพื่อที่จะได้นำค่าไปใช้ในการคำนวณต่อไป

10. ประยุกต์แคลคูลัสพีดเดอร์ มาใช้รวมสัญญาณของเซนเซอร์ใจโรสโคปและเซนเซอร์ แอคเซลเลอโรมิเตอร์

11. ควบคุมความเร็วมอเตอร์โดยใช้พัลส์วidthมอดูเลชัน (Pulse Width Modulation)

12. การจำลองการทำงานของหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุคโดยใช้เลปวิว

1.5 รายละเอียดของปฏิญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ ขอบเขต ขั้นตอนการศึกษาและการจัดทำโครงการ พร้อมด้วยรายละเอียดโดยย่อของปฏิญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง ในบทนี้จะกล่าวถึง แบบจำลองทางคณิตศาสตร์ของ หุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุคโดยใช้หลักการ อินเวิร์ทเพนดูลัม ทฤษฎีการออกแบบตัวควบคุมที่เหมาะสมที่สุดทั้งแบบ ไม่ต่อเนื่องทางเวลาและแบบต่อเนื่องทางเวลา หลักการทำงานของ ไมโครคอนโทรลเลอร์และการควบคุมความเร็วดีซีมอเตอร์ (DC Motor)

บทที่ 3 หลักการออกแบบ กล่าวถึงการออกแบบ โครงสร้างของหุ่นยนต์ขับเคลื่อนสองล้อ แบบสมดุคการออกแบบวงจรและการออกแบบตัวควบคุมที่เหมาะสมที่สุด

บทที่ 4 การทดลองและผลการทดลอง เช่น การออกแบบตัวควบคุมที่เหมาะสมที่สุด การทดสอบเซนเซอร์ใจโรสโคปและแอคเซลเลอโรมิเตอร์ การออกแบบวงจร การออกแบบและเขียน โปรแกรมควบคุมไมโครคอนโทรลเลอร์ เป็นต้น

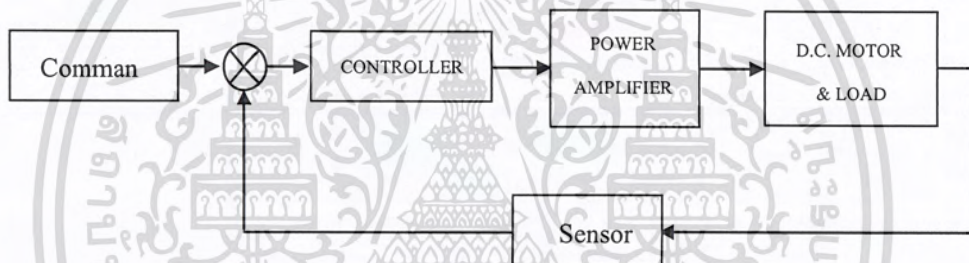
บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้นและแนวทางการปรับปรุงพัฒนาโครงการนี้ต่อไป

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

2.1 กล่าวนำ

การควบคุมการทรงตัวและการเคลื่อนที่ของหุ่นยนต์ขับเคลื่อนแบบสองล้อนั้นเป็นการรวมองค์ประกอบหลายๆองค์ประกอบเข้าด้วยกัน อันได้แก่หลักการของอินเวอร์ทเพนดูลัม หลักการของดีซีมอเตอร์ หลักการของการควบคุมควบคุมที่เหมาะสมที่สุด ทั้งแบบไม่ต่อเนื่องทางเวลาและแบบต่อเนื่องทางเวลา หลักการทำงานของเซนเซอร์ หลักการทางด้านเพาเวอร์แอมพลิไฟเออร์ (Power Amplifier) หลักการออกแบบโครงสร้างของรถ เป็นต้น ซึ่งขั้นตอนการทำงานจะแสดงดังแผนภาพต่อไปนี้



รูปที่ 2.1 แสดงระบบการทำงานของหุ่นยนต์ขับเคลื่อนแบบสองล้อ

จากรูป 2.1 แสดงให้เห็นว่าหุ่นยนต์จะทำงานได้ จำเป็นต้องรู้ถึงหลายองค์ประกอบด้วยกัน ซึ่งจะแบ่งออกเป็น 2 ส่วนใหญ่ดังนี้

2.1.1 หลักการทำงานและทฤษฎีต่างๆ ดังนี้

2.1.1.1 หลักการทำงานของดีซีมอเตอร์

- พฤติกรรมของมอเตอร์และโมเดลทางคณิตศาสตร์ของมอเตอร์

2.1.1.2 หลักการของคอนโทรลเลอร์

- หลักการของระบบควบคุมที่เหมาะสมที่สุดแบบไม่ต่อเนื่องทางเวลา

และต่อเนื่องทางเวลา

2.1.1.3 หลักการทำงานของเซนเซอร์

- ไจโรสโคปและแอกเซลเลอโรมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.4 หลักการของวงจรถิ่ชี่มอเตอรื

2.1.1.5 หลักการของไมโครคอนโทรลเลอร์

2.1.2 หลักการออกแบบ

2.1.2.1 การออกแบบโครงสร้างและการเลือกใช้อุปกรณ์

- คำนึงถึงปัจจัยต่างๆ เช่น น้ำหนัก จุดศูนย์ถ่วง ความสูง เป็นต้น
- การออกแบบโครงสร้างของหุ่นยนต์โดยใช้โปรแกรมโซลิดเวิร์ค

2.1.2.2 การออกแบบคอนโทรลเลอร์

- การออกแบบโดยใช้ตัวควบคุมที่เหมาะสมที่สุดแบบแอลคิวจี (LQG)
- การออกแบบโดยใช้โปรแกรมเมทแลป ในการคำนวณ

2.1.2.3 การออกแบบไมโครคอนโทรลเลอร์โดยใช้โปรแกรมเอ็มพีแลป

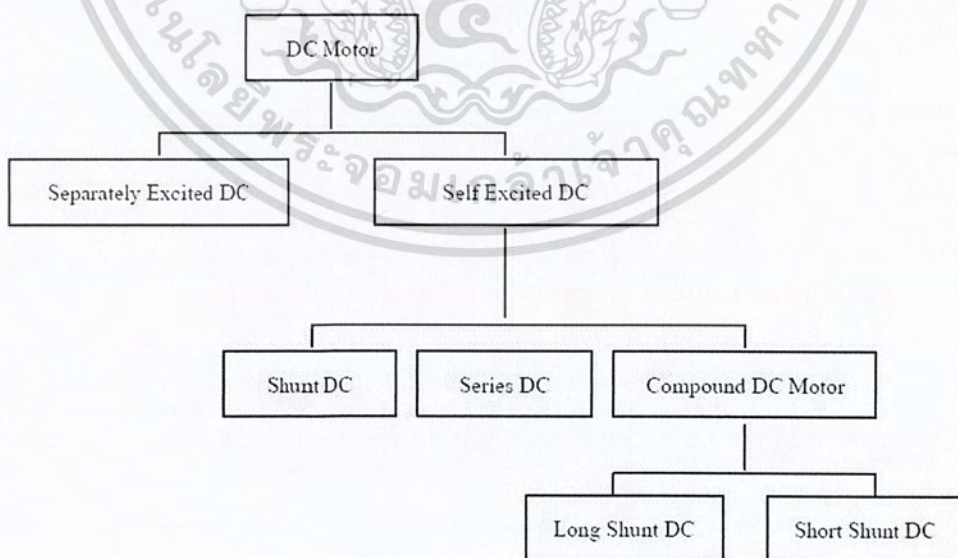
2.1.2.4 การออกแบบวงจรถิ่ชี่มอเตอรืไฟฟ้ากระแสตรง

2.1.2.5 การออกแบบตัวจำลองการทำงานโดยใช้แลปวิว

2.1.2.6 การออกแบบแคลคูลัสพีตเตอรื

2.2 ความรู้พื้นฐานเกี่ยวกับ ดิ่ชี่มอเตอรื

ดิ่ชี่มอเตอรื คือเครื่องก้าเนิดพลังงานที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล โดยแบ่งตามลักษณะของการกระตุ้น ได้ดังนี้



รูปที่ 2.2 แสดงดิ่ชี่มอเตอรืแบ่งตามลักษณะของการกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดีซีมอเตอร์ เป็นทรานสดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษ คือแรงบิดของเพลลา ของดีซีมอเตอร์ จะเป็นสัดส่วน โดยตรงกับกระแสอาร์เมเจอร์แรงบิดของเพลลาของ

ดีซีมอเตอร์ จะเป็นสัดส่วน โดยตรงกับกระแสอาร์เมเจอร์ (Armature) แรงบิดของเพลลาของ ดี.ซี.มอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการแสดงในรูปที่ 2.3 ในที่นี้ กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ (Field) ที่ประกอบด้วยเส้นแรงแม่เหล็ก ϕ และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

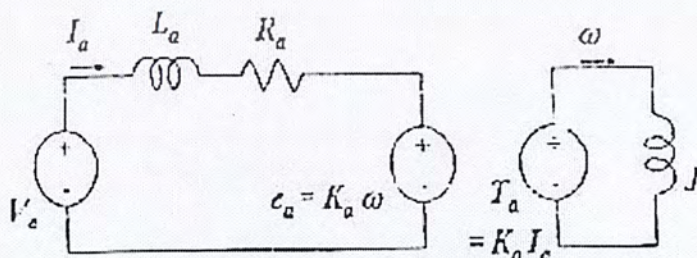
$$T = K\phi I \quad (2.1)$$

เมื่อ	T	คือ	แรงบิดของเพลลามีหน่วยเป็นนิวตัน - เมตร
	ϕ	คือ	เส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์ (Weber)
	I	คือ	กระแสเป็นแอมแปร์ (Ampere)
	K	คือ	ตัวคงที่

ดังนั้นแรงบิดของเพลลา จะเป็นสัดส่วน โดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส เมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดโวลต์เตจ (Voltage) ตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลลาและด้านกรไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลลามอเตอร์คือ

$$E = K\phi\omega \quad (2.2)$$

เมื่อ	E	คือ	โวลต์เตจย้อนกลับอีเอ็มเอฟ (Emf) มีหน่วยเป็นโวลต์
	ϕ	คือ	เส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์
	ω	คือ	ความเร็วของมอเตอร์มีหน่วยเป็นเรเดียน / วินาที



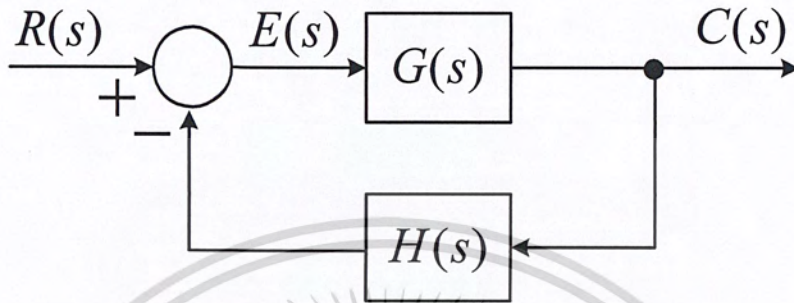
รูปที่ 2.3 วงจรสมมูลของดีซีมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการของระบบควบคุมควบคุมที่เหมาะสมที่สุด

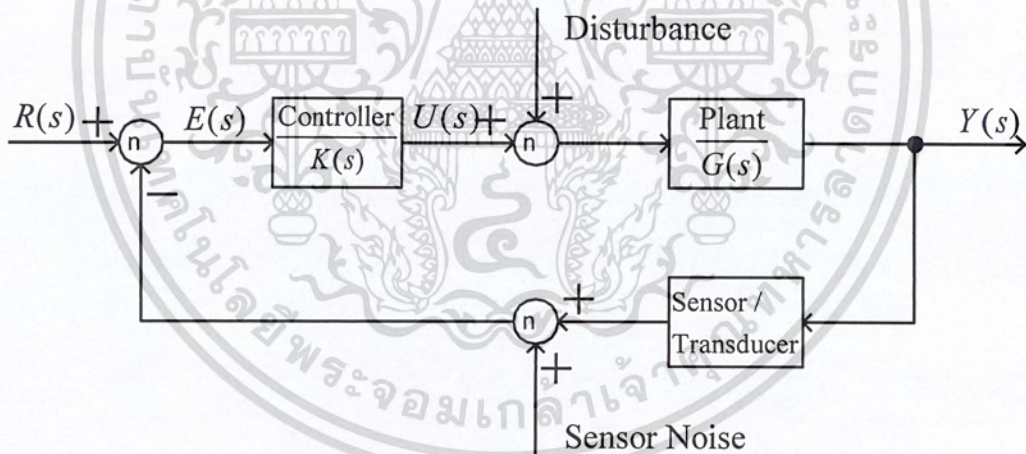
2.3.1 หลักการออกแบบระบบควบคุม

การออกแบบตัวควบคุมในแนวตั้งเดิม นั้น ถ้าเราเขียนแผนภาพกรอบง่ายๆ จะได้



รูปที่ 2.4 แสดงการออกแบบตัวควบคุมในแนวตั้งเดิม

ถ้าพิจารณาว่ามีสิ่งรบกวนในวงควบคุม เราก็อาจเขียนแผนภาพกรอบใหม่ได้เป็น

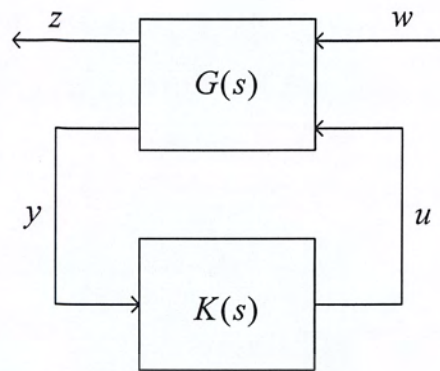


รูปที่ 2.5 แสดงถึงสิ่งรบกวนที่เกิดขึ้นในวงควบคุม

การออกแบบตัวควบคุมในแนวใหม่ แสดงด้วยแผนภาพกรอบง่ายๆ ที่ประกอบด้วย

- เจเนอรัล ไรซ์เพลนท (Generalized Plant) $G(s)$
- ตัวควบคุม $K(s)$ ที่ต้องการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการออกแบบตัวควบคุมในแนวใหม่

โดยที่

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

รูปที่ 2.7 แสดงถึงความสัมพันธ์ระหว่างตัวแปรและสัญญาณ

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

รูปที่ 2.8 แสดงถึงลักษณะของเจนเนอรัลไรซ์แพลนท์

$$\frac{d}{dt}x(t) = Ax(t) + B_1w(t) + B_2u(t)$$

$$z(t) = C_1x(t) + D_{12}u(t)$$

$$y(t) = C_2x(t) + D_{21}w(t)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เจนอรัลไรซ์แพลนท์ $G(s)$ ประกอบไปด้วย

- แพลนท์

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

รูปที่ 2.9 แสดงถึงลักษณะของแพลนท์

- ฟังก์ชันถ่วงน้ำหนัก (Weighting Functions)

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

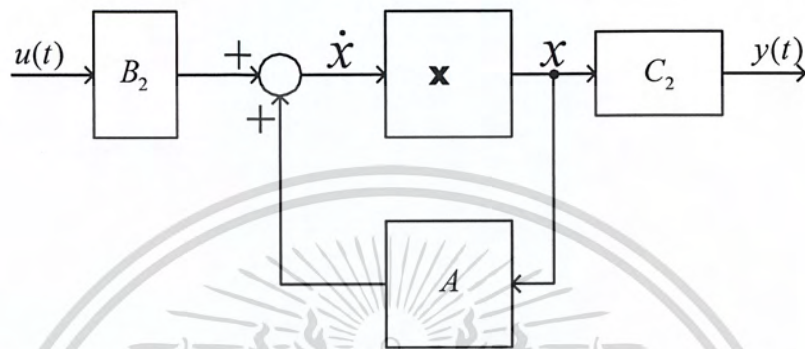
รูปที่ 2.10 แสดงถึงลักษณะของฟังก์ชันถ่วงน้ำหนัก

$$\begin{array}{|l}
 \frac{d}{dt}x(t) = Ax(t) + \boxed{\quad} + B_2u(t) \\
 y(t) = C_2x(t) + \boxed{\quad}
 \end{array}
 \rightarrow y(t)$$

Plant Dynamics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



2.3.2 การออกแบบตัวควบคุมแอสทิวอาร์ (LQR)

ระบบที่ได้รับการปรับ คชนี้สมรรถนะ (Performance Indexes) ให้มีค่าต่ำที่สุด เรามักเรียกกันว่า ระบบควบคุมที่เหมาะสมหรือดีที่สุด

โดยทั่วไป คชนี้สมรรถนะ ของระบบเขียนได้เป็น

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

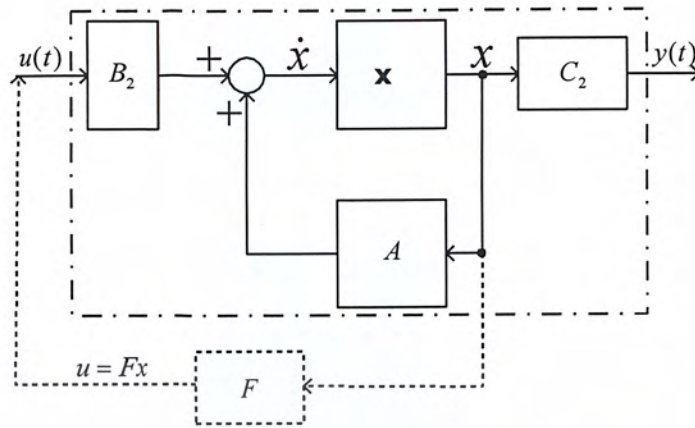
$$\begin{array}{l} u_1 \rightarrow \\ u_2 \rightarrow \\ \vdots \\ u_m \rightarrow \end{array} \begin{array}{c} \boxed{\text{Control}} \\ \boxed{\text{system}} \end{array} \begin{array}{l} \rightarrow x_1 \\ \rightarrow x_2 \\ \vdots \\ \rightarrow x_n \end{array} \left[\begin{array}{c} u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \right] = \left[\begin{array}{cccc} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & & & f_{2n} \\ \vdots & & & \vdots \\ f_{m1} & & \cdots & f_{mn} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array} \right]$$

$$u = Fx$$

การหา $u = Fx$ ที่เป็นฟังก์ชันของ x เพื่อให้ได้ J ต่ำที่สุด

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงถึงลักษณะของระบบ LQR

เราจะไปหา P จากสมการริคคาดี (Riccati Equation)

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

(2.3)

คำตอบ (Solutions) ของ P ที่ได้จากการกำหนด ฟังก์ชันถ่วงน้ำหนัก Q, R จะเป็นกฎการควบคุมที่เหมาะสมที่สุด

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

(2.4)

สรุป

อัตราขยายสถานะป้อนกลับ (State Feedback Gain) F ทำให้ ตำแหน่งของโพลูปิด (Closed-Loop Poles) ไปอยู่ ณ จุดต่างๆ ได้ ถ้าระบบสามารถควบคุมได้ กล่าวคือ

$$|M_c| = |B_2 \quad AB_2 \quad A^2 B_2 \quad \dots \quad A^{n-1} B_2| \neq 0$$

(2.5)

ผลจากการที่โพลูปิดอยู่ ณ ตำแหน่งใด ๆ ก็จะทำให้ได้ผลตอบสนองด้านเอาต์พุต (Output Responses) แตกต่างกัน

ผลของ Q และ R ทำให้ได้ตำแหน่งโพลูปิดเข้าหาหรือออกจากแกนอิมเมจินนรี (Imaginary) ตามค่าที่ให้เข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณา

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

$$z^T(t) = [C_1 x(t) + D_{12} u(t)]^T = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^T \begin{bmatrix} C_1 & D_{12} \end{bmatrix}^T \tag{2.6}$$

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

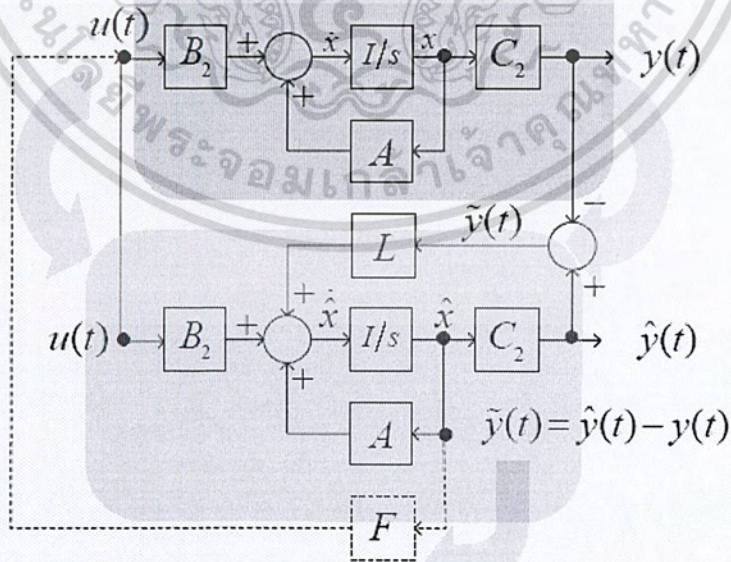
$$= x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u + \underbrace{x^T C_1^T D_{12} u + u^T D_{12}^T C_1 x}_{\text{cross-term}}$$

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

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

$$\int_0^\infty (z^T z) dt = \|z(t)\|_2^2 \quad \|z(t)\|_2^2 \equiv \int_0^\infty (x^T Q x + u^T R u) dt = J \tag{2.7}$$

2.3.3 การออกแบบตัวควบคุม LQE



รูปที่ 2.15 แสดงถึงลักษณะของระบบ LQR ร่วมกับ LQE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราต้องการสร้างระบบทางพลศาสตร์หรือที่ทราบกันว่า คือตัวสังเกต ที่สามารถประมาณสแตตภายใน (Internal State) $x(t)$ จากข้อมูลความรู้ที่ให้มา อันได้แก่ปริมาณเข้า $u(t)$ และปริมาณออก $y(t)$ โดยงานนี้สามารถที่จะสำเร็จลุล่วงได้

ซึ่งรูปข้างบนแสดงถึงการแยกออกเป็น 2 ส่วน อันได้แก่ ส่วนที่เป็นทางพลศาสตร์ของแพลนท์จริงๆ (A, B_2, C_2) และส่วนที่ทำการแก้ไขความคลาดเคลื่อน $L\tilde{y} = L(\hat{y} - y)$ สำหรับเมตริกซ์ L นั้น เราเรียกว่า อัตราขยายของตัวสังเกต เราจะสังเกตได้ว่า ตัวสังเกตจะมี n สแตตภายใน \hat{x} และสองปริมาณเข้า ได้แก่ $u(t)$ และ $y(t)$ เรากำลังจะแสดงให้เห็นว่า \hat{x} นั้นจะให้ค่าประมาณของทุกสแตต $x(t)$ ถ้า L ถูกเลือกอย่างถูกต้อง ดังนั้น \hat{x} ก็คือสแตตที่ประมาณและปริมาณออกของตัวสังเกตนั่นเอง

สมการสแตตของตัวสังเกตคือ

$$\dot{\hat{x}} = A\hat{x} + B_2u + L\tilde{y} = A\hat{x} + B_2u + L(\hat{y} - y) \quad (2.8)$$

หรือ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly \quad (2.9)$$

นี่คือระบบทางพลศาสตร์อันดับที่ n ที่มีสแตตเริ่มต้น (Initial State) $\hat{x}(0)$ เท่ากับค่าประมาณเริ่มต้นของสแตต ส่วนเมตริกซ์ L ที่เป็นอัตราขยายของตัวสังเกตนั้น เราจะต้องเลือกค่าของมัน เพื่อที่ว่า ถึงแม้ค่าประมาณเริ่มต้น $\hat{x}(0)$ จะไม่เท่ากับสแตตเริ่มต้นจริง $x(0)$ ครั้นเวลาผ่านไป สแตตที่ประมาณ $\hat{x}(t)$ จะลู่เข้าหาสแตตจริง $x(t)$

ปริมาณ $\tilde{y}(t) = \hat{y}(t) - y(t)$ นั้นเราจะเรียกว่า ความคลาดเคลื่อนของการประมาณค่าปริมาณออก (Output Estimation Error) ในการเลือก L เราจะนิยามให้ ความคลาดเคลื่อนของการประมาณค่าสแตต (State Estimation Error) $\tilde{x}(t) = \hat{x}(t) - x(t) = e(t)$ และเขียนพลศาสตร์ของมันได้ดังนี้

$$\begin{aligned}
\dot{\tilde{x}} &= \dot{\hat{x}} - \dot{x} = \dot{e} \\
&= [A\hat{x} + B_2\bar{u} + L(\hat{y} - y)] - [Ax + B_2\bar{u}] \\
&= A(\hat{x} - x) + L(\hat{y} - y) \\
&= A(\hat{x} - x) + LC_2\hat{x} - Ly \\
&= A(\hat{x} - x) + LC_2\hat{x} - LC_2x \\
&= A(\hat{x} - x) + LC_2(\hat{x} - x) \\
&= (A + LC_2)(\hat{x} - x)
\end{aligned}$$

หรือ

$$\dot{\tilde{x}} = (A + LC_2)\tilde{x} \equiv A_o\tilde{x} \quad (2.10)$$

เราจะสังเกตได้ว่าปริมาณควบคุมทางเข้า (Control Input) \bar{u} ไม่ปรากฏให้เราเห็นเลยจากการหักล้างกันเอง ทั้งนี้ก็เนื่องมาจากว่าปริมาณเข้า \bar{u} ได้ถูกป้อนเข้าตรงๆ เข้าไปใน ตัวสังเกต โดยผ่านเมตริกซ์ B_2

สมการนี้เป็นที่ทราบกันว่า คือความคลาดเคลื่อนพลศาสตร์ (Error Dynamics) ซึ่งหลายๆ ที่มาในทฤษฎีของระบบควบคุมแบบป้อนกลับ แขนงอยู่บนจุดสำคัญในเรื่องของการวิเคราะห์ความคลาดเคลื่อนพลศาสตร์ เช่นวิชาที่เกี่ยวกับการวัดทางวิศวกรรม (Engineering Measurements) เป็นต้น จากสมการนี้

$$(\dot{\tilde{x}} = (A + LC_2)\tilde{x} \equiv A_o\tilde{x}) \quad (2.11)$$

เราจะเห็นได้อย่างง่ายดายเลยว่า トラบเท่าที่เราเลือกค่าอัตราขยายของตัวสังเกต L เพื่อให้ตัวเมตริกซ์วงปิดของตัวสังเกต (Closed-Loop Observer Matrix) จะได้มีเสถียรภาพแบบเชิงเส้นกำกับ (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.) ถ้าเราเปรียบเทียบปัญหานี้กับปัญหาของการเลือกค่าอัตราขยายของตัวแปรสแตตป้อนกลับ (State Variable Feedback; SVFB) F เพื่อว่า

$$A_{CL} = (A + B_2F) \quad (2.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$A_o^T = (A + LC_2)^T = A^T + C_2^T L^T \quad (2.13)$$

ขณะนี้ สมการข้างบน เมื่อเราดูแล้ว จะเห็นได้ว่ามันก็เหมือนกับปัญหาของการเลือกค่าอัตราขยายของตัวแปรสเตตป้อนกลับ 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 ปัญหาของการออกแบบเพื่อเลือกค่าอัตราขยายของตัวแปรสเตตป้อนกลับ F นั้น ใช้ (A, B_2) ในขณะที่ การออกแบบตัวสังเกต ใช้ (A, C_2) จริงๆ แล้ว ทั้งสองปัญหานี้ มีลักษณะเหมือนกัน กล่าวคือ ถ้าเราเปรียบเทียบการออกแบบอัตราขยายของตัวแปรสเตตป้อนกลับที่เกี่ยวข้องกับ (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)
- ใช้เทคนิคในการออกแบบ เพื่อหาค่าอัตราขยายของ ตัวแปรสเตตป้อนกลับ เพื่อให้ระบบมีเสถียรภาพ F โดยวิธีการใดวิธีการหนึ่ง เช่น เทคนิคในการวางตำแหน่งโพล (Poles Assignment or Poles Placement) หรือไม่ก็อาจจะใช้เทคนิคในการหาอัตราขยาย F ที่เหมาะสมที่สุด จากการที่เราได้วางเงื่อนไขไว้แล้ว

- เปลี่ยนชื่อจาก F^T ไปเป็น L

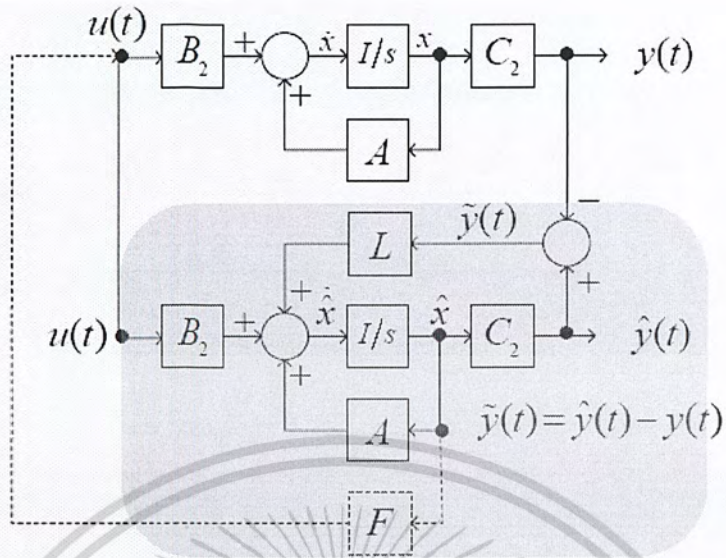
จากที่ผ่านมา เรามีสมการสเตตของตัวสังเกตเป็น

$$\dot{\hat{x}} = A\hat{x} + B_2u + L\hat{y} = A\hat{x} + B_2u + L(\hat{y} - y) \quad (2.14)$$

หรือ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly \quad (2.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงถึงลักษณะของระบบ LQE

ในการออกแบบตัวควบคุมที่เหมาะสมที่สุด โดยการป้อนกลับด้วยอัตราขยายสเตต F ไปเป็นปริมาณเข้า $u = Fx$ ให้แก่แพลนท์ เพื่อให้ได้ดัชนีสมรรถนะ J มีค่าต่ำสุด เรียกว่า “LQR”

เดิมทีนั้น ยังไม่มีตัวสังเกตหรือตัวประมาณค่าสเตต เราจึงป้อนกลับจากการวัดสเตต x ตรงๆ ฆาบัดนี้ เรามีตัวประมาณค่าสเตต x นั้นแล้ว ซึ่งค่าประมาณของ x ก็คือ \hat{x} ถ้าหากว่าเราใช้วิธีการหาอัตราขยาย L เช่นเดียวกับที่หา F ใน “LQR” แล้ว เราจะเรียกมันว่า Linear Quadratic Estimator (LQE) และถ้าเราใช้การป้อนกลับด้วยอัตราขยายสเตต F กับ \hat{x} เพื่อไปเป็นปริมาณเข้า $u = F\hat{x}$ ให้แก่แพลนท์ เราจะเรียกตัวควบคุมที่มีทั้ง LQE และ LQR รวมกันนี้ว่า Linear Quadratic Gaussian (LQG) Controller ดังนั้น เมื่อเราแทนค่า $u = F\hat{x}$ (โดยที่มัน คือปริมาณออกของ ตัวควบคุม LQG) ในสมการสมการสเตตของตัวสังเกต จะได้

$$\begin{aligned} \dot{\hat{x}} &= (A + LC_2)\hat{x} + B_2u - Ly \\ \dot{\hat{x}} &= (A + B_2F + LC_2)\hat{x} - Ly \end{aligned} \tag{2.16}$$

สรุป สมการสเตตและสมการปริมาณออก ของตัวควบคุม LQG คือ

$$\begin{aligned} \dot{\hat{x}} &= (A + B_2F + LC_2)\hat{x} - Ly \\ u &= F\hat{x} \end{aligned} \tag{2.17}$$

สามารถเขียนในรูปขวเลข (Shorthand) ได้เป็น

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

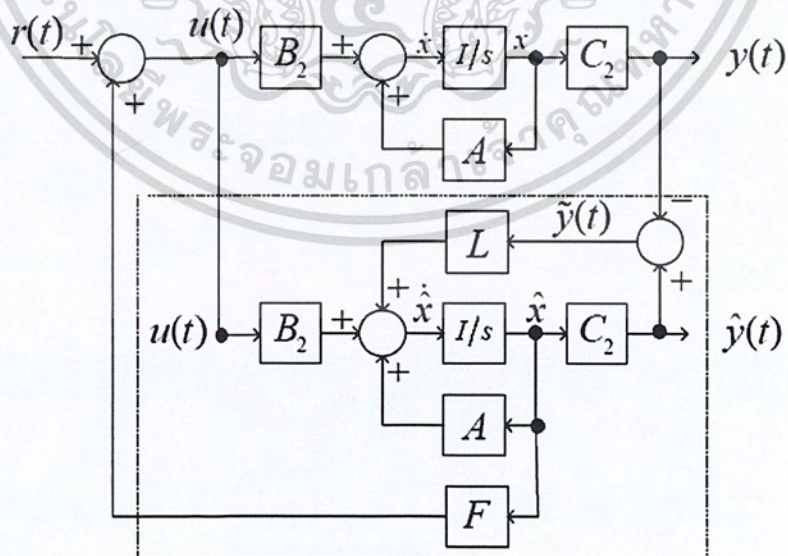
ซึ่งตัวควบคุม LQG นี้ จะออกแบบเพื่อนำไปควบคุมแพลนท์

$$u(t) \rightarrow \begin{cases} \frac{d}{dt}x(t) = Ax(t) + B_2u(t) \\ y(t) = C_2x(t) \end{cases} \rightarrow y(t)$$

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 $r(t)$



รูปที่ 2.17 แสดงถึงลักษณะของระบบ LQG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความคลาดเคลื่อนของการประมาณค่าปริมาณออก (Output Estimation Error)

$$\tilde{y} = \hat{y} - y \quad (2.18)$$

ความคลาดเคลื่อนของการประมาณค่าสแตต (State Estimation Error)

$$\tilde{x} = \hat{x} - x = e \quad (2.19)$$

ดังนั้น อนุพันธ์ ของความคลาดเคลื่อนหรือความคลาดเคลื่อนทางพลศาสตร์

$$\dot{\tilde{x}} = \dot{\hat{x}} - \dot{x} = \dot{e} \quad (2.20)$$

แต่จากที่สมการสแตตของตัวประมาณ

$$\dot{\hat{x}} = (A + LC_2)\hat{x} + B_2u - Ly \quad (2.21)$$

และสมการสแตตของระบบทางพลศาสตร์

$$\dot{x} = Ax + B_2u \quad (2.22)$$

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

$$\begin{aligned} \dot{x} &= Ax + B_2u; \quad \boxed{u = r(t) + F\hat{x}} \\ &= Ax + B_2r(t) + B_2F\hat{x}; \quad \boxed{\hat{x} = \tilde{x} + x} \\ &= Ax + B_2r(t) + B_2F(\tilde{x} + x) \\ \dot{x} &= (A + B_2F)x + B_2F\tilde{x} + B_2r(t) \quad (2.23) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบวงปิด ที่ประกอบด้วย แพลนท์ และตัวควบคุม จะมีสมการสแตตเป็น

$$\begin{aligned}\dot{x} &= (A + B_2F)x + B_2F\tilde{x} + B_2r(t) \\ \dot{\tilde{x}} &= (A + LC_2)\tilde{x}\end{aligned}\quad (2.24)$$

และถ้าพิจารณาว่าปริมาณออกของแพลนท์ เป็นปริมาณออก แล้ว จะได้

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

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} (A + B_2F) & (B_2F) \\ 0 & (A + LC_2) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} r(t) \quad (2.25)$$

$$y = \begin{bmatrix} C_2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = C_2x \quad (2.26)$$

ฟังก์ชันถ่ายโอนของแพลนท์

$$\begin{aligned}u(t) &\rightarrow \begin{cases} \dot{x}(t) = Ax(t) + B_2u(t) \\ y(t) = C_2x(t) \end{cases} \rightarrow y(t) \\ U(s) &\rightarrow \boxed{C_2(sI - A)^{-1}B_2} \rightarrow Y(s) \end{aligned}$$

ฟังก์ชันถ่ายโอนวงปิด

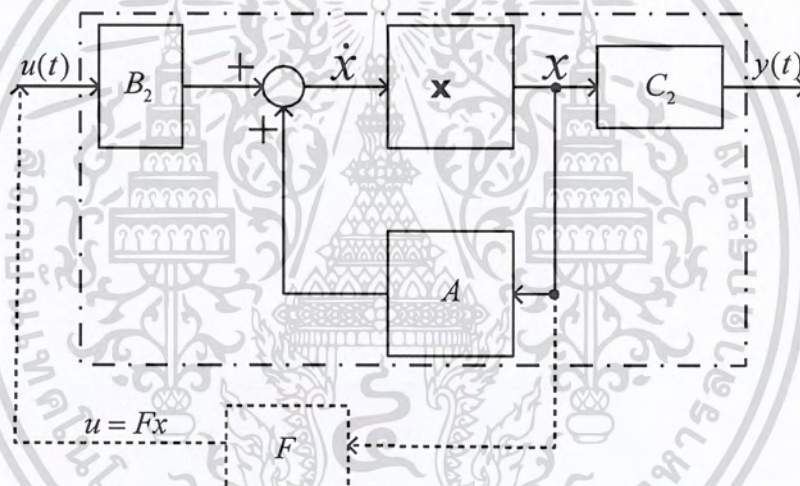
$$\begin{aligned}H(S) &= \begin{bmatrix} C_2 & 0 \end{bmatrix} \begin{bmatrix} sI - (A + B_2F) & -(B_2F) \\ 0 & sI - (A + LC_2) \end{bmatrix}^{-1} \begin{bmatrix} B_2 \\ 0 \end{bmatrix} \\ H(S) &= C_2 \left[sI - (A + B_2F) \right]^{-1} B_2 \quad (2.27) \end{aligned}$$

ความเป็นทวิ และความสามารถสังเกต (Duality and Observability)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ แพลนท์ (A, B_2, C_2) มา จะเป็นที่ยอมรับกันว่า แพลนท์ (A^T, C_2^T, B_2^T) ก็คือระบบคู่ สำหรับ (A, B_2, C_2) ในระบบคู่ที่ว่านี้ ผลที่เกิดขึ้นเนื่องมาจาก ปริมาณเข้า และปริมาณออก สามารถแลกเปลี่ยนระหว่างกันและกันได้เป็นอย่างดี

ระบบ (A, B_2, C_2) จะเรียกได้ว่า เข้าถึงได้ (Reachable) ถ้าหากว่าปริมาณควบคุมทางเข้า (Control Input) u สามารถที่จะเลือกขับ (Drive) สแตตเริ่มต้น (Initial State) $x(0)$ ใดๆ ไปยัง สแตตสุดท้าย (Final State) $x(T)$ ที่เราต้องการ ที่เวลาสุดท้าย (Final Time) T ได้ เรารู้ว่า ถ้าระบบสามารถเข้าถึงได้แล้ว เราก็สามารถที่จะกำหนดตำแหน่งหรือวางโพล (Poles) ณ ที่ใดๆ ได้ตามอำเภอใจของเรา นอกจากนี้เรายังรู้อีกว่า ถ้าระบบสามารถเข้าถึงได้แล้ว เราสามารถออกแบบเพื่อเลือกค่าอัตราขยายของตัวแปรสแตตป้อนกลับ F ของปัญหา LQR (Linear Quadratic Regulator) โดยสรุปได้ดังนี้



รูปที่ 2.18 แสดงถึงลักษณะของระบบ LQR

LQR เป็นปัญหาของการหา F สำหรับระบบ (A, B_2) ซึ่ง $u = Fx$ เพื่อที่ว่า

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt = \int_0^{\infty} (x^T C_1^T C_1 x + u^T D_{12}^T D_{12} u) dt \quad \text{มีค่าต่ำที่สุดโดย}$$

$$F = -R^{-1} B_2^T P \quad \text{หาได้จาก } P \text{ ซึ่งเป็นคำตอบของสมการรีกคาติ}$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ A, B_2 ทราบค่าได้จากสมการสแตตของระบบ ($\dot{x} = Ax + B_2u$) และเมตริกซ์ Q, R เป็นน้ำหนักที่เราให้ความสำคัญแก่การเบี่ยงเบนของสแตตจากจุดกำเนิด ($x^T Qx$) และค่าใช้จ่ายในการควบคุม ($u^T Ru$) ตามลำดับ

2.3.4 ระบบคู่ (Dual System)

$$(A, B_2, C_2) \Leftrightarrow (A^T, C_2^T, B_2^T) \quad (2.28)$$

จากที่ว่าในระบบคู่ นั้น ผลที่เกิดขึ้นเนื่องมาจาก ปริมาณเข้าและปริมาณออก สามารถแลกเปลี่ยนระหว่างกันและกันได้เป็นอย่างดี

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

เจเนอรัลไรซ์แพลนท์ $G(s)$ คือ

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

สมการสแตตและสมการปริมาณออก ของตัวควบคุม LQG คือ

$$\begin{aligned} \dot{\hat{x}} &= (A + B_2 F + L C_2) \hat{x} - L y \\ u &= F \hat{x} \end{aligned}$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป $\boxed{\text{LQR ทหา } F}$; $\boxed{\text{LQE ทหา } L}$; $\boxed{\text{LQG ทหาทั้ง } F \text{ และ } L}$

พิจารณาความสัมพันธ์ระหว่างดัชนีสมรรถนะ J และความคลาดเคลื่อน $z(t)$

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

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

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

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

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

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

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

เราจะเห็นได้ว่า ตัวควบคุม LQR ซึ่งเป็นส่วนที่ออกแบบเพื่อหาค่า F ของตัวควบคุม LQG นั้น เป็นกรณีพิเศษของ H_2 Optimal Control เพราะตัด ครอสเทอม(Cross-Term) ของ $Z^T Z$ ทิ้งไป เราจึงเรียก ตัวควบคุม LQG ว่าเป็น H_2 Basic Problem วงควบคุมที่ประกอบไปด้วยตัวควบคุม LQG และแพลนท์ นั้น จะมีฟังก์ชันถ่ายโอนวงปิดเป็น

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระบบวงปิดเป็น

$$\begin{aligned}\dot{x} &= (A + B_2F)x + B_2F\tilde{x} + B_2r(t) \\ \dot{\tilde{x}} &= (A + LC_2)\tilde{x} \\ \begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} &= \begin{bmatrix} (A + B_2F) & (B_2F) \\ 0 & (A + LC_2) \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} + \begin{bmatrix} B_2 \\ 0 \end{bmatrix} r(t) \\ y &= [C_2 \quad 0] \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} = C_2x\end{aligned}$$

จะเห็นได้ว่าเมตริกซ์ระบบ (System Matrix) วงปิด คือ

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

ดังนั้น เราสามารถหาโพลวงปิด (Closed-Loop Poles) ได้จาก

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

เมื่อ I_n เป็นเมตริกซ์เอกลักษณ์ (Identity Matrix) ขนาด $n \times n$ แต่เนื่องจากว่า A_{LQG} เป็นเมตริกซ์สามเหลี่ยมบน (Upper Triangular Matrix) ดังนั้น ผลคูณของตัวกำหนด ตามเมตริกซ์เฉียง (Diagonal Matrix) คือ

$$\Delta(s) = |sI_n - (A + B_2F)| |sI_n - (A + LC_2)| \quad (2.29)$$

เราจะเห็นได้ว่า โพลวงปิด จะมีจำนวน $2n$ ตัว ถ้าเราออกแบบตัวควบคุมที่มีตัวสังเกต หรือตัวประมาณค่าสเตรตกรวมอยู่ด้วย เพราะโพลในวงปิดจะมีทั้งโพลของตัวสังเกตรวมกันอยู่กับ โพลของแพลนท์ ซึ่งเรียกกันว่า “หลักการแยกกัน (Separation Principle)” และถ้าจะว่าไปแล้ว มัน เป็นหัวใจของทฤษฎีควบคุมสมัยใหม่ (Modern Control Theory) เลยก็ว่าได้ความหมายของ Separation Principle ก็คือว่า มันจะมีอยู่ 2 ขั้นตอนการปฏิบัติเพื่อออกแบบตัวควบคุม ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้เทคนิคอันไหนก็ได้ เพื่อเลือกเมตริกซ์ป้อนกลับ (Feedback Matrix) F โดยสมมติว่ามันสามารถใช้การป้อนกลับด้วย Full state ได้

- ออกแบบตัวสังเกต L

โดยที่ แพลนที่มีสมการสแตตเป็น

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

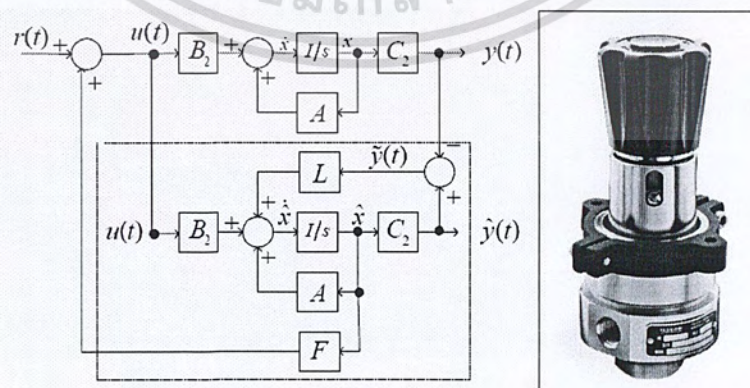
ตัวสังเกตมีสมการสแตตเป็น

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

ปริมาณควบคุมทางเข้า

$$u = r(t) + F\hat{x}$$

ถ้าหากว่า $r(t)$ ซึ่งเราเคยชินกันว่ามัน คือปริมาณอ้างอิงทางเข้าหรือคำสั่ง (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 ซึ่งใช้ไอซีประเภทที่ทีแอลว่า ตัวรักษาระดับแรงดันไฟฟ้ากระแสตรง 5 โวลต์ ก็หมายถึงว่า ปริมาณออกของแหล่งจ่ายแรงดันไฟฟ้ากระแสตรง จะถูกรักษาให้มีค่าคงที่ที่ 5 โวลต์ ตัวอย่างเช่น ตัวรักษาระดับความดัน (Pressure Regulator) เป็นต้น



รูปที่ 2.19 แสดงถึงลักษณะของตัวปรับแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

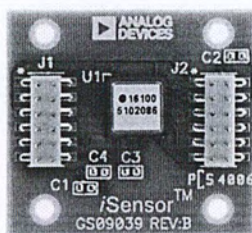
2.4 หลักการทำงานของเซนเซอร์

ในการควบคุมการเคลื่อนที่ของหุ่นยนต์รักษาสมดุลให้ได้ตามต้องการนั้นการควบคุมที่เป็นสิ่งสำคัญที่สุด คือการควบคุมการหมุนของมอเตอร์ซึ่งเป็นสิ่งสำคัญที่สุดที่จะทำให้หุ่นยนต์เคลื่อนที่ไปด้วยความเร็วที่ต้องการทิศทางที่ถูกต้องแม่นยำ ซึ่งการควบคุมการทำงานของมอเตอร์ เป็นการควบคุมแบบดิจิทัล ซึ่งได้จากการการเขียนโปรแกรมในเอ็มพีแลปโดยใช้ไมโครคอนโทรลเลอร์คือ dsPIC30f6010A เป็นตัวควบคุมมอเตอร์กระแสตรง ซึ่งทำงานร่วมกับตัวเซ็นเซอร์วัดความเร็วและเซนเซอร์วัดความเอียง โดยจะตรวจจับหรือวัดสัญญาณป้อนกลับค่ามุมแกว่งและตำแหน่งของหุ่นยนต์เพื่อรักษาสมดุล

2.4.1 เซนเซอร์ไจโรสโคป

โครงการนี้ใช้ไจโรสโคปทำหน้าที่ในการวัดความเร็วเชิงมุมของหุ่นยนต์เมื่อมีการเคลื่อนที่หรือมีการเปลี่ยนแปลงของมุมเอียง ซึ่งเอาต์พุตของไจโรสโคปนั้นก็จะมีค่าขึ้นลงตามทิศทางของการเอียงของหุ่นยนต์ หากดูเผินๆแล้วก็เหมือนกับว่าเราสามารถที่จะใช้แค่ไจโรสโคป เป็นเซนเซอร์เพียงอย่างเดียวก็น่าจะพอ แต่ในความเป็นจริงไจโรสโคป ทำงานในลักษณะที่เป็นการตอบสนองเชิงพลวัต (Dynamic Response) นั้นหมายความว่าเมื่อหุ่นยนต์มีการเอียงหรือมีการเปลี่ยนแปลงมุมไปมาเอาต์พุตที่จ่ายออกมาจากตัวไจโรสโคป ก็จะมีการสวิงขึ้นหรือลงขึ้นอยู่กับว่าหุ่นยนต์นั้นเอียงไปในทิศทางใด แต่หากเมื่อไรก็ตามที่หุ่นยนต์เคลื่อนที่ไปแล้วหยุดการเคลื่อนที่ไม่ว่าจะหยุดในมุมใดก็ตามแต่ เอาต์พุตของไจโรสโคป ก็จะมีค่าสวิงกลับมาที่จุดอ้างอิงเหมือนเดิม ซึ่งในไจโรสโคปที่เราเลือกใช้นั้นจะมีค่าแรงดันอ้างอิงเท่ากับ 2.5 V การที่มันมีพฤติกรรมแบบนี้ ทำให้ตัวไจโรสโคปนั้นไม่สามารถวัดค่าทางสถิต (Static) ได้ ถ้าหุ่นยนต์หยุดนิ่งเมื่อไหร่ก็จะให้ค่าเอาต์พุตกลับมายังจุดอ้างอิงเสมอ การที่มันมีพฤติกรรมแบบนี้ทำให้เกิดค่าเออเรอร์ (Error) ขึ้นได้แม้ขณะที่หุ่นยนต์เอียงค้างอยู่ เพราะมันจะมองดูเหมือนกับว่าหุ่นยนต์นั้นตั้งตรงอยู่ตลอดเวลา ซึ่งถ้าหากเวลาผ่านไปนานๆ ค่าของมุมเอียงนี้จะเพิ่มขึ้นเรื่อยๆและเกิดการสะสมของเออเรอร์ ที่มากขึ้นตามและนั่นก็คือการเอียงที่มากขึ้นเช่นกัน แต่ในความเป็นจริงแล้วหุ่นยนต์อาจจะไม่ได้เอียงก็เป็นได้

ดังนั้นในที่สุดก็จะทำให้หุ่นยนต์นั้นล้มลงมานั่นเอง ค่าเอาต์พุตของไจโรสโคป จะมีค่ามากหรือน้อยขึ้นอยู่กับความเร็วหรือช้าของหุ่นยนต์ ถ้าหากว่าหุ่นยนต์เอียงเร็วหรือว่าล้มเร็วก็จะให้เอาต์พุตที่มีการสวิงมาก ในทางตรงกันข้ามถ้าหากหุ่นยนต์มีการเอียงอย่างช้าๆ ก็จะทำให้เอาต์พุตเปลี่ยนแปลงไปแค่นิดเดียวจากจุดอ้างอิง



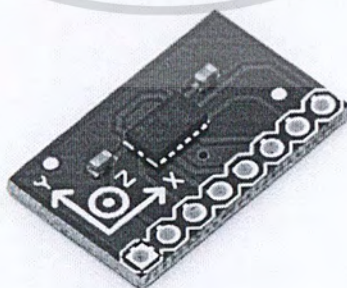
รูปที่ 2.20 เซนเซอร์ใจโรสโคป

2.4.2 เซนเซอร์แอกเซลเลอโรมิเตอร์

แอกเซลเลอโรมิเตอร์ เป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งใช้ในการวัดความเร่งซึ่งสามารถประยุกต์ใช้งานได้หลากหลาย โดยเราใช้แอกเซลเลอโรมิเตอร์ ในการวัดความเร่งที่เป็นแรงดึงดูดของโลก นั่นก็คือค่า g นั่นเอง ดังนั้นเราสามารถสรุปได้ว่า ถ้าเรามีแอกเซลเลอโรมิเตอร์ เราก็สามารถวัดแรงโน้มถ่วงของโลกได้ ซึ่งแน่นอนมันสามารถจะวัดได้ทั้งปริมาณและทิศทางของความเร่ง

ข้อดีของแอกเซลเลอโรมิเตอร์ นั่นคือ เมื่อหุ่นยนต์มีการเอียงตัวค้างสถานะอยู่มันก็ยังคงให้เอาต์พุตออกมาเพื่อที่จะบอกให้รู้สถานะของการเอียงของหุ่นยนต์ ค่าเอาต์พุตที่ได้นั้นสามารถที่จะเอามาทำการคำนวณมุมได้ ทั้งนี้แอกเซลเลอโรมิเตอร์ ไม่ใช่แค่จะวัดความเอียงในขณะที่หุ่นยนต์อยู่นิ่งๆเท่านั้น แต่แอกเซลเลอโรมิเตอร์ ยังสามารถที่จะวัดความเอียงทั้งแบบพลวัต (Dynamic) ได้ด้วย ดูแล้วก็เหมือนกับว่าแอกเซลเลอโรมิเตอร์ เพียงอย่างเดียวก็สามารถที่จะทำงานเป็นเซนเซอร์ให้หุ่นยนต์ของเราได้แล้วโดยไม่ต้องใช้ใจโรสโคป ร่วมด้วย

แต่ในความเป็นจริงแล้วแอกเซลเลอโรมิเตอร์ มีความเร็วตอบสนองการเปลี่ยนแปลงของมุมน้อย ก็คือ มันสามารถที่จะวัดมุมได้ช้านั่นเอง ดังนั้นจึงต้องมีความจำเป็นที่จะต้องใช้ใจโรสโคปเข้ามาทำงานร่วมกับ แอกเซลเลอโรมิเตอร์และนี่เองก็เป็นที่มาของการใช้เซนเซอร์ 2 ตัวร่วมกัน

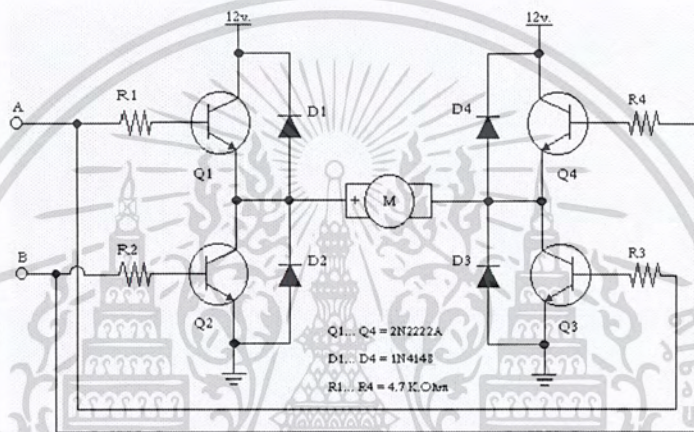


รูปที่ 2.21 เซนเซอร์แอกเซลเลอโรมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

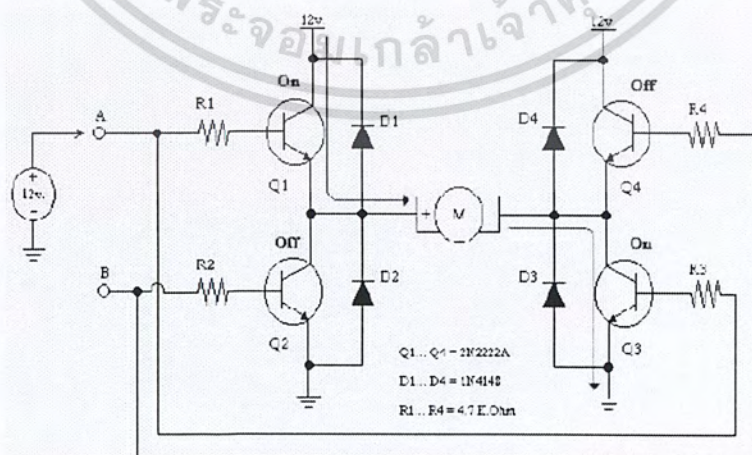
2.5 การควบคุมการหมุนของ ดีซีมอเตอร์

การควบคุมทิศทางการหมุนของ ดีซีมอเตอร์ โดยสร้างวงจร H-Bridge Switching Transistor วงจรในการควบคุมนี้ประกอบไปด้วย ทรานซิสเตอร์, ตัวต้านทานและไดโอด ซึ่งสาเหตุที่เลือก ทรานซิสเตอร์ เนื่องจากข้อดีที่สามารถตอบสนองจังหวะของการปิด/เปิดสวิตช์ได้นับล้านครั้งต่อวินาทีและที่สำคัญไม่ทำให้เกิดปัญหาการบวมจากสนามแม่เหล็ก โดยตัวต้านทานจะทำหน้าที่จำกัดกระแส Ib ส่วนไดโอดทำหน้าที่ป้องกันกระแสไหลย้อนกลับจากมอเตอร์ ในขณะที่ทรานซิสเตอร์หยุดทำงาน



รูปที่ 2.22 แสดงวงจร H-Bridge Switching Transistor

2.5.1 กรณีที่ Q1 และ Q3 ทำงาน

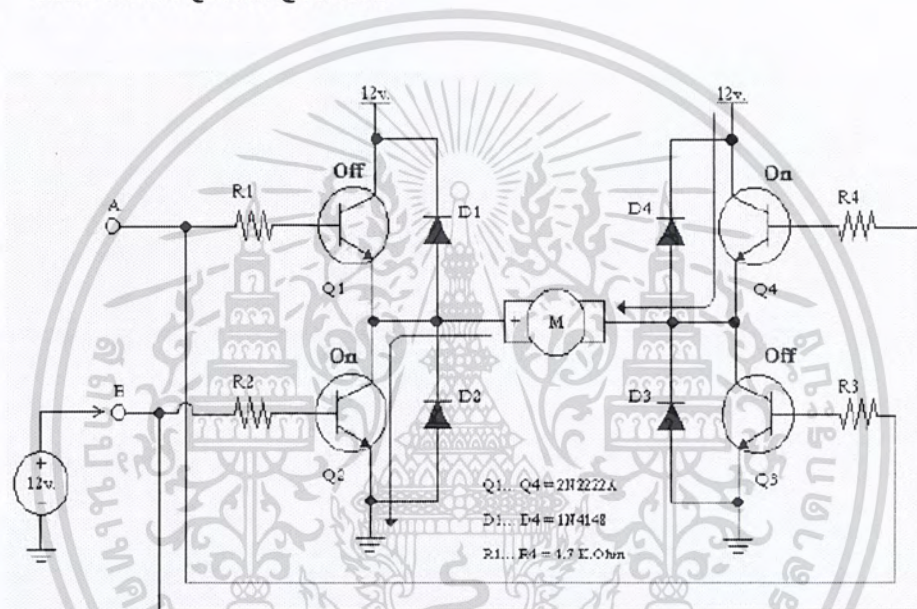


รูปที่ 2.23 แสดงวงจร H-Bridge Switching Transistor เมื่อ Q1 และ Q3 ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการจ่ายแรงดัน 12 V เข้าที่จุด A ทำให้มีกระแสไหลผ่าน R1 เข้าสู่ขาเบส (Base) ของ Q1 และมีกระแสไหลผ่าน R3 เข้าสู่ขาเบส ของ Q3 ทำให้ Q1 และ Q3 ทำงาน (On) เปรียบเสมือน สวิตช์ปิดวงจร ส่งผลให้มีกระแสไหลจากแหล่งจ่าย 12V ผ่านขาคอลเลกเตอร์ (Collector) และ อิมิตเตอร์ (Emitter) ของ Q1 ผ่านเข้าสู่ขั้วบวก (+) ของมอเตอร์ ผ่านไปยังขาคอลเลกเตอร์และ อิมิตเตอร์ ของ Q3 ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางบวกและครบวงจร จึงทำให้มอเตอร์ สามารถหมุนในทิศทางฟอร์เวิร์ด (Forward) ได้

2.5.2 กรณีที่ Q2 และ Q4 ทำงาน



รูปที่ 2.24 แสดงวงจร H-Bridge Switching Transistor เมื่อ Q2 และ Q4 ทำงาน

เมื่อมีการจ่ายแรงดัน 12 V เข้าที่จุด B ทำให้มีกระแสไหลผ่าน R2 เข้าสู่ขาเบส ของ Q2 และมีกระแสไหลผ่าน R4 เข้าสู่ขาเบส ของ Q4 ทำให้ Q2 และ Q4 ทำงาน (On) เปรียบเสมือน สวิตช์ปิด วงจร ส่งผลให้มีกระแสไหลจากแหล่งจ่าย 12V ผ่านขาคอลเลกเตอร์ และอิมิตเตอร์ ของ Q4 ผ่านเข้าสู่ขั้วบวก (-) ของมอเตอร์ ผ่านไปยังขาคอลเลกเตอร์และอิมิตเตอร์ ของ Q2 ทำให้มีกระแสไหลผ่าน มอเตอร์ในทิศทางบวกและครบวงจร จึงทำให้มอเตอร์หมุนในทิศทางรีเวิร์ส (Reverse) ได้

2.6 การออกแบบไมโครคอนโทรลเลอร์

2.6.1 ข้อมูลเบื้องต้นของไมโครคอนโทรลเลอร์แบบ dsPIC

dsPIC คือชื่อของไมโครคอนโทรลเลอร์แบบ 16 บิต มีชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรลเลอร์อนุกรมใหม่นี้ว่า Digital Signal Controller หรือ DSP นั่นหมายความว่า dsPIC เป็น ไมโครคอนโทรลเลอร์ ที่ได้รับการออกแบบมาเป็นพิเศษ เพื่องานประมวลผลสัญญาณดิจิทัลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

ในการใช้งานนั้นจะเลือกรุ่น dsPIC30F6010 เป็นไมโครคอนโทรลเลอร์ขนาด 80 ขา แบบ Thin Quad Flat Pack (TQFP) สาเหตุที่ใช้รุ่นนี้เนื่องจากมีโมดูลของ พัลส์วิดท์มอดูเลชัน ถึง 4 คู่ ซึ่งเท่ากับความต้องการ อีกทั้งหน่วยความจำและจำนวนพอร์ทก็มีให้ใช้มาก เพียงพอกับความ ต้องการอีกด้วย ซึ่งคุณสมบัติต่าง ๆ จะกล่าวอยู่ในภาคผนวก



รูปที่ 2.25 ไมโครคอนโทรลเลอร์ dsPIC30F6010A

2.6.2 หน่วยกำเนิดสัญญาณพัลส์วิดท์มอดูเลชัน

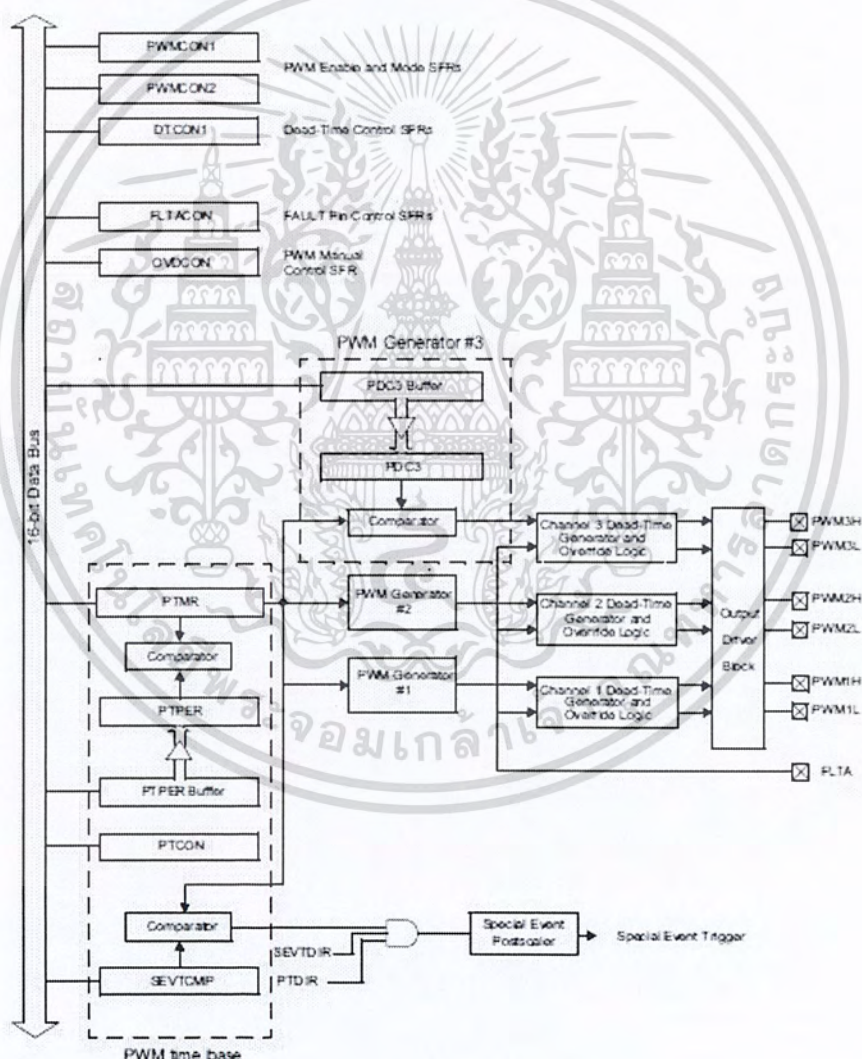
อีกหนึ่งหน่วยฟังก์ชันที่ใช้ในโครงการ คือหน่วยกำเนิดสัญญาณพัลส์วิดท์มอดูเลชัน ซึ่งใน dsPIC มีหน่วยกำเนิดสัญญาณพัลส์วิดท์มอดูเลชัน เพื่อควบคุมมอเตอร์โดยเฉพาะเรียกว่า หน่วย พัลส์วิดท์มอดูเลชัน สำหรับ dsPIC30F6010A มีช่องใช้งาน 6 ช่อง จึงสามารถขับมอเตอร์ไฟฟ้ากระแสตรงได้ 3 ตัว ซึ่งในโครงการนี้จะใช้เพียง 4 ช่องเท่านั้น สำหรับควบคุมมอเตอร์ไฟฟ้ากระแสตรง 2 ตัว

ส่วนประกอบของหน่วยนี้ คือส่วนกำเนิดสัญญาณพัลส์วิดท์มอดูเลชันซึ่งได้ค่าฐานเวลามาจากรีจิสเตอร์ PTMR และ PTPER ในหน่วย MCPWM นี้สามารถกำหนดค่าความกว้างพัลส์หรือค่าดิวตี้ไซเคิล (Duty Cycle) ในส่วนกำเนิดสัญญาณพัลส์วิดท์มอดูเลชัน แต่ละส่วนได้เป็นอิสระต่อกัน นอกจากนี้ยังสามารถกำหนดการทำงานของขาเอาต์พุตของหน่วย MCPWM โดยตรงผ่านทางรีจิสเตอร์ OVDCON สัญญาณที่ออกจากหน่วย MCPWM จะมีขาพอร์ท 2 ขาคู่ช่องนั้น คือขาเอาต์พุตด้านแรงดันต่ำ PWMXL หรือเรียกว่าคู่เอาต์พุตฐานเวลาของสัญญาณพัลส์วิดท์มอดูเลชันของหน่วย MCPWM สามารถกำหนดให้ทำงานได้ 4 โหมด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมดเปลี่ยนแปลงค่าอิสระ
- โหมดทำงานครั้งเดียว
- โหมดนับค่าขึ้นหรือลงอย่างต่อเนื่อง
- โหมดนับค่าขึ้นหรือลงอย่างต่อเนื่องพร้อมการอินเทอร์รัปต์เพื่อปรับปรุงค่า

การเลือกโหมดทำได้โดยการกำหนดค่าที่บิต PTMOD1 และ PTMOD0 ซึ่งเป็น บิต 1 และ 0 ในรีจิสเตอร์ PTCAN ซึ่งในโครงงานนี้จะเลือกใช้โหมดเปลี่ยนแปลงค่าอิสระ ในโหมดนี้ค่าฐานเวลาจะเพิ่มขึ้นจนกระทั่งตรงกับค่าในรีจิสเตอร์ PTER จากนั้นรีจิสเตอร์ PTMR จะรีเซ็ตและทำการนับค่าเพิ่มขึ้นต่อเนื่องไปอีกตราบเท่าที่ บิต PTEN ยังคงมีค่าเป็น '1' อยู่



รูปที่ 2.26 แผนภาพของโมดูล MCPWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนค่าความกว้างพัลส์ ของสัญญาณพัลส์วิดท์มอดูเลตชั้น ของหน่วย MCPWM กำหนดได้จากรีจิสเตอร์ PDC1 - PDC4 ซึ่งต่างก็มีบัพเฟอร์เพื่อป้องกันสัญญาณรบกวนเมื่อมีการปรับปรุ้งค่าของสัญญาณ PWM โดยค่าความกว้างพัลส์ของสัญญาณพัลส์วิดท์มอดูเลตชั้นจะถูกปรับปรุ้งค่าตามข้อมูลที่เขียนลงในรีจิสเตอร์ PDCX จากนั้นค่าจากรีจิสเตอร์ PDCX จะถูกส่งไปยังบัพเฟอร์เพื่อทำการเปรียบเทียบ เมื่อมีการเปลี่ยนแปลงข้อมูลในรีจิสเตอร์ PDCX เรียบบร็อย ข้อมูลนั้นจึงถูกส่งไปยังบัพเฟอร์ข้อมูลนั้นจึงถูกส่งไปยังบัพเฟอร์เพื่อทำงานต่อไป ทำให้ไม่เกิดการติดขัดหรือเกิดความผิดพลาดในขณะที่เปลี่ยนค่าอิสระหรือโหมดทำงานครั้งเดียว ความกว้างพัลส์ของสัญญาณพัลส์วิดท์มอดูเลตชั้น ถูกปรับปรุ้ง เมื่อค่ารีจิสเตอร์ PTMR เท่ากับ PTPER และเมื่อรีจิสเตอร์ PTMR เกิดการรีเซตเป็น '0'

นอกจากนั้นใน MCPWM ยังมีความสามารถพิเศษอีกประการหนึ่ง คือการป้องกันการเปลี่ยนแปลง ค่าความกว้างพัลส์ ทำได้โดยกำหนด บิต UDIS ซึ่งเป็นบิต 2 ของรีจิสเตอร์ PWMCON2 ให้มีค่าเป็น '1'

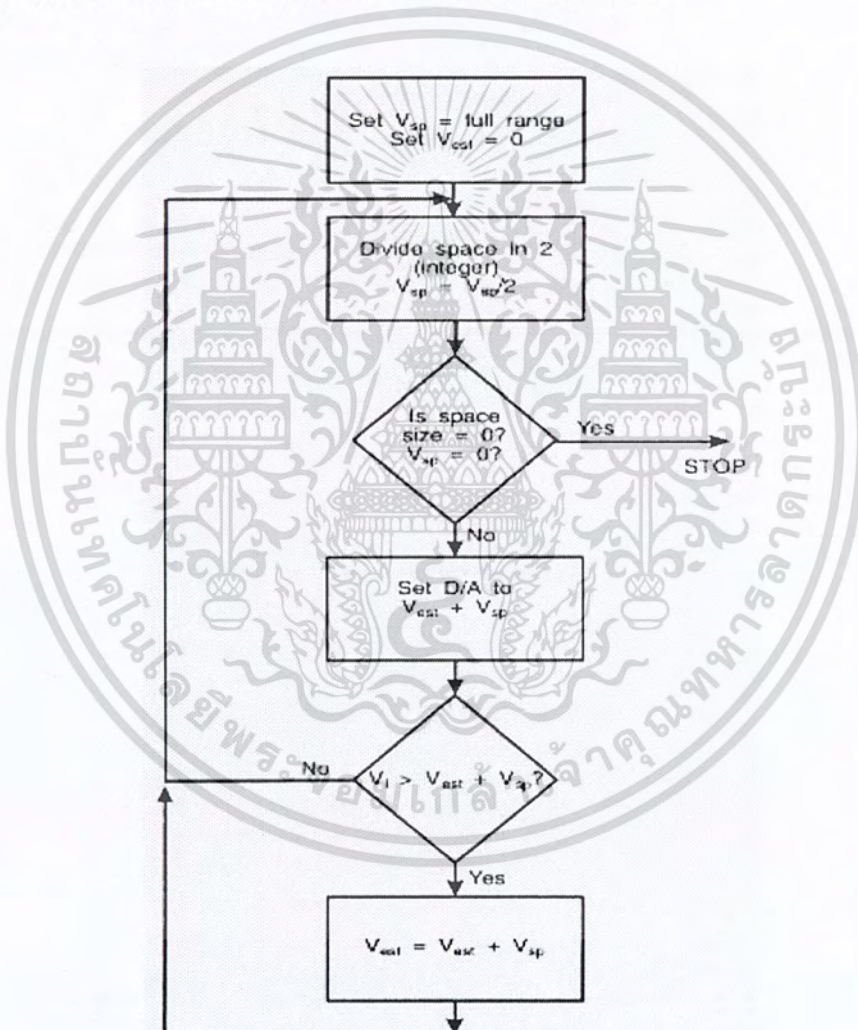
2.7 การแปลงสัญญาณอนาล็อก - ดิจิตอล (Analog to Digital Converter : ADC)

สัญญาณอนาล็อกที่อ่านได้จากเซนเซอร์ใจโรสโคป จะถูกแปลงเป็นสัญญาณดิจิตอลด้วยโมดูล ADC เพื่อที่เราจะสามารถนำค่านั้นไปใช้ในการคำนวณต่อไป ฟังก์ชัน ADC ในตัวคอนโทรลเลอร์เบอร์ dsPIC30F6010A มีขนาดความละเอียดของ ADC ที่ขนาด 10 Bit และ 12 Bit ให้เลือกใช้โดยที่ 10 Bit มีค่าอยู่ระหว่าง 0 - 1023 และ 12 Bit มีค่าอยู่ระหว่าง 0 - 4095

ในการเลือกใช้ความละเอียดของจำนวน Bit ADC เราใช้แบบ 10 Bit ซึ่งมีความเร็วในการคอนเวิร์ต (Converts) ที่สูงเนื่องจากเซนเซอร์ ของเราเป็นใจโรสโคป ซึ่งเอาต์พุต จะมีค่าเป็นสัญญาณอนาล็อก (Analog) และโดยเอาต์พุตของเซนเซอร์จะมีพฤติกรรม คือเมื่อมีการเอียงของตัวเซนเซอร์ก็อาจจะเกิดการเปลี่ยนแปลงของค่าแรงดันไฟฟ้า จากแรงดันอ้างอิงไปจนถึงระดับแรงดันหนึ่งของการเอียงนั้นๆแล้วจากนั้นก็จะสวิงกลับไปยังค่าแรงดันอ้างอิงเดิมแม้ว่าตัวเซนเซอร์เองจะไม่เอียงกลับ ก็คือตัวใจโรสโคป จะวัดได้เฉพาะเวลาเซนเซอร์ เกิดการเคลื่อนไหลขึ้นหรือเซนเซอร์ที่ใช้วัดเป็นแบบพลวัตนั่นเอง ดังนั้นอัตราการเกิดการเปลี่ยนแปลงของสัญญาณเอาต์พุตจากใจโรสโคป นี้จึงมีการแซมปิ้ง (Sampling) ของ ADC ให้มากเข้าไว้ เพื่อไม่ให้เกิดค่าเออเรอร์จากการคอนเวิร์ตสัญญาณ

สัญญาณที่ใช้ในอุปกรณ์อิเล็กทรอนิกส์ มี 2 ชนิด คือสัญญาณอนาล็อกและสัญญาณดิจิตอล สัญญาณอนาล็อกจะใช้ในอุปกรณ์ทั่วไปและใช้ในควบคุมแบบเก่าในปัจจุบันมีไมโครโปรเซสเซอร์และไมโครคอนโทรลเลอร์ เข้ามาช่วยในการควบคุมอุปกรณ์ต่างๆมากมาย ซึ่งทำให้การควบคุมนั้นทำได้ง่ายและรวดเร็วยิ่งขึ้น แต่ในการควบคุมนั้นเราจำเป็นต้องใช้สัญญาณดิจิตอลในการติดต่อกับ

ไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ แต่ในความเป็นจริงนั้นเราใช้สัญญาณอนาล็อกในการควบคุม ดังนั้นเราจึงจำเป็นต้องมีการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล แล้วจึงนำสัญญาณนั้นเข้ามาสู่ไมโครโปรเซสเซอร์หรือไมโครคอนโทรลเลอร์ เพื่อให้ควบคุมระบบต่อไป การแปลงแบบประมาณค่าใกล้เคียงหรือซัคเซสซีฟแอสปรัอกซิเมชัน (Successive Approximation) ใช้หลักการของ “Binary Search” ในการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล โดยนำค่าผลลัพธ์มาเปรียบเทียบกับค่ากึ่งกลางของช่วง เพื่อทราบค่าว่าค่านั้นๆมากกว่าหรือน้อยกว่า โดยจะปรับช่วงให้แคบลงมาเรื่อยๆ แล้วเปรียบเทียบกับผลลัพธ์กับค่ากึ่งกลางของช่วงไปเรื่อยๆ จนได้ผลลัพธ์ที่ต้องการ จากหลักการที่กล่าวมาอาจเขียน Flow Chart ได้ดังนี้



รูปที่ 2.27 แสดงลำดับการทำงานตัวแปลงสัญญาณอนาล็อก – ดิจิตอล

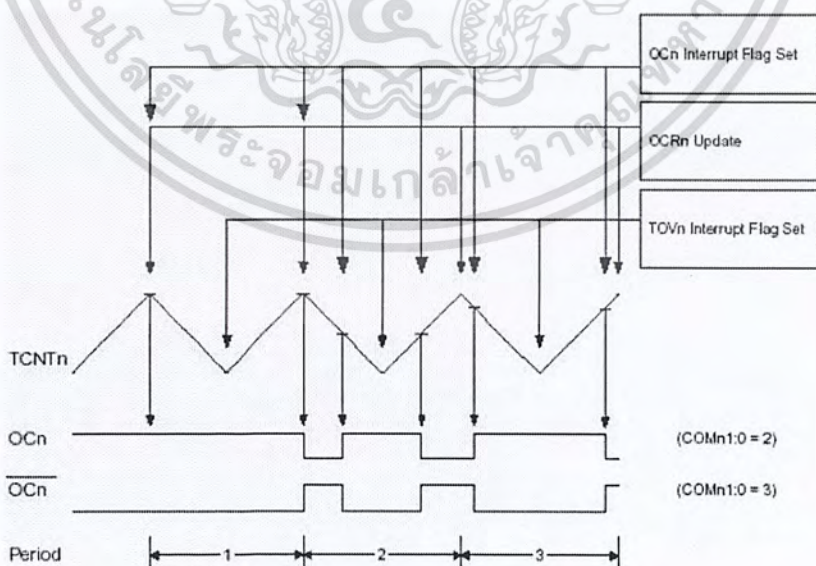
ข้อดีของวิธี ซัคเซสซีฟแอสปรัอกซิเมชัน คือเวลาที่ใช้ในการหาคำตอบ n รอบแน่นอน (สำหรับ n Bit Converter ซึ่งอ้างอิงได้ $2n$ ระดับ และระดับ V_{in} ที่คงที่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 พัลส์วidthที่มอดูเลตขั้้น (Pulse Width Modulation)

พัลส์วidthที่มอดูเลตขั้้น คือหลักการที่ทำให้แรงดันเอาต์พุทของอินเวอร์เตอร์มีลักษณะเป็น สัญญาณ-พัลส์สัเพื่อที่จะสามารถควบคุมระดับความเร็วของมอเตอร์ได้ โดยควบคุมจากแรงดัน โดย จะอาศัยการป้อนกลับค่าแรงดันที่เอาต์พุทและเปรียบเทียบกับแรงดันอ้างอิง (Reference Voltage) ของวงจร เพื่อตรวจจับการเปลี่ยนแปลงของแรงดันที่เอาต์พุท ค่าความแตกต่างที่ได้จะถูกขยายโดย วงจรขยายความแตกต่างก่อนที่จะส่งต่อไปยังวงจร PWM โดยค่าแรงดันที่ได้จากวงจร ขยายความ แตกต่าง จะถูกเปรียบเทียบกับแรงดันรูปฟันเลื่อยของ PWM อีกครั้งหนึ่ง เอาต์พุทที่ได้จากวงจร PWM จะมีลักษณะเป็นพัลส์สัที่เหลี่ยมซึ่งมีคาบเวลาคงที่เท่ากับคาบเวลาของแรงดันรูปฟันเลื่อยและ มีความกว้างของพัลส์สัซึ่งเปลี่ยนแปลงไปตามผลมอดูเลตขั้้นของค่าแรงดันค่าความกว้างของพัลส์สันี้ เองที่จะเป็นตัวกำหนดระดับความเร็วของมอเตอร์

การอ่านค่าของสัญญาณ PWM ใช้ฟังก์ชันอินพุตแคปเจอร์ (Input Capture) ในการจับ สัญญาณขอบขาของ PWM แล้วเทียบกับเวลาเพื่อที่จะหาค่าดิวตี้ไซเคิล (Duty Cycle) ของสัญญาณ PWM จากเอาต์พุท ของตัวแอกเซลเลอโรมิเตอร์นั้นจะมีค่าของดิวตี้ไซเคิล อยู่ระหว่าง 20 – 80 % ดังนั้นการตรวจจับดิวตี้ไซเคิล ของแอกเซลเลอโรมิเตอร์ เราจะหาค่ารอบ (Period) ก่อน เพื่อนำใช้ ในการคำนวณโดยค่า T หาได้จากการนำค่าของผลต่างที่เกิดขึ้นจากการเริ่มนับก็ คือขอบขาขึ้น ของพัลส์ ณ จุด t_1 และขอบขาลงของพัลส์ ณ จุด t_2 นำค่าทั้งสองจุดมาลบกับ ก็จะได้ค่าของผลต่าง และเมื่อได้ผลต่างเกิดขึ้น ก็สามารถที่จะหาค่าดิวตี้ไซเคิลได้ โดยเทียบช่วงเวลาดังแต่เริ่มจนจบ ของพัลส์แต่ละลูกแปลงย้อนกลับมาให้อยู่ในช่วงของดิวตี้ไซเคิล

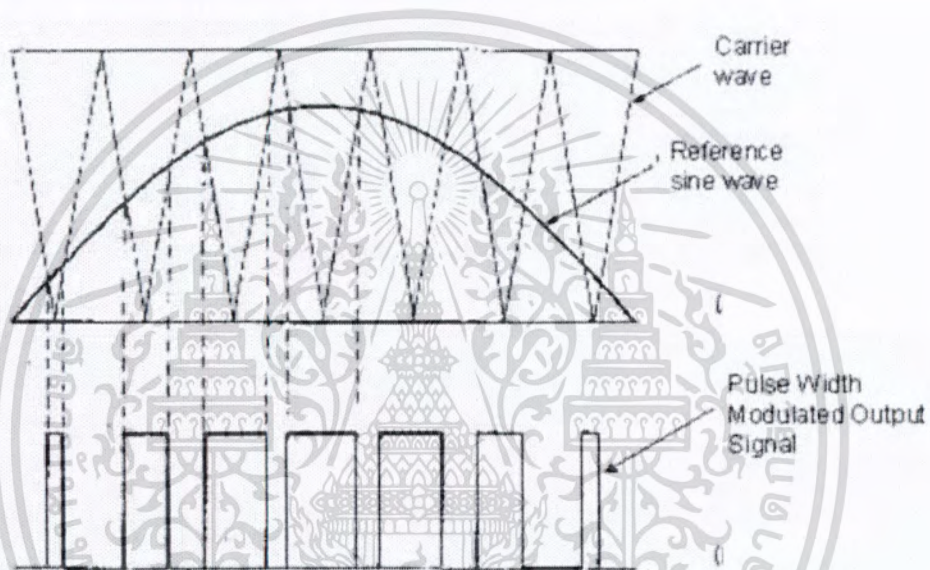


รูปที่ 2.28 แสดงลักษณะความกว้างของพัลส์จาก PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1 หลักการของพัลส์วิดท์มอดูเลชัน

เป็นวิธีการนำสัญญาณสองสัญญาณมา ทำการมอดูเลชันกันแล้วจะได้สัญญาณพัลส์ที่มีพัลส์วิดท์ขนาดต่างๆ กันตัวอย่างเช่น การนำสัญญาณรูปคลื่นไซน์ (Sine Wave) ที่มีความถี่ต่ำมาทำการมอดูเลชันกับสัญญาณรูปคลื่นสามเหลี่ยม (Triangle Wave) ที่มีความถี่สูงจะได้สัญญาณพัลส์ที่มีขนาดพัลส์วิดท์ เปลี่ยนแปลงตามระดับสัญญาณรูปคลื่นไซน์และมีความถี่ตามสัญญาณรูปคลื่นสามเหลี่ยม ดังรูปที่ 2.28 ถ้านำสัญญาณดังกล่าวไปผ่านวงจร โลพาสฟิลเตอร์ (Low Pass Filter) ก็จะได้สัญญาณรูปคลื่นไซน์เหมือนเดิม



รูปที่ 2.29 แสดงการควบคุมลักษณะพัลส์วิดท์มอดูเลชัน

2.8.2 เหตุผลที่เลือกใช้ PWM ในการควบคุมความเร็วมอเตอร์

PWM ง่ายในการอินเทอร์เฟสกับไมโครคอนโทรลเลอร์และใช้เพียงแค่อำนาจที่ชุดสัญญาณเดียวในการควบคุมความเร็ว

PWM มีประสิทธิภาพ คือเพาเวอร์ซัพพลาย (Power Supply) จะจ่ายกำลังได้เต็มที่ทั้งเปิดและปิด (Full On And Full Off)

PWM ทำให้ได้ค่า ทอร์ก (Torque) และความเร็วสูงสุดของมอเตอร์ เป็นเพราะเพาเวอร์ซัพพลายจะจ่ายกำลังได้เต็มที่ทั้ง เปิดและปิด (Full On And Full Off)

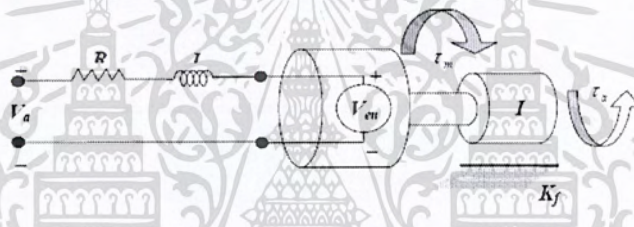
บทที่ 3

การออกแบบระบบควบคุม ไมโครคอนโทรลเลอร์ วงจรมอเตอร์ แบบจำลองทางคณิตศาสตร์ และการออกแบบตัวหุ่นยนต์

3.1 แบบจำลองทางคณิตศาสตร์ของหุ่นยนต์

3.1.1 แบบจำลองทางคณิตศาสตร์ของดีซีมอเตอร์

ในบทนี้จะกล่าวถึงพฤติกรรมของดีซีมอเตอร์ โดยจะนำมาเขียนในลักษณะของแบบจำลองทางคณิตศาสตร์ (Math Model) ดังนี้



รูปที่ 3.1 แบบจำลองโครงสร้างของมอเตอร์

เมื่อใส่โวลต์เตจจากภายนอกเข้ามา จะเกิดกระแสขึ้นในขดลวดอาร์เมเจอร์ โดยที่มอเตอร์จะสร้าง ทอร์ค (T) ขึ้นมา โดยจะเป็นสัดส่วนกับกระแสที่บายความสัมพันธ์ได้ดังนี้

$$\tau_m = k_m I \quad (3.1)$$

ความต้านทาน R และขดลวดตัวนำ L ต่ออนุกรมอยู่กับ V_{emf} (Back Electromotive Force Voltage) เกิดขึ้นเนื่องจากขดลวดของมอเตอร์เคลื่อนที่ผ่านสนามแม่เหล็ก โดยสามารถประมาณค่าให้มีความสัมพันธ์เป็นเชิงเส้นกับอัตราเร็วเชิงมุม (ω) ได้ดังนี้

$$V_e = k_e \omega \quad (3.2)$$

ใช้ กฎของผลรวมของแรงเคลื่อนไฟฟ้าภายในเท่ากับศูนย์ (Kirchoff's Voltage Law) ดังนี้

$$V_a - Ri - L \frac{di}{dt} = 0 \quad (3.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาสมการการเคลื่อนที่ของมอเตอร์ โดยประมาณค่าแรงเสียดทานของมอเตอร์ให้เป็นเชิงเส้นกับอัตราเร็วเชิงมุม โดยที่ I_R คือการเสถียรที่ขดลวดอาร์เมเจอร์ (Inertial Load of Armature) จากกฎของนิวตัน

$$\sum M = \tau_m - k_f \omega - \tau_a = I_R \omega \quad (3.4)$$

แทนสมการ (3.1) และ (3.2) ลงใน (3.3) และ (3.4) และจัดใหม่ให้อยู่ในรูปของอนุพันธ์

$$\frac{d\omega}{dt} = \frac{k_m}{I_R} i - \frac{\tau_a}{I_R} \quad (3.5)$$

$$\frac{di}{dt} = \frac{R}{L} i + \frac{k_e}{L} \omega + \frac{V_a}{L} \quad (3.6)$$

เนื่องจากเป็นดีซีมอเตอร์ไม่คิดแรงเสียดทานที่เกิดขึ้นจากสมการที่ (3.5) และ (3.6) จะได้ว่า

$$i = -\frac{k_e}{R} \omega + \frac{1}{R} V_a \quad (3.7)$$

$$\frac{d\omega}{dt} = \frac{k_m}{I_R} i - \frac{\tau_a}{I_R} \quad (3.8)$$

แทนค่าสมการ (3.7) ลงในสมการ (3.8)

$$\frac{d\omega}{dt} = -\frac{k_m k_e}{I_R R} \omega + \frac{1}{I_R R} V_a - \frac{\tau_a}{I_R} \quad (3.9)$$

สามารถแสดงเป็นสมการปริภูมิสถานะ (State Space Equation) โดยมีตัวแปรที่เราสนใจคือ θ และ ω อินพุทของมอเตอร์ คือ โวลต์เดจและทอร์ค

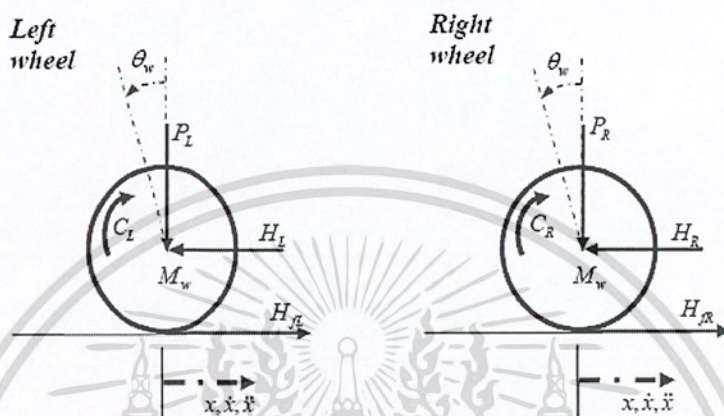
$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{I_R R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_R R} & \frac{-1}{I_R} \end{bmatrix} \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (3.10)$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} V_a \\ \tau_a \end{bmatrix} \quad (3.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 พฤติกรรมของหุ่นยนต์สมดุล

ในการอธิบายพฤติกรรมของหุ่นยนต์สมดุล เราจะทำการแยกพิจารณาออกเป็น 2 ส่วน คือ 1. ส่วนของล้อ 2. ส่วนของเพนดูลัม โดยทั้งสองส่วนนี้สามารถอธิบายพฤติกรรมของหุ่นยนต์สมดุล ได้พิจารณาแผนภาพวัตถุอิสระ (Free Body Diagram) ของล้อทั้ง 2 ข้าง



รูปที่ 3.2 แผนภาพวัตถุอิสระของล้อทั้ง 2 ข้าง

ใช้กฎของนิวตัน โดยพิจารณาผลรวมของแรงแนวแกนนอนในทิศทาง x

$$\begin{aligned}\sum F_x &= Ma \\ M_w \ddot{x} &= H_{fR} - H_{fL}\end{aligned}\quad (3.12)$$

โมเมนต์รอบจุดศูนย์กลางของล้อ

$$\begin{aligned}\sum M_o &= I\alpha \\ I_w \ddot{\theta}_w &= C_R - H_{fR}r\end{aligned}\quad (3.13)$$

จากคี่ซีมอเตอร์และทอร์คสามารถแทนด้วย

$$\tau_m = I_R \frac{d\omega}{dt} + \tau_a \quad (3.14)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนรูปสมการโดยการแทนค่าตัวแปรของดีซีมอเตอร์ โดยทอร์กที่มอเตอร์ให้กับล้อยคือ

$$C = I_R \frac{d\omega}{dt} = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a \quad (3.15)$$

แทนค่า (3.15) ใน (3.13)

$$I_w \ddot{\theta}_w = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a - H_{fR} r \quad (3.16)$$

$$H_{fR} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w \quad (3.17)$$

นำ (3.17) แทนค่าใน (3.12)

ล้อยข้างซ้าย

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_L \quad (3.18)$$

ล้อยข้างขวา

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \dot{\theta}_w + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_R \quad (3.19)$$

เปลี่ยนการเคลื่อนที่เชิงมุมเป็นการเคลื่อนที่เชิงเส้น

ล้อยข้างซ้าย

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x} - H_L \quad (3.20)$$

ล้อยข้างขวา

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x} - H_R \quad (3.21)$$

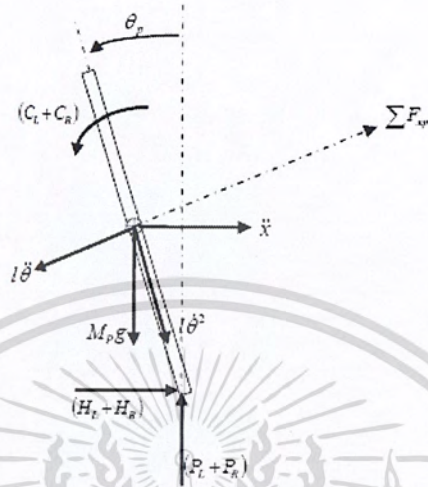
รวมสมการทั้งสองล้อยโดยการบวกสมการ (3.20) เข้ากับ (3.21)

$$2 \left(M_w + \frac{I_w}{r^2} \right) \ddot{x} = \frac{-2k_m k_e}{Rr^2} \dot{x} + \frac{2k_m}{Rr} V_a - (H_L + H_R) \quad (3.22)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 หลักการของอินเวิร์ทเพนดูลัม

พิจารณาส่วนโครงของรถที่เป็นอินเวิร์ทเพนดูลัม พิจารณาจากแผนภาพวัตถุอิสระ



รูปที่ 3.3 แผนภาพวัตถุอิสระของ โครงรถส่วนที่เป็นก้านเพนดูลัม

จากกฎการเคลื่อนที่ของนิวตัน (Newton's Law of Motion) ผลบวกในแรงตามแนวแกน X

$$\sum F_x = M_p \ddot{x}$$

$$(H_L + H_R) - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p = M_p \ddot{x} \quad (3.23)$$

ดังนั้น

$$(H_L + H_R) = M_p \ddot{x} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (3.24)$$

พิจารณาผลบวกของแรงตามแนวแกนตั้งฉากกับเพนดูลัม

$$\sum F_{xp} = M_p \ddot{x} \cos \theta_p$$

$$(H_L + H_R) \cos \theta_p + (P_L + P_R) \sin \theta_p - M_p g \sin \theta_p - M_p l \ddot{\theta}_p = M_p \ddot{x} \cos \theta_p \quad (3.25)$$

พิจารณาผลบวกของโมเมนต์ รอบจุดศูนย์กลางมวล (Central of Mass) ของเพนดูลัม

$$\sum M_o = I \alpha$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p - (C_L + C_R) = I_p \ddot{\theta}_p \quad (3.26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทอร์คที่ให้แก่เพนดูลัมที่ได้จากมอเตอร์ 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)

$$-(H_L + H_R)l \cos \theta_p - (P_L + P_R)l \sin \theta_p - \left(\frac{-2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} \right) = I_p \ddot{\theta}_p$$

$$-(H_L + H_R)l \cos \theta_p - (P_L + P_R)l \sin \theta_p = I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} \quad (3.27)$$

คูณ -1 ในสมการ (3.25)

$$\left[-(H_L + H_R)l \cos \theta_p - (P_L + P_R)l \sin \theta_p \right] + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p l \ddot{x} \cos \theta_p \quad (3.28)$$

แทนค่า (3.27) ลงใน (3.28) จะได้

$$I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p l \ddot{x} \cos \theta_p \quad (3.29)$$

จากสมการที่ (3.24) แทนค่าลงใน (3.22)

$$2 \left(M_w + \frac{I_w}{r^2} \right) \ddot{x} = \frac{-2k_m k_e \dot{x}}{Rr^2} + \frac{2k_m V_a}{Rr} - M_p \ddot{x} - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \ddot{\theta}_p^2 \sin \theta_p \quad (3.30)$$

จัดสมการ (3.29) และ (3.30) ทั้ง 2 สมการไม่เป็นเชิงเส้น (Non-Linear Equations)

$$(I_p + M_p l^2) \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} + M_p g l \sin \theta_p = -M_p l \ddot{x} \cos \theta_p \quad (3.31)$$

$$\frac{2k_m V_a}{Rr} = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e \dot{x}}{Rr^2} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \ddot{\theta}_p^2 \sin \theta_p \quad (3.32)$$

เมื่อให้ $\theta_p = \pi + \theta$ เมื่อ θ คือมุมที่มีการเปลี่ยนแปลงน้อยมากและมีทิศทางไปด้านหน้า

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

ทำสมการให้เป็น สมการเชิงเส้น

$$(I_p + M_p l^2) \ddot{\theta} - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m V_a}{R} - M_p g l \theta = M_p l \ddot{x} \quad (3.33)$$

$$\frac{2k_m V_a}{Rr} = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e \dot{x}}{Rr^2} - M_p l \ddot{\theta} \quad (3.34)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดสมการ (3.33) และ (3.34) ใหม่

$$\ddot{\theta} = \frac{M_p l}{(I_p + M_p l^2)} \ddot{x} + \frac{2k_m k_e}{Rr(I_p + M_p l^2)} \dot{x} - \frac{2k_m}{R(I_p + M_p l^2)} V_a + \frac{M_p g l}{(I_p + M_p l^2)} \theta \quad (3.35)$$

$$\ddot{x} = \frac{2k_m}{Rr \left(2M_w + \frac{2I_w}{r^2} + M_p \right)} V_a - \frac{2k_m k_e}{Rr^2 \left(2M_w + \frac{2I_w}{r^2} + M_p \right)} \dot{x} + \frac{M_p l}{\left(2M_w + \frac{2I_w}{r^2} + M_p \right)} \ddot{\theta} \quad (3.36)$$

นำสมการ (3.35) แทนใน (3.34) และแทนค่า (3.36) ใน (3.33) จะได้ จากนั้นเขียนเป็นสมการปริภูมิสถานะ จะได้

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{21} \\ 0 \\ B_{41} \end{bmatrix} V_a$$

เมื่อ

$$A_{22} = \frac{2k_m k_e (M_p I_r - I_p - M_p l^2)}{Rr^2 \alpha}, A_{23} = \frac{M_p^2 g l^2}{\alpha},$$

$$A_{42} = \frac{2k_m k_e (r\beta - M_p l)}{Rr^2 \alpha}, A_{43} = \frac{M_p g l \beta}{\alpha},$$

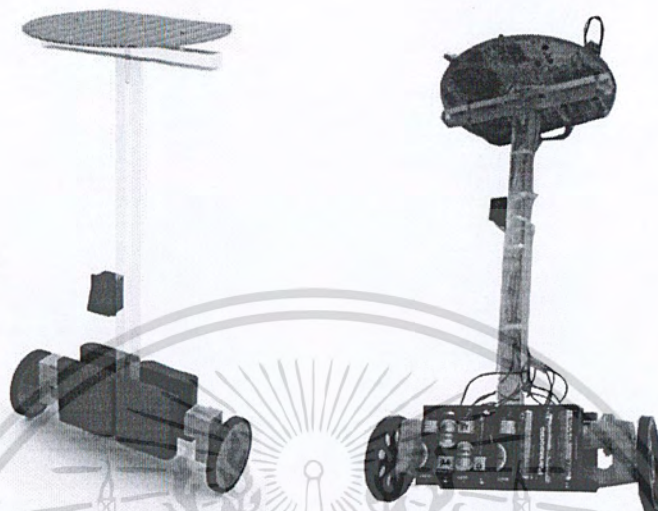
$$B_{21} = \frac{2k_m (I_p + M_p l^2 - M_p I_r)}{Rr \alpha}, B_{41} = \frac{2k_m (M_p l - r\beta)}{Rr \alpha}$$

และ

$$\beta = \left(2M_w + \frac{2I_w}{r^2} + M_p \right), \alpha = \left[I_p \beta + 2M_p l^2 \left(M_w + \frac{I_w}{r^2} \right) \right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบควบคุม



รูปที่ 3.4 หุ่นยนต์ขับเคลื่อนสองล้อแบบสมมูลโดยใช้หลักการอินเวอร์ทเพนดูลัม

จากทฤษฎีของออปติมอลคอนโทรลและโมเดลทางคณิตศาสตร์สามารถหาค่าพารามิเตอร์ต่างๆ ได้ดังต่อไปนี้

ตารางที่ 3.1 ค่าพารามิเตอร์ต่างๆ ของตัวหุ่นยนต์

V_a	แรงดันที่ป้อนให้ (V)
k_m	ค่าคงที่ของแรงบิด ($0.4528 \text{ N} \cdot \text{m/A}$)
k_e	ค่าคงที่ของแรงเคลื่อนไฟฟ้าย้อนกลับ ($0.5674 \text{ V} \cdot \text{sec/rad}$)
M_p	มวลของเพนดูลัม (0.8 kg)
l	ความยาวของก้าน Pendulum ที่จุดศูนย์กลางมวล (0.14 m)
r	รัศมีของวงล้อ (0.035 m)
I_p	โมเมนต์ความเฉื่อยของเพนดูลัม ($0.008 \text{ kg} \cdot \text{m}^2$)
g	ค่าคงที่แรงโน้มถ่วงของโลก (9.81 m/s^2)
R	ค่าความต้านทาน (14.5 Ohm)
M_w	มวลของวงล้อทั้งสองที่เชื่อมต่อกันระหว่างเพนดูลัม (0.03 kg)
I_w	โมเมนต์ความเฉื่อยของวงล้อ ($0.000018 \text{ kg} \cdot \text{m}^2$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากค่าในตารางเราจะใช้โปรแกรมเมทแลปเข้ามาช่วยในการคำนวณเพื่อหาค่าพารามิเตอร์ต่างๆได้ดังนี้

$$\text{OLPoles} = \quad 0 \quad -67.9979 \quad 7.8615 \quad -6.9813$$

เราจะเห็นได้ว่า ระบบนี้ไม่เสถียร เพราะมีโพล 1 ตัว อยู่ที่ซีกครึ่งขวาที่

$$s = 7.8615$$

ดังนั้น สมการปริภูมิสถานะและสมการปริมาณออก (Output Equations) แบบต่อเนื่องทางเวลา (Continuous Time) สามารถเขียนได้เป็น

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -67.1177 & 14.4489 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -274.6916 & 114.7379 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 4.1401 \\ 0 \\ 16.9443 \end{bmatrix} V_a$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x \\ \theta \end{bmatrix}$$

3.2.1 การออกแบบแอสทิวอาร์ (LQR)

การออกแบบ LQR นั้น จะเป็นการหา $u = Fx$ หรือหาอัตราขยายสแตตป้อนกลับ F เพื่อให้ได้ $J = \frac{1}{2} \sum_{k=0}^{\infty} [x^T Q x + u^T R u]$ มีค่าต่ำที่สุดที่เราได้ให้น้ำหนัก ฟังก์ชันถ่วงน้ำหนัก Q, R ไว้แล้ว กรณีที่ง่ายที่สุด ในที่นี้ เราจะให้ $R = 1$ และ

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

เมื่อเรรันในโปรแกรมเมทแลป แล้วจะได้ผลลัพธ์ของ F ดังนี้

$$F = [1.0000 \quad 32.7080 \quad -70.1292 \quad -8.9235]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถหาโพลลูปิด (Closed-Loop System) ของเมทริกซ์ $H = A - B_2F$

$$H = \begin{bmatrix} 0 & 0.0010 & 0 & 0 \\ 0.0041 & 0.0683 & -0.2759 & -0.0369 \\ 0 & 0 & 1 & 0.0010 \\ 0.0169 & 0.2795 & -1.0736 & -0.1512 \end{bmatrix}$$

เราจะได้โพลลูปิดอยู่ที่ $S = -67.9975, -0.0617, -7.8801$ และ -6.9648

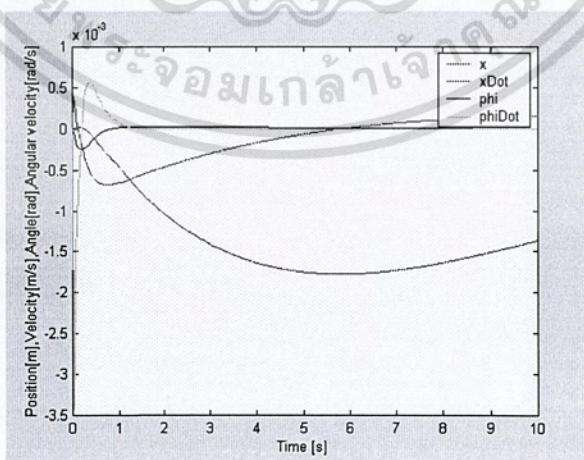
3.2.2 การออกแบบ LQE

ในการทำงานเดียวกันเราสามารถใส่ระบบคู่ $[(A, B_2, C_2) \longleftrightarrow (A^T, C_2^T, B_2^T)]$ จะได้อัตรา
การขยายตัวสังเกต (Observer Gain) L จากนั้นเราจะเขียน M-File เพื่อหาค่า L ได้ดังนี้

$$L = \begin{bmatrix} -1.0074 & -0.3416 \\ -0.0658 & -3.0330 \\ -0.3416 & -15.7486 \\ -2.6910 & -123.5669 \end{bmatrix}$$

โพลลูปิดของระบบตัวสังเกต โดยโพลของ $(A + LC_2)$ จะอยู่ที่ $s_1 = -1.0000$, $s_2 = -8.0988$,
 $s_3 = -6.7767$, $s_4 = -67.9981$

เมื่อทำการออกแบบ LQR และ LQE เพื่อหาค่าอัตราขยาย F (LQR) และอัตราขยาย L (LQE) จะเห็นว่าระบบอยู่ในเสถียรภาพเนื่องจากโพลลูปิดอยู่ทางซีกซ้ายของแกนจริงซึ่ง
ผลตอบสนองต่ออิมพัลส์ (Impulse) เป็น ไปดังภาพ



รูปที่ 3.5 ผลตอบสนองของระบบเมื่อกำหนด $q_{11} = 1$ กับ $q_{33} = 1$ และ $R = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองของระบบจะขึ้นอยู่กับค่าน้ำหนักของเมตริก Q และเมตริก R โดย

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

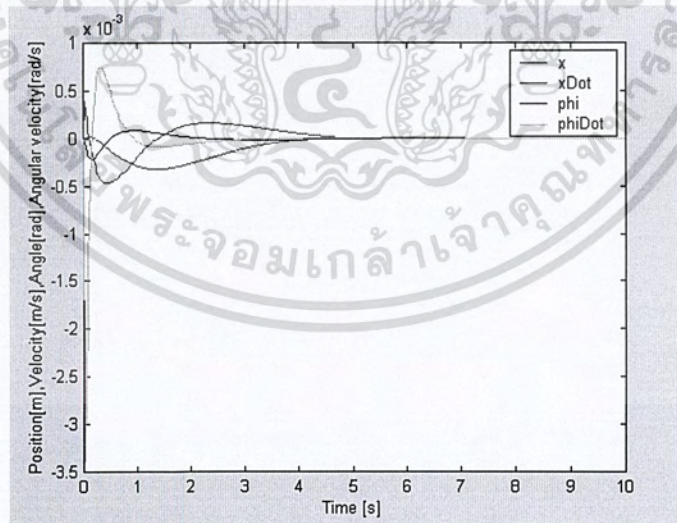
โดยที่ q_{11} จะเป็นตัวน้ำหนักของ X (Position)

q_{33} จะเป็นตัวน้ำหนักของ θ (Angular)

$$R = 1$$

เมื่อเราทำการเปลี่ยนแปลงค่า Q และ R เป็นค่าต่างจะพบว่าผลตอบสนองของระบบจะเปลี่ยนไป

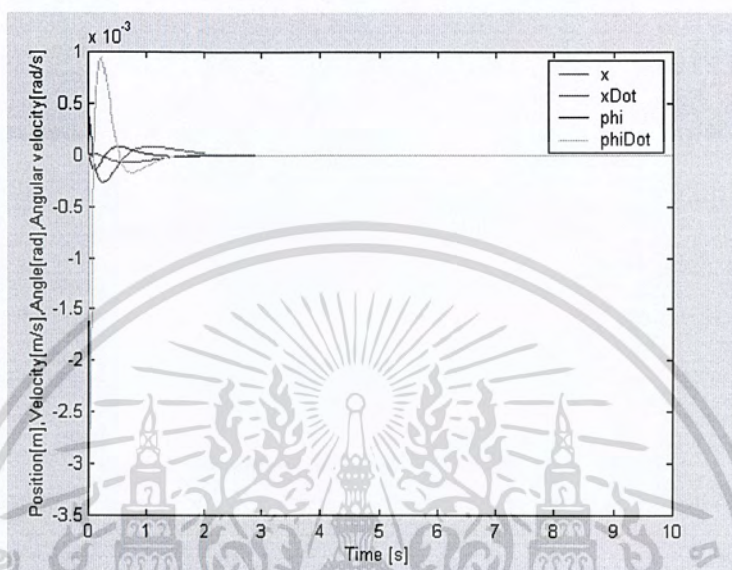
ค่าตัวน้ำหนักของเมตริก Q เป็นตัวบ่งชี้ถึงความสำคัญของตัวแปรสแตจ (State Variable) เมื่อค่าในเมตริก Q ของตัวแปรสแตจใดมีค่าสูงแสดงว่าเราให้ความสำคัญต่อผลตอบสนองของตัวแปรสแตจนั้นสูง ส่วนค่าในเมตริก Q ของตัวแปรสแตจที่มีค่าต่ำบ่งบอกถึงว่าเราให้ความสนใจต่อผลตอบสนองแก่ตัวแปรสแตจนั้นน้อย



รูปที่ 3.6 ผลตอบสนองเมื่อทำการเปลี่ยนแปลงค่าน้ำหนัก $q_{11} = 1000$ กับ $q_{33} = 1$ และ $R = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการเปรียบเทียบกันระหว่าง รูปที่ 3.6 กับรูปที่ 3.7 โดยการเพิ่มค่า q_{11} จาก 1 เป็น 1000 โดย q_{11} คือค่าน้ำหนักในเมตริก Q ที่เป็นน้ำหนักของตัวแปรสแดง X (Position) จากเส้นกราฟสีน้ำเงินจะเห็นได้ว่าช่วงเวลาอยู่ตัว (Setting Time) มีค่าลดลง

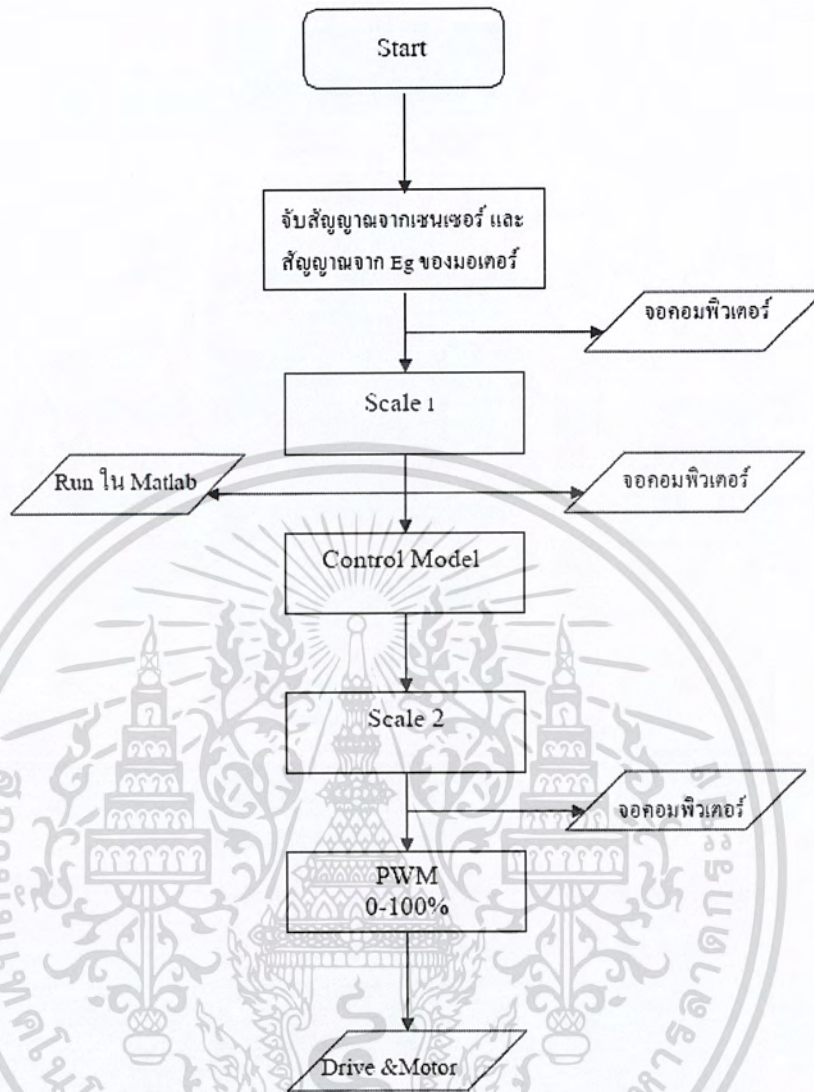


รูปที่ 3.7 ผลตอบสนองเมื่อค่าน้ำหนักเป็น $q_{11}=1000$ กับ $q_{33}=1000$ และ $R=1/100$

การลดค่าของ R ลงจะทำให้ช่วงเวลาอยู่ตัว ลดลงเนื่องจากโพลของระบบออกห่างจากแกนจินตภาพ ทำให้ระบบเร็วขึ้นและการเพิ่มค่าน้ำหนักของเมตริก Q ทำให้ช่วงเวลาอยู่ตัวลดลงก็เนื่องมาจากเหตุผลเดียวกับการลดค่า R คือทำให้โพลวิ่งออกห่างจากแกนจินตภาพ

3.3 การออกแบบโปรแกรมบนไมโครคอนโทรลเลอร์

ในส่วนของการเขียนโปรแกรมนั้น จะใช้ภาษาซีในการเขียนและใช้โปรแกรมเอ็มพีแลปเป็นตัวคอมไพล์ การเขียนโปรแกรมทั้งหมดนี้จะอยู่ภายใต้แบบจำลองทางคณิตศาสตร์ที่ได้ศึกษามาแล้วทั้งหมด เพื่อให้การควบคุมมีประสิทธิภาพที่สุด ซึ่งการออกแบบโปรแกรมนั้นจะเริ่มตั้งแต่มีสัญญาณเข้า คำนวณในโมเดลทางคณิตศาสตร์ จนกระทั่งได้สัญญาณเอาต์พุตออกมา ซึ่งจะนำไปตามแผนภาพโดยรวมดังต่อไปนี้



รูปที่ 3.8 แผนภาพโดยรวมของการเขียน โปรแกรม

จากแผนภาพการทำงานของไมโครคอนโทรลเลอร์จะเริ่มด้วยการจับสัญญาณจากเซนเซอร์ 2 แหล่งด้วยกันคือ 1. ใจโรสโคป ซึ่งเป็นตัววัดความเร็วเชิงมุม สัญญาณที่ได้จะเป็นพัลส์วิดท์มอดูเลตชั้นที่ความถี่ประมาณ 5 KHz หรือ 0.20 ms เพื่อให้มีความละเอียดและความไวต่อการจับระดับสัญญาณจิงไทเมอร์แคปเจอร์ (Timer Capture) 32 Bit ซึ่งเป็นความเร็วที่สูงที่สุดของการจับสัญญาณ แหล่งที่ 2 คือแรงดันย้อนกลับของมอเตอร์ (Eg) ซึ่งจะถูกนำมาหาค่าความเร็วเชิงมุมจากโมเดลทางคณิตศาสตร์ของมอเตอร์ ค่าความเร็วนี้จะถูกนำมาคำนวณหาระยะทาง ที่รถเคลื่อนที่ตอนนี้แสดงผลบนหน้าจอกอมพิวเตอรืเพื่อง่ายต่อการสังเกต จากนั้นนำตัวเลขของการจับสัญญาณที่ได้ มาทำสเกลให้อยู่ในรูปของความเร็วเชิงมุม (rad/s) และระยะทาง (m)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเลขที่ได้จากขั้นตอนแรกจะถูกนำมาใช้ในการคำนวณ ในโมเดลทางคณิตศาสตร์ของตัวคอนโทรลที่ได้จากการออกแบบ จากนั้นก็จะได้ค่าสัญญาณควบคุม (Manipulate Value : MV)

ค่าตัวเลขของค่า MV ที่ได้จะถูกสเกลอีกครั้งให้เป็น 0 - 100 % เพื่อจะส่งไปให้ส่วนของโปรแกรมที่เขียนเพื่อสร้าง PWM เพื่อให้ได้ความถี่ 0-100% และความถี่ประมาณ 1 KHz ขึ้นไป ขั้นตอนสุดท้ายของโปรแกรมคือการส่งสัญญาณ PWM 4 คู่ ซึ่งแต่ละคู่จะทำงานเป็นแบบโหมคคอมพลิเมนต์ารี (Complementary Mode) คือมีด้านสูง (High) และด้านต่ำ (Low) ซึ่งจะมีค่าเวลาดายของคู่ PWM เดดไทม์ (Dead time) มากกว่า 200 นาโนวินาทีขึ้นไป มอเตอร์ 1 ตัวจะใช้ PWM 2 คู่ เพื่อที่จะควบคุมได้ทั้งความเร็วและทิศทางของมอเตอร์ ซึ่งรายละเอียดของวงจรที่ขับมอเตอร์และมอเตอร์คอนโทรลจะกล่าวในหัวข้อต่อไป สำหรับโปรแกรมที่เขียนผู้ที่สนใจสามารถดูได้ที่ภาคผนวก

3.4 การออกแบบวงจรขับมอเตอร์

ในการออกแบบวงจรขับมอเตอร์อาศัยหลักการขับมอเตอร์แบบ 4 ควอดแรนและคำนึงถึงกระแสกับแรงดันทางด้านเอาต์พุตซึ่งในที่นี้เราจำกัดกระแสไว้ไม่เกิน 15 แอมแปร์และแรงดันอยู่ที่ -12 ถึง 12 โวลต์ สำหรับการทำงานของวงจรนี้จะรับสัญญาณอินพุต แบบพัลส์วิดมอดดูเลชัน ความถี่ประมาณ 2.5 KHz ซึ่งเป็นสัญญาณที่ได้จากไมโครคอนโทรลเลอร์โดยค่าความถี่จะขึ้นอยู่กับผลของการคำนวณทางคณิตศาสตร์ในตัวไมโครคอนโทรลเลอร์สัญญาณพัลส์วิด ที่ออกมา 1 คู่ จะประกอบไปด้วยสัญญาณแบบสูงและต่ำซึ่งสัญญาณทั้งสองนั้นจะเป็นแบบคอมพลิเมนต์ารีมีเดดไทม์ ประมาณ 200 ns

สัญญาณทั้งสองจะถูกส่งเข้าไปที่ตัวเกทไดร์ฟ (Gate Drive) เพื่อขยายสัญญาณทางด้านเอาต์พุต จะได้ สูงและต่ำ เพื่อไปสั่งการทำงานของตัวสวิตซ์ซึ่ง ซึ่งในโปรเจกต์นี้ใช้มอสเฟต เนื่องจากความสามารถในการทนต่อกระแส แรงดันและสามารถสวิตซ์ที่ความถี่สูงๆได้

มอเตอร์ 1 ตัวใช้สัญญาณพัลส์วิด 2 คู่ เพื่อให้สามารถเลือกทิศทางของกระแสได้แรงดันที่ป้อนให้มอเตอร์นั้นจะอยู่ที่ -12 ถึง 12 โวลต์ ซึ่งตัวที่จะกำหนดขนาดของแรงดันที่จะป้อนให้มอเตอร์ก็คือสัญญาณพัลส์วิดที่ปรับค่าความถี่ได้ตั้งแต่ 0 - 100% โดยที่

$$V_a = T_{on}/(T_{off} + T_{on}) * V_{battery} = \text{duty cycle} * V_{battery}$$

3.5 การจำลองการทำงานของตัวหุ่นยนต์ขับเคลื่อนแบบสองล้อ

ในการออกแบบจำลองการทำงานของหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุจะใช้โปรแกรมแลปวิว (LabVIEW) ในการจำลองการทำงานของหุ่นยนต์ ที่สามารถเปลี่ยนแปลงค่าต่างๆได้ เพื่อที่จะดูผลตอบสนองของตัวหุ่นยนต์โดยแสดงเป็นภาพสามมิติ

แต่ก่อนที่จะมาออกแบบและทำการจำลองการทำงานของหุ่นยนต์ขับเคลื่อนสองล้อแบบสมดุ จะขอกล่าวถึงการใช้งาน โปรแกรมแลปวิวและเพราะเหตุใดจึงเลือกใช้โปรแกรมนี้

3.5.1 การใช้งานโปรแกรมแลปวิว

อันดับแรกแลปวิว เป็นโปรแกรมคอมพิวเตอร์ที่สร้าง เพื่อนำมาใช้ในด้านการวัดและเครื่องมือวัดสำหรับงานทางวิศวกรรม แลปวิว ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่ง หมายความว่า เป็นโปรแกรมที่สร้าง เครื่องมือวัดเสมือนจริงในห้องปฏิบัติการทางวิศวกรรม ดังนั้นจุดประสงค์หลักของการทำงานของโปรแกรมนี้ก็คือการจัดการในด้านการวัดและเครื่องมือวัด อย่างมีประสิทธิภาพและในตัวของโปรแกรมจะประกอบไปด้วยฟังก์ชันที่ช่วยในการวัดมากมายและแน่นอนที่สุด โปรแกรมนี้จะมีประโยชน์อย่างสูงเมื่อใช้ร่วมกับเครื่องมือวัดทางวิศวกรรมต่างๆ

สิ่งที่แลปวิว แตกต่างจากโปรแกรมอื่นอย่างเห็นได้ชัดที่สุดก็คือแลปวิวนี้เป็น โปรแกรมประเภท GUI (Graphic User Interface) โดยสมบูรณ์ นั่นคือเราไม่จำเป็นต้องเขียน โค้ด (Code) หรือคำสั่งใดๆ ทั้งสิ้นและที่สำคัญลักษณะภาษาที่ใช้ในโปรแกรมนี้เราจะเรียกว่าเป็น ภาษารูปภาพหรือเรียกอีกอย่างว่าภาษา G (Graphical Language) ซึ่งจะแทนการเขียน โปรแกรมเป็นบรรทัดอย่างที่เรารู้คุ้นเคยกับภาษาพื้นฐาน เช่น ซี, เบสิก (Basic) หรือ FORTRAN ด้วย รูปภาพหรือสัญลักษณ์ทั้งหมดซึ่งแม้ว่าในเบื้องต้นเราอาจจะสับสนกับการจัดเรียงหรือเขียน โปรแกรมบ้าง แต่เมื่อเรารู้คุ้นเคยกับการใช้โปรแกรมนี้แล้ว เราจะพบว่าแลปวิวนี้ มีความสะดวกและสามารถลดเวลาในการเขียนโปรแกรมลงไปได้มาก โดยเฉพาะในงานเขียน โปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อกับอุปกรณ์อื่นๆ เพื่อใช้ในการวัดและการควบคุม

สำหรับผู้ที่เคยใช้โปรแกรมประเภทที่ใช้ตัวหนังสือหรือที่เรียกว่า เท็กซ์เบส (Text Base)ทั้งหลาย คงจะทราบถึงความยุ่งยากในการจัดการกับตำแหน่งการส่งผ่านข้อมูลตามอุปกรณ์เชื่อมต่อ เช่นพอร์ตหรือการ์ด (Card) ต่างๆ รวมถึงการจัดวางตำแหน่งในหน่วยความจำเพื่อที่จะสามารถรวบรวมข้อมูลมาใช้ในการคำนวณและเก็บข้อมูลให้ได้ประโยชน์สูงสุด ปัญหาเหล่านี้ได้รับการแก้ไขในแลปวิว โดยได้มีการบรรจุโปรแกรมจำนวนมากหรือไลบรารี (Library) ไว้สำหรับจัดการกับปัญหาเหล่านั้น ไม่ว่าจะอุปกรณ์การเชื่อมต่อจะเป็น DAQ (Data Acquisition), GPIB (General Purpose Interface Bus หรือก่อนหน้านี้รู้จักกันในชื่อ Hewlett Packard Interface Bus, HP-IB), พอร์ตอนุกรม

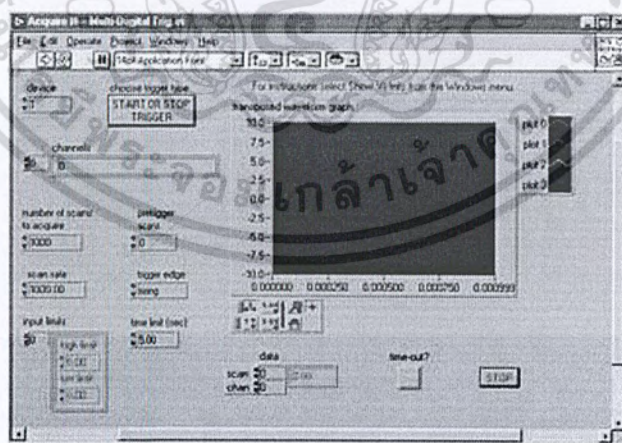
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Serial Port) เพื่อใช้ติดต่อกับอุปกรณ์ที่ส่งผ่านข้อมูลแบบอนุกรม (Serial Instrument) รวมถึงการวิเคราะห์ข้อมูลที่ได้ด้วยวิธีการต่างๆ นอกจากนี้ในโรบารี เหล่านี้ยังได้บรรจุฟังก์ชันการทำงานที่สำคัญอีกหลายประการเช่น Signal Generation, Signal Processing, Filters, สถิติ, พีชคณิตและคณิตศาสตร์อื่นๆ ดังนั้นแลปวิวจึงทำให้การวัดและการใช้เครื่องมือวัดกลายเป็นเรื่องง่ายลงไปมากและทำให้คอมพิวเตอร์ส่วนบุคคลกลายเป็นเครื่องมือทางด้านกรวัดหลายชนิดอยู่ในเครื่องเดียว

โปรแกรมที่เขียนขึ้นมาโดยแลปวิวจะเรียกว่า Virtual Instrument (VI) เพราะลักษณะที่ปรากฏทางจอภาพเมื่อผู้ใช้ใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรมในขณะเดียวกันหลังจากของอุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของฟังก์ชัน, Subroutines และโปรแกรมหลักเหมือนกับภาษาทั่วไป สำหรับ VI หนึ่งๆ จะประกอบด้วยส่วนประกอบที่สำคัญสามส่วนคือ

3.5.2 แผงควบคุมด้านหน้า (Front Panel)

แผงควบคุมด้านหน้า จะเป็นส่วนที่ใช้สื่อความกันระหว่างผู้ใช้กับโปรแกรม (หรือที่นิยมเรียก User Interface) โดยทั่วไปจะมีลักษณะเหมือนกับหน้าปัทม์ของของเครื่องมือหรืออุปกรณ์ที่ใช้งานด้านการวัดต่างๆ ไปด้วย โดยทั่วไปจะประกอบด้วย สวิตช์ปิดเปิด, ปุ่มบิด, ปุ่มกด จอแสดงผลหรือแม้แต่ค่าที่ผู้ใช้สามารถกำหนด ดังนั้นสำหรับผู้ที่ยังคุ้นเคยกับการเขียน โปรแกรมประเภทวิซวล (Visual) ทั้งหลายคงจะเข้าใจถ้าหากบอกว่าแผงควบคุมด้านหน้า นี้จะเปรียบเสมือนเป็น GUI ของโปรแกรมหรือ VI นั่นเอง



รูปที่ 3.9 แสดงแผงควบคุมด้านหน้าของโปรแกรมแลปวิวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

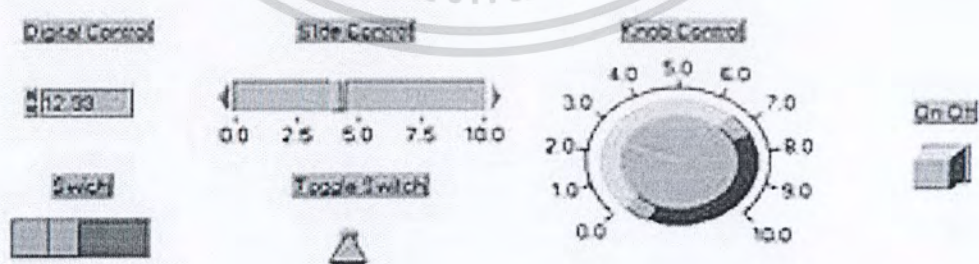
ในขณะที่เราให้ VI ทำงานอยู่นั้นหน้าปัทม์นี้จะต้องทำงานร่วมอยู่ด้วย เพื่อให้ผู้ใช้หรือผู้ควบคุมสามารถให้ข้อมูลเข้าสู่โปรแกรมและเมื่อข้อมูลได้รับการประมวลผลแล้ว ก็จะแสดงออกมาทางแผงควบคุมด้านหน้านี้ ดังนั้นหากจะเปรียบกับ โปรแกรมสำเร็จรูปอื่นๆ และแผงควบคุมด้านหน้า นี้ก็คือ Graphic User Interface (GUI) ของแลปวิว นั่นเอง ตัวอย่างของลักษณะของแผงควบคุมด้านหน้า ในแลปวิว เป็นไปตามรูป ซึ่งในขั้นแรกนี้ผู้ที่ยังไม่มีความคุ้นเคยกับโปรแกรมนี้อาจมองว่าการ สร้างองค์ประกอบต่างๆ จะมีความยุ่งยาก แต่ถ้าหากเราเริ่มทำความเข้าใจกับแลปวิว แล้วเราจะพบว่า การเขียนแผงควบคุมด้านหน้า ในลักษณะในรูปนี้ไม่ใช่สิ่งที่ยุ่งยากหรือสิ้นเปลืองเวลาในการเขียนเลย

แผงควบคุมด้านหน้า ของแลปวิว จะมีส่วนประกอบที่สำคัญ 2 แบบ คือตัวควบคุม (Controlled) และตัวแสดงผล (Indicator) ซึ่งส่วนประกอบทั้ง 2 จะมีการทำงานต่างกันและหน้าที่ตรงกันข้ามกัน ดังมีรายละเอียดต่อไปนี้

- ส่วนควบคุม (Controls)

ส่วนควบคุม มีหน้าที่เป็นตัวควบคุมคือให้ค่าหรืออินพุตจากผู้ใช้ ลักษณะของส่วนควบคุม เช่น ปุ่มปรับค่า, สะพานเปิด – ปิดไฟ, แท่งเลื่อนเพื่อปรับค่า, การให้ค่าด้วยตัวเลขดิจิทัล (Digital) หรืออื่นๆ ดังนั้นจากหลักการของส่วนควบคุม ก็หมายความว่า เป็นการกำหนดค่าหรือแหล่ง (Source) ของข้อมูล โดยปกติเราจะไม่สามารถนำข้อมูลมาแสดงผลที่ ควบคุมได้และถ้าหากเราพยายามที่จะให้ ส่วนควบคุม แสดงผลข้อมูลก็จะเกิดความผิดพลาดขึ้นใน VI ของเราทันที

ตัวอย่างของวัตถุ (Object) ที่ปกติแล้วจะทำหน้าที่เป็นส่วนควบคุม บนแผงควบคุมด้านหน้า บางประเภท จะแสดงในภาพที่ 3.10 เราจะสังเกตเห็นว่าหากเปรียบเทียบกับในอุปกรณ์เครื่องมือวัดจริงแล้ว อุปกรณ์เหล่านี้จะได้รับการกำหนดค่าจากผู้ใช้ ดังนั้นจะเห็นว่า แลปวิว พยายามทำให้เราได้รู้สึกว่าการใช้งานกับเครื่องมือจริงๆ อยู่



รูปที่ 3.10 แสดงส่วนควบคุมของโปรแกรมแลปวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแสดงผล (Indicators)

ตัวแสดงผล มีหน้าที่เป็นตัวแสดงผลเพียงอย่างเดียว โดยจะรับค่าที่ได้จากแหล่งข้อมูลมา แสดงผลซึ่งอาจปรากฏในรูปของกราฟ, เข็มชี้, ระดับของเหลวหรืออื่นๆ ตัวแสดงผลนี้เปรียบเสมือนเอาต์พุต เพื่อให้ผู้ใช้ได้ทราบค่าสิ่งที่เรากำลังวิเคราะห์ห้อยู่และผู้ใช้ไม่สามารถปรับค่าบนตัวแสดงผล ได้โดยตรงแต่จะต้องมีแหล่งข้อมูลที่ส่งให้กับตัวแสดงผลเหล่านี้ ดังนั้นเราอาจมองตัวแสดงผลว่าเป็นเหมือนซิงค์ (Sink) ของข้อมูล ตัวอย่างของวัตถุที่ปกติแล้วจะมีเป็นตัวแสดงผลบางชนิดได้แสดงในรูปต่อไปนี้

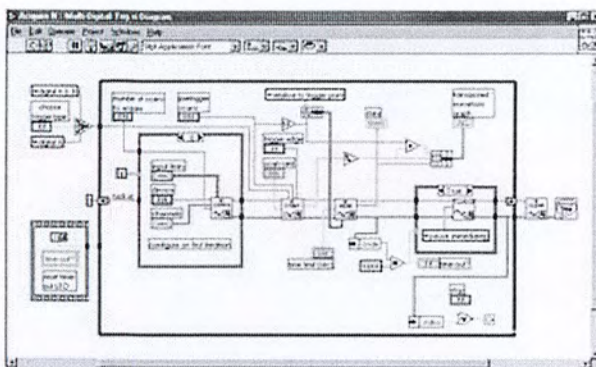


รูปที่ 3.11 แสดงตัวแสดงผลของ โปรแกรมแลปวิว

ในการเขียน VI อันดับแรกคือการเขียนหน้าปัทม์ซึ่งผู้ใช้จะต้องออกแบบส่วนนี้จะจัดวางให้เหมาะสม ซึ่งจะกล่าวถึงในบทต่อไป สำหรับในขั้นนี้เราเพียงแต่เน้นว่าบนหน้าปัทม์ แผงควบคุมด้านหน้า จะประกอบด้วยสองส่วนและการที่เราจะเลือกส่วนควบคุมและตัวแสดงผล เป็นเรื่องสำคัญเพราะทั้งสองนี้ไม่สามารถแทนกันได้ นั่นคือเราไม่สามารถกำหนดค่าส่วนควบคุม แสดงค่าได้และไม่สามารถนำค่าจากตัวแสดงผล ออกไปเป็นข้อมูลของระบบได้

3.5.3 บล็อกไดอะแกรม(Block Diagram)

บล็อกไดอะแกรม เปรียบเสมือนกับซอร์ซโค้ด (Source Code) หรือ โปรแกรมของแลปวิว ซึ่งปรากฏว่าอยู่ในรูปของภาษา G ซึ่งบล็อกไดอะแกรมนี้ ถือว่าเป็นเอ็กซ์ซึคิวทีฟโปรแกรม (Executable Program) คือสามารถที่จะทำงานได้ทันทีและข้อดีอีกประการหนึ่งก็คือแลปวิว จะมีการตรวจสอบความผิดพลาดของโปรแกรมตลอดเวลา ทำให้โปรแกรมจะทำงานได้ก็ต่อเมื่อไม่มีข้อผิดพลาดในโปรแกรมเท่านั้น โดยผู้ใช้สามารถที่จะดูรายละเอียดของความผิดพลาดแสดงให้เห็นได้ตลอดเวลา ทำให้การเขียนโปรแกรมนั้นง่ายขึ้นมาก ส่วนประกอบภายในบล็อกไดอะแกรมนี้จะประกอบด้วย ฟังก์ชัน ค่าคงที่ โปรแกรมควบคุมการทำงานหรือโครงสร้าง



รูปที่ 3.12 แสดงบล็อกไดอะแกรมของโปรแกรมเลปวิว

ถ้าหากเราพิจารณาองค์ประกอบ ของบล็อกไดอะแกรมแล้ว เราจะพบว่าในส่วนของ บล็อกไดอะแกรมจะมีส่วนประกอบที่สำคัญ 3 ส่วน คือ

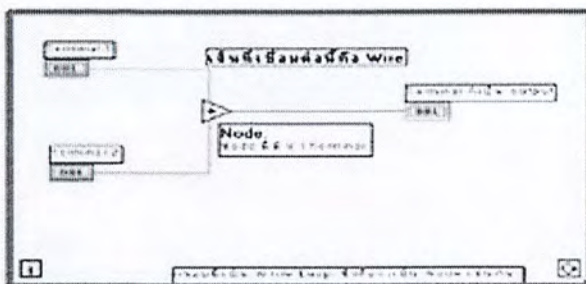
- เทอร์มินอล (Terminal)

ทุกครั้งที่เราสร้างส่วนควบคุมหรือตัวแสดงผล บนแผงควบคุมด้านหน้าในวินโด (Window) ของบล็อกไดอะแกรมจะปรากฏเทอร์มินอลขึ้น เทอร์มินอลก็คล้ายกับสถานีของข้อมูล ก็จะเป็นทั้งสถานีต้นทางของข้อมูล ถ้าเทอร์มินอลนั้นเป็นเทอร์มินอลของส่วนควบคุมและ ขณะเดียวกันจะเป็นทั้งสถานีปลายทางของข้อมูลถ้าเทอร์มินอลนั้นเป็นเทอร์มินอลของตัวแสดงผล ข้อที่ควรเข้าใจอย่างหนึ่งก็คือวัตถุ นี้เกิดขึ้นจากการเขียนขึ้น บนแผงควบคุม ด้านหน้าดังนั้นเมื่อเราจะ ไม่สามารถลบเทอร์มินอลออกจากบล็อก ไดอะแกรมได้และถ้าหากเราจะ ลบส่วนควบคุมหรือตัวแสดงผล นั้นออกไปจากแผงควบคุมด้านหน้าแล้วเทอร์มินอล เหล่านี้ก็จะ หายไปจากบล็อก ไดอะแกรมเช่นกัน สำหรับลักษณะของตัวแปรต่างๆเราจะกล่าวในบทต่อไป

- โหนด (Node)

โหนดใช้เรียกวัตถุที่ทำกรรมวิธีใดๆ เพื่อประมวลข้อมูลในบล็อก ไดอะแกรม เช่นเดียวกับที่เราเขียนโฟลว์ชาร์ต (Flow Chart) แล้วใช้สัญลักษณ์ต่างๆแทนวิธีการวิเคราะห์ข้อมูล เมื่อมีข้อมูลเข้าสู่โหนดสิ่งที่เกิดขึ้นภายใน โหนด ก็จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไปนั้น จะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร หาค่าเฉลี่ยหรือเป็นประเภท การเปรียบเทียบข้อมูล ว่ามากหรือน้อยกว่าหรืออื่นๆ ซึ่งจะเป็นการประมวลผลทางคณิตศาสตร์ ทั่วไป นอกเหนือจากนั้น โหนดนี้จะมีส่วนที่เรียกว่า ฟังก์ชัน (Function) แบบต่างๆ ซึ่งจะเหมือนกับ ฟังก์ชันสำเร็จรูปเช่น ไซน์ (Sine), โคไซน์ (Cosine), ลอการิทึม (Log) เป็นต้น ซึ่งก็จะเหมือนกับในภาษา ที่เป็นตัวอักษรต่างๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงถึงลักษณะของโหนดที่บรรจุอยู่ในบล็อกไดอะแกรมของแลปวิว

- การต่อสาย (Wires)

ขณะที่เรามีที่มาของข้อมูล ส่วนประมวลหรือปรับแต่งข้อมูลและส่วนแสดงผลข้อมูล เรียบร้อยแล้วขั้นต่อไป คือเราจะต้องสามารถควบคุมการส่งผ่านข้อมูลให้เป็นไปตามที่เราต้องการ อุปกรณ์ที่ใช้ในแลปวิวก็คือการต่อสายหรือการต่อสาย ซึ่งจะเป็นการเชื่อมการส่งข้อมูลระหว่างเทอร์มินอลหรือโหนดต่างๆที่มีบล็อกไดอะแกรมนี้เข้าด้วยกัน โดยการต่อสายนี้จะเป็นการกำหนดเส้นทางของข้อมูลว่าเมื่อออกจากเทอร์มินอลหนึ่งแล้ว จะกำหนดการไหลไปที่โหนดใดบ้างมีลำดับเป็นอย่างไรและสุดท้ายจะแสดงผลที่เทอร์มินอลใดนั่นเอง ซึ่งการเชื่อมต่อสายนี้จะทำให้เราเข้าใจถึงหลักการของดาต้าโฟลว์โปรแกรมมิ่ง (Data Flow Programming) ได้ดีขึ้น

เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษรหรือค่าจริงเท็จ (Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบแลปวิวจึงได้กำหนดให้ลักษณะของการต่อสาย สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน

เราจะเข้าใจหลักการของดาต้าโฟลว์โปรแกรมมิ่งได้ ในขณะที่เราต่อสายระหว่างเทอร์มินอลและโหนดต่างๆ เข้าด้วยกัน ซึ่งหลักการทำงานของดาต้าโฟลว์โปรแกรมมิ่งจะต่างจากการเขียนโปรแกรมโดยใช้ตัวอักษร เพราะในโปรแกรมตัวอักษรคำสั่ง จะถูกส่งเข้าสู่ส่วนประมวลผลที่ละบรรทัด เพื่อการคำนวณตามลำดับบรรทัด แต่ในดาต้าโฟลว์นั้น โปรแกรมจะคำนวณเมื่อมีข้อมูลส่งเข้ามาถึงอินพุตของโหนดนั้นๆ ครบและเมื่อข้อมูลส่งครบเข้าถึงโหนดนั้นแล้ว จะมีการประมวลผลและส่งค่าที่ได้ออกไปตามการต่อเชื่อมสาย การคำนวณจะเสร็จสิ้นในแต่ละรอบเมื่อข้อมูลส่งข้ามไปถึงเทอร์มินอลสุดท้าย การประมวลผลไม่ได้เป็นไปตามลำดับการจัดวางคือไม่ได้ทำจากซ้ายไปขวาหรือบนลงล่าง แต่เป็นไปตามขั้นตอนการเดินทางของข้อมูลซึ่งในการเขียนโปรแกรมในเบื้องต้น ผู้ที่คุ้นเคยกับภาษาตัวหนังสือโดยทั่วไปอาจจะต้องใช้เวลาสักครู่ๆ เพื่อที่จะทำความเข้าใจการทำงานของแลปวิว

3.5.4 ไอคอน (Icon) และคอนเนคเตอร์ (Connector)

เปรียบเสมือนโปรแกรมย่อยซ้ำรูปที่ (Subroutine) ในโปรแกรมปกติทั่วไป โดยไอคอนจะหมายถึงบล็อกไคอะแกรมตัวหนึ่งที่มีการส่งข้อมูลเข้าและออกผ่านทางคอนเนคเตอร์ ซึ่งในแลปวิว เราจะเรียก ซับรูทีน นี้ว่า subVI ข้อดีของการเขียนโปรแกรมด้วยภาษา G นี้ก็คือเราสามารถสร้าง VI ทีละส่วนขึ้นมาให้ทำงานด้วยตัวเองได้อย่างอิสระ จากนั้นในภายหลังหากเราต้องการเราก็สามารถเขียนโปรแกรมอื่นขึ้นมาเพื่อเรียกใช้งาน VI ที่เราเคยสร้างขึ้นก่อนหน้านี้ทีละตัว ซึ่งทำให้ VI ที่เราเขียนขึ้นก่อนกลายเป็น subVI ไป การเขียนในลักษณะนี้เราเรียกว่า เขียนเป็น โมดูล (Module)

สำหรับลักษณะทั่วไปของไอคอนและคอนเนคเตอร์ จะแสดงในรูปต่อไปนี้ เราจะเห็นว่าเมื่อเราแสดงในรูปของคอนเนคเตอร์ เราจะพบว่า มีช่องต่อข้อมูลหรือที่เรียกว่าเทอร์มินอลปรากฏให้เห็น

คำศัพท์ต่างๆที่ใช้กันในแลปวิวนี้ ออกจะแตกต่างจากที่เราใช้กันในภาษาการเขียนโปรแกรมตัวหนังสือทั่วไปในหลายๆด้าน ดังนั้นเพื่อให้ผู้ที่เริ่มใช้แลปวิวเข้าใจถึงศัพท์ต่างๆ ที่ใช้ในโปรแกรม เราจึงขอเปรียบเทียบศัพท์ใช้ในแลปวิวกับโปรแกรมพื้นฐานต่างๆ ไปตามตารางที่ได้แสดงต่อไปนี้

ตารางที่ 3.2 ศัพท์ที่ใช้ในโปรแกรมแลปวิว

LabVIEW	โปรแกรมพื้นฐาน	หน้าที่
VI	Program	ตัวโปรแกรมหลัก
Function	function	ฟังก์ชันสำเร็จรูปที่สร้างขึ้นมากับโปรแกรมนั้นเช่น sin, log เป็นต้น
SubVI	Subroutine	โปรแกรมย่อยที่ถูกเรียกใช้โดยโปรแกรมหลัก
Front Panel	user interface	ส่วนที่ติดต่อกับผู้ใช้
Block Diagram	Program code	การเขียนตามขั้นตอนของทีแต่ละโปรแกรมกำหนดขึ้น

3.6 การออกแบบแคลมาลฟิตเตอร์ (Kalman Filter)

ตอนแรกเริ่มหัดใช้แคลมาลฟิตเตอร์ นั้นไม่สามารถทำได้ เริ่มแรกผมใช้แบบจำลองทางคณิตศาสตร์ที่ได้จาก System Identification ซึ่งอยู่ในรูปของระบบแบบต่อเนื่อง (Continuous System) แล้วก็ใช้แบบจำลองนี้ไปหาค่าเกนและหาพลนท์ของแคลมาลฟิตเตอร์ หลังจากนั้นก็นำไปทดลองจริงพบว่าผลที่ได้มันคนละเรื่องกันเลย

เราสามารถวัดตำแหน่งได้ แต่เราต้องการประมาณความเร็วของมอเตอร์ ซึ่งมันไม่ได้อย่างที่เราคิดเลย แต่เมื่อเราค่อยๆคิดและลองนำแบบจำลองแปลงเป็นดิสครีต (Discrete) แล้ว ค่อยนำไปหาอัตราการขยายแคลมาลและพลนท์ของแคลมาล พบว่าผลที่ได้มันเกือบเกือบจะเท่ากับของจริงที่วัดได้เลย จากการที่เราศึกษาเรื่องแคลมาลฟิตเตอร์ จากที่ไม่เข้าใจเลยว่ามันทำงานอย่างไร จากการทดลองและศึกษา โดยใช้เวลาประมาณหนึ่งอาทิตย์ ผมก็ได้พบทางออกแล้วว่าแคลมาลฟิตเตอร์ นั้นสามารถทำงานได้และทำงานได้อย่างมีประสิทธิภาพเป็นอย่างมาก โดยที่ไม่เปลืองหน่วยความจำ (Memory) ของเราเลยและผลที่ได้ก็เกือบเท่าของจริงที่วัดมาเลย

ซึ่งสามารถเขียนขั้นตอนได้ดังนี้ เรามี Transfer function ของระบบเราก็แปลงให้เป็นสมการปริภูมิสถานะ โดยใช้คำสั่ง $Gc=ss(A,B,C,D)$ ในโปรแกรมเมทแลป เมื่อเราได้แบบจำลองแบบต่อเนื่อง แล้วจากนั้นก็ใช้คำสั่ง $Gd=c2d(Gc,Ts)$, Ts คือเวลาการสุ่ม (Sampling Time) ของระบบ เราก็จะได้ปริภูมิสถานะของระบบที่เป็นแบบ discrete แล้วหลังจากนั้นแล้วก็หาค่าเกนของแคลมาลฟิตเตอร์ โดยใช้คำสั่ง $[kal,L,P,M]=kal(Gd,Q,R)$ ก็จะได้ อัตราการขยายแคลมาล M จากนั้นก็นำไปเขียน Measurement Update Equation

$$x[n|n] = x[n|n-1] + M(y[n] - x[n|n-1])$$

$x[n|n-1]$ อ่านว่า ประมาณสเตต x ที่เวลา n โดยใช้ข้อมูลจากเวลา $n-1$

ซึ่งเรามีตัวแปรครบหมดและสามารถแทนเข้าไปสมการด้านบนครบหมดและต่อจากนั้นเราก็เขียนสมการ Time update

$$x[n+1|n] = Ax[n] + Bu[n]$$

แล้วก็นำสมการนี้ไปเขียนในโปรแกรมต่างๆต่อไป เช่น แลปวิว/ไมโครคอนโทรลเลอร์ เป็นอันเสร็จสิ้นขบวนการประมาณสเตตโดยใช้ ดิสครีตแคลมาลฟิตเตอร์แบบเชิงเส้น (Linear)

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองทำงานของใจโรสโคป

จากการเคลื่อนที่ของหุ่นยนต์ทำให้ใจโรสโคป เกิดการเปลี่ยนแปลงสัญญาณที่เอาต์พุต ซึ่งเมื่อใจโรสโคปเคลื่อนที่ไปมา ก็จะทำให้แรงดันเปลี่ยนแปลงขึ้นลงจากแรงดันอ้างอิงของขอมันคือ 2.5 V และมันก็จะแกว่งขึ้นลงที่ 0.25 V ถึง 4.75 V



รูปที่ 4.1 ผลการทดสอบใจโรสโคปขณะอยู่กับที่

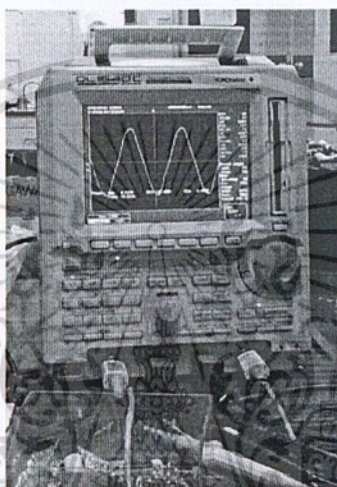


รูปที่ 4.2 ผลการทดสอบใจโรสโคปขณะเคลื่อนที่ไปมา

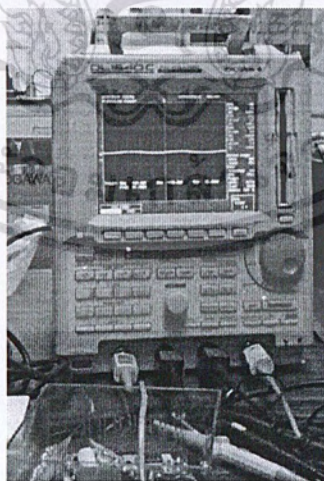
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองทำงานของแอกเซลเลอโรมิเตอร์

การทดสอบการทำงานของแอกเซลเลอโรมิเตอร์ นั้นเราจะเริ่มตั้งแต่การวัดค่าของสัญญาณออกมาแล้วทำการแปลงเอาต์พุตออกมาในรูปของ PWM โดยที่ความถี่ของ PWM จะมีค่าคงที่ ซึ่งเท่ากับ 1 kHz แต่คิวิตีไซเคิล จะเปลี่ยนไปตามความเอียงของเซนเซอร์ ซึ่งการเอียงของมันจะมีช่วงของคิวิตีไซเคิลอยู่ระหว่าง 20% - 80% เมื่อเราใช้ออสซิลโลสโคป (Oscilloscope) วัดสัญญาณเอาต์พุตของแอกเซลเลอโรมิเตอร์ จะได้สัญญาณดังรูป



รูปที่ 4.3 ผลการทดสอบแอกเซลเลอโรมิเตอร์เมื่อเซนเซอร์เอียง



รูปที่ 4.4 ผลการทดสอบแอกเซลเลอโรมิเตอร์เมื่อเซนเซอร์ตั้งฉากกับพื้นโลก

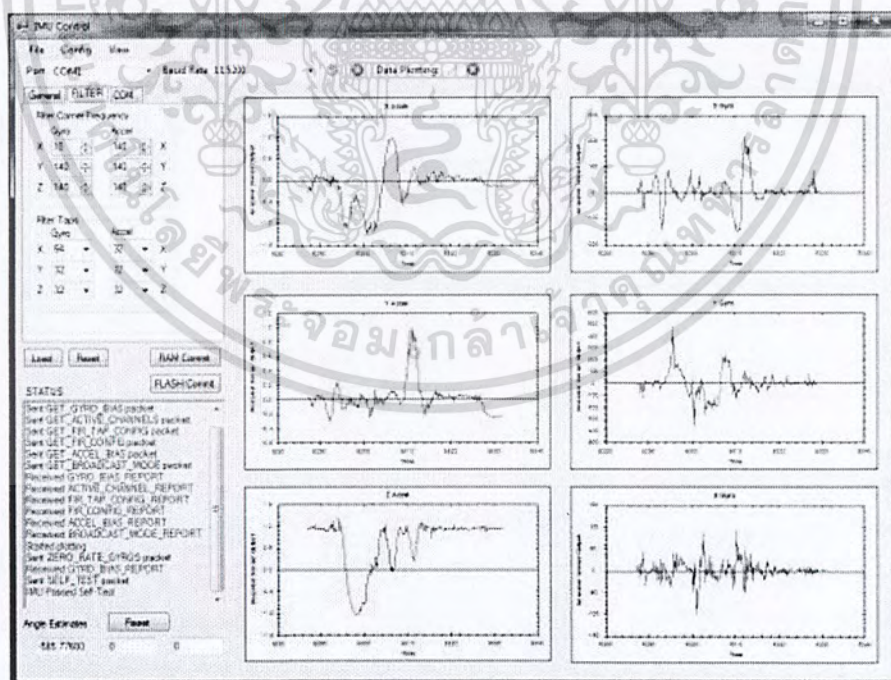
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการรวมสัญญาณเซนเซอร์ด้วยแคลมาลพิตเตอร์

การทดสอบเซนเซอร์เราได้ทำการทดสอบ โดยติดตั้งไว้กับตัวดีกรีสเกลลิง(Degree Scaling) ที่เราสร้างขึ้นมาเองดังรูปที่ 4.5 ซึ่งจะเป็นตัวทดสอบความถูกต้องของการวัดมุมจริงแล้วเทียบกับค่าที่อ่านได้ใน ไมโครคอนโทรลเลอร์ โดยสามารถทดสอบได้ตั้งแต่มุม 0 - 180 องศา

เราได้ทำการอ่านค่าของเซนเซอร์ทั้ง 2 ตัว โดยใช้ไมโครคอนโทรลเลอร์อ่านค่าเข้ามาทุกๆ 18 ms แล้วทำการพล็อต (Plot) ค่าออกไปผ่านทางยูอาร์ที (UART) ของไมโครคอนโทรลเลอร์ แล้วเชื่อมต่อเข้ากับพอร์ต (Port) อาร์เอสสองสามสอง (RS232) ของคอมพิวเตอร์แล้วปริ้นท์ (Print) ออกมา โดยใช้ไฮเปอร์เทอร์มินอล (Hyper Terminal) ซึ่งเราสามารถเก็บค่าที่ปริ้นท์ออกมานั้นอยู่ในรูปของ เท็กไฟล์ (Text File) ซึ่งจะเป็นการง่ายต่อการนำค่าที่ได้ไปซิมูเลท (Simulate) และพล็อตเป็นกราฟด้วยโปรแกรมแมทแล็บ ซึ่งจะสามารถเห็นความแตกต่างที่เกิดขึ้นได้ เมื่อนำค่าจากเซนเซอร์แต่ละตัวมาพล็อตเทียบกัน เราได้ทำการพล็อตกราฟของมุมเอียงที่ได้จากการรวมกันของสัญญาณสองสัญญาณด้วยแคลมาลพิตเตอร์ ดังรูปที่ 4.5

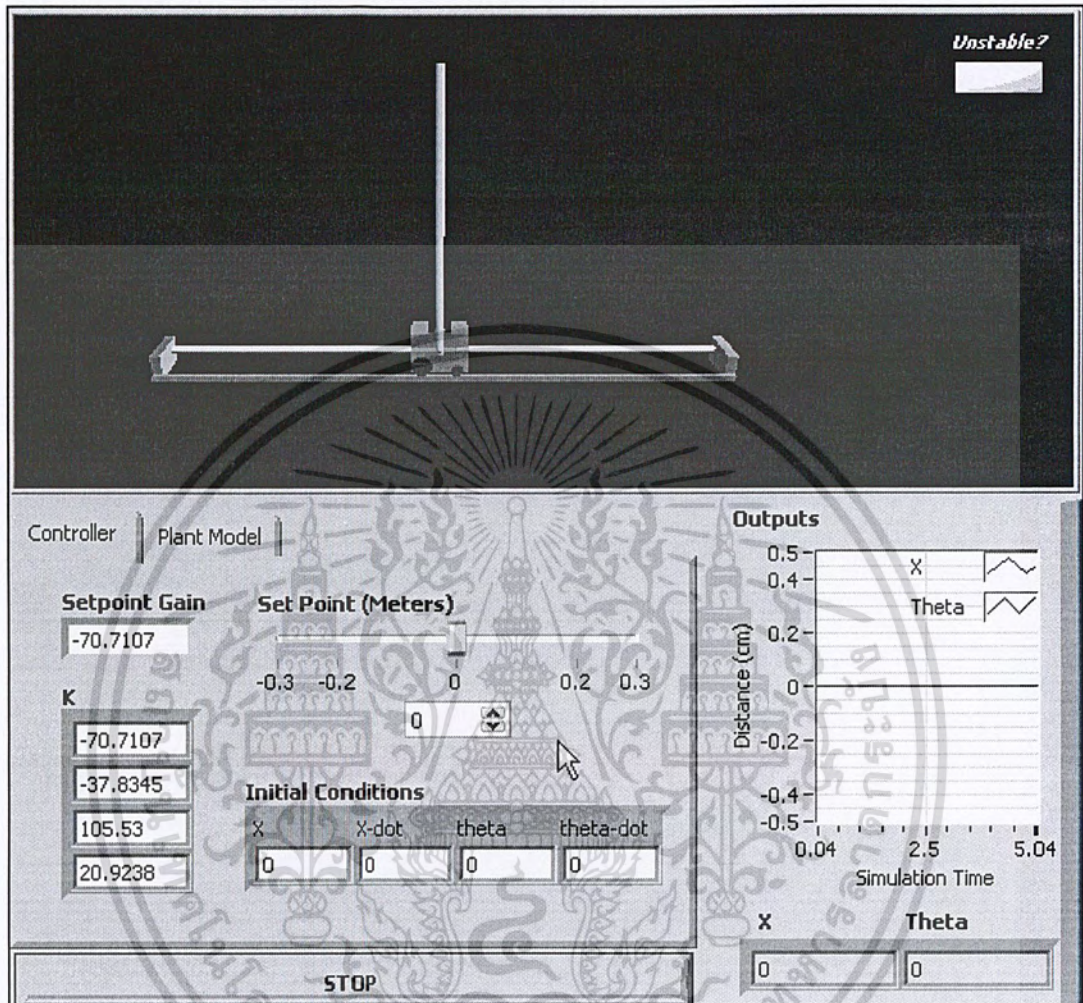
จากกราฟเส้นประแสดงเอาต์พุตของไจโรสโคปมีหน่วยเป็นองศา/วินาที (Degree/Second) กราฟเส้นทึบแสดงถึงเอาต์พุตของแอกเซลเลอโรมิเตอร์ ซึ่งมีหน่วยเป็นองศาและกราฟที่เป็นจุดแสดงผลลัพธ์ที่ได้จากการใช้แคลมาลพิตเตอร์ ในการรวมข้อมูลของทั้งสองเซนเซอร์



รูปที่ 4.5 ทดลองการรวมสัญญาณของไจโรสโคปและแอกเซลเลอโรมิเตอร์ด้วยแคลมาลพิตเตอร์

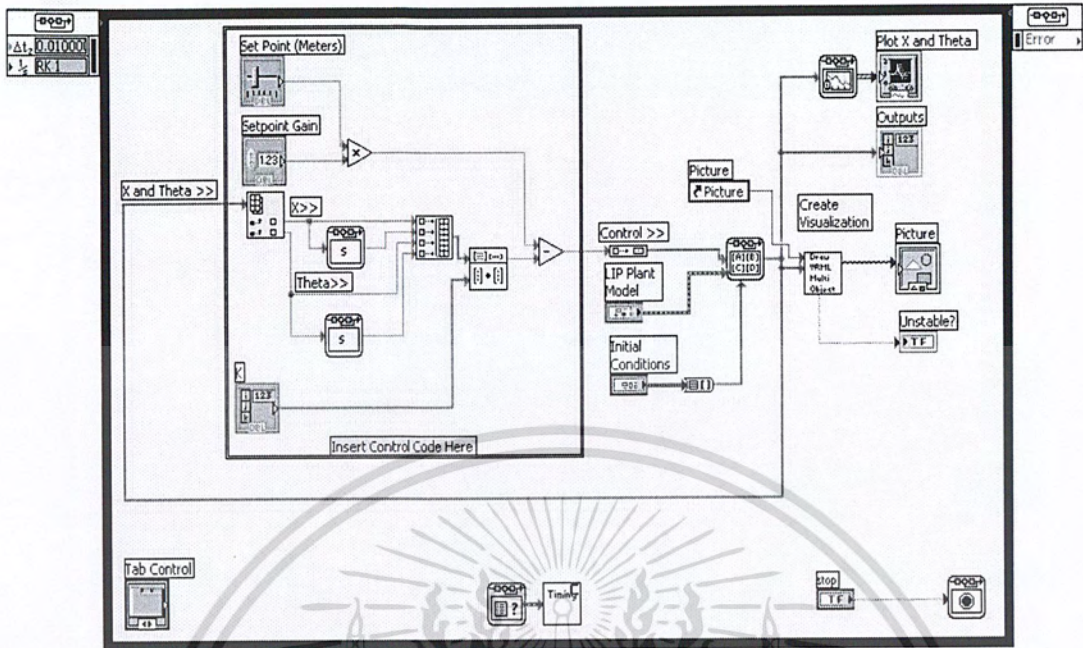
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองการจำลองการควบคุม



รูปที่ 4.6 ทดลองการจำลองระบบอินเวอร์ทเพนดูลัม โดยใช้โปรแกรมแลปวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงบล็อกโคแอดแกรมของระบบอินเวอร์ทิเพนดูลัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองเมื่อระบบถูกรบกวน ทำให้ก้านเพนดูลัมเปลี่ยนแปลงตำแหน่งไปจากจุดสมดุล เช่น เซอร์จะวัดการเอียงไปของก้านเพนดูลัมและวัดระยะทางที่ล้อเคลื่อนที่ไปจากจุดเริ่มต้น โดยนำสัญญาณการเอียง มุมของก้านเพนดูลัมและระยะทางที่ตัวหุ่นยนต์ที่เคลื่อนที่ไปจากจุดสมดุล มาคำนวณ เพื่อหาค่าของสัญญาณควบคุม ไปทำการควบคุมระบบผ่านทางมอเตอร์ ระบบยังสามารถรักษาสถิตอยู่ได้และจะค่อยๆปรับค่าตำแหน่งของก้านเพนดูลัม เพื่อให้ระบบเข้าสู่สมดุลอีกครั้งหนึ่ง

5.2 ปัญหา

1. มุมของก้านเพนดูลัมที่เปลี่ยนแปลงไปจากจุดสมดุลจะกลับสู่สมดุลได้จะขึ้นอยู่กับกำลังของมอเตอร์ เนื่องจากมีข้อจำกัดของกำลังมอเตอร์มุมของก้านเพนดูลัม จึงสามารถเปลี่ยนแปลงได้แค่สิบห้าองศา
2. ปัญหาที่เกิดจากเซนเซอร์ เมื่อปล่อยให้โรสโคปเอาไว้ตัวมันเองจะเกิดสัญญาณผิดพลาดขึ้นเองเพิ่มขึ้น ตามระยะเวลาและไม่สามารถรู้จุดอ้างอิงของตัวเองได้และปัญหาที่สำคัญที่สุดกับเซนเซอร์ชนิดนี้ คือปัญหาเรื่องของการแกว่งของสัญญาณ ทำให้การจับสัญญาณมีความผิดพลาดและส่งผลทำให้การควบคุมผิดพลาดด้วย
3. ชุดเกียร์เกิดมีระยะคลอน ทำให้ระบบการควบคุมไม่ดีเท่าที่ควร
4. มีผลของแรงเสียดทานและน้ำหนักของสายไฟที่แกว่ง

5.3 ข้อเสนอแนะ

1. เปลี่ยนขนาดของมอเตอร์ให้มีกำลังมากขึ้นเพื่อให้ระบบสามารถเปลี่ยนแปลงได้มากขึ้น
2. ปัญหาที่เกิดจากตัวโรสโคป แก้ไขโดยการเพิ่มตัวเซนเซอร์เข้าไป เพื่อเป็นจุดอ้างอิงให้กับตัว โรสโคปหรือใช้ตัวเซนเซอร์ ที่มีประสิทธิภาพมากกว่านี้เพื่อแก้ปัญหาของการเกิดความผิดพลาด ขึ้นของตัว โรสโคปได้
3. เปลี่ยนชุดเกียร์ให้มีระยะคลอนน้อยที่สุด
4. เปลี่ยนโครงสร้างให้มีความสมดุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ผ.ศ.ดร. ธนิตย์ ศรีสุวรรณวัฒน์ “Optimal Control System” เอกสารประกอบการเรียนการสอน.
- [2] Katsuhiko Ogata, Modern Control Engineering, 2nd ed. Englewood Cliffs, New Jersey: Prentice-Hall International Editions, 1990.
- [3] นคร ภักดีชาติ, ชัยวัฒน์ ลิ้มพรจิตรวิไล “dsPIC Microcontroller: Basic Experiment in C Programming with MPLAB C30” Innovative Experiment Co.,Ltd.
- [4] Rich Chi Ooi “Balancing a Two-Wheeled Autonomous Robot” The University of Western Australia,thesis 2003



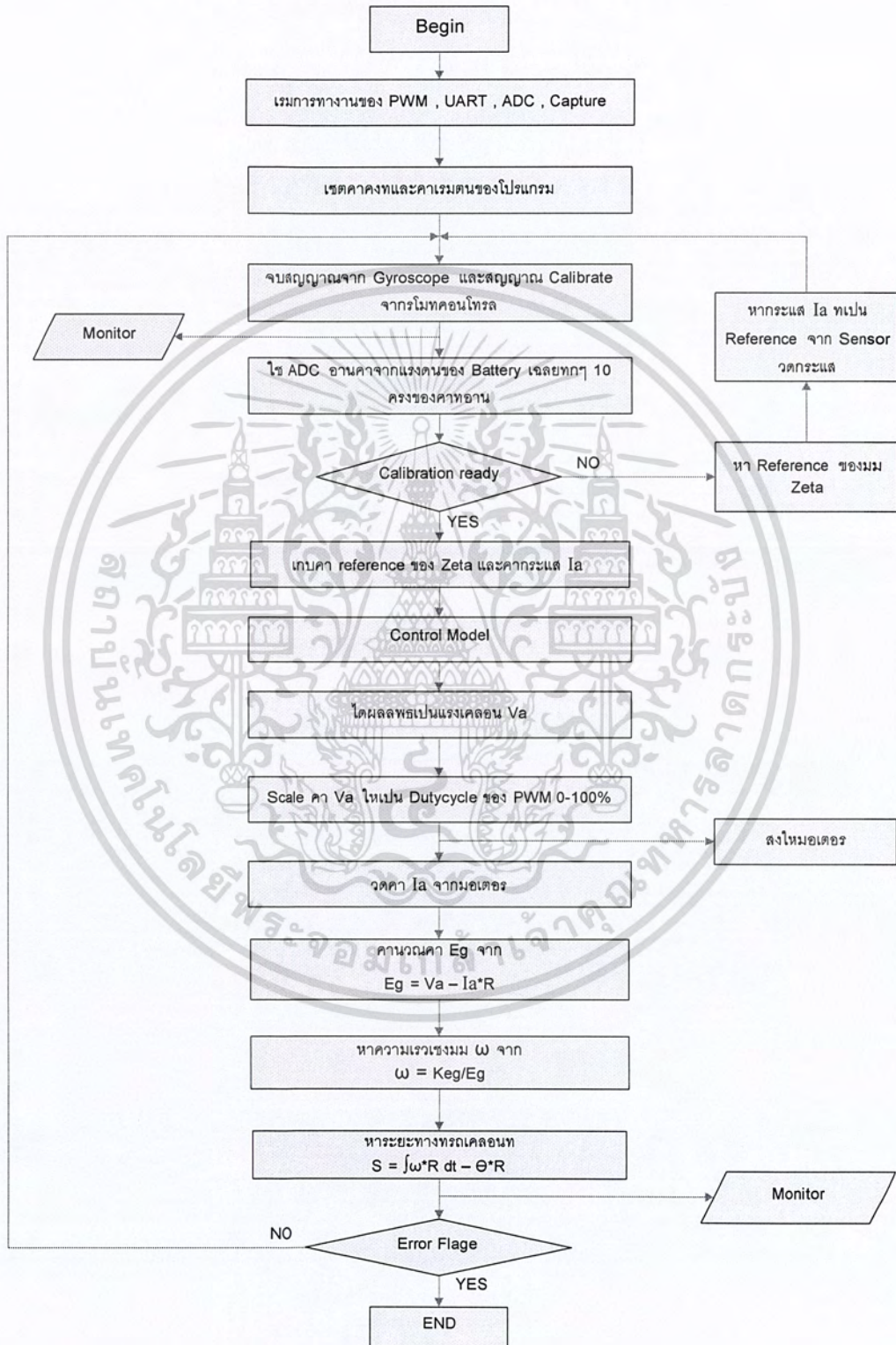
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมที่เขียนบนไมโครคอนโทรลเลอร์ dsPIC



รูปที่ ก.1 แผนภาพขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
    Xtal 9.6 MHz , Oscillator Mode Xt w/PLL 8x
    Watchdog Timer Disable
    update 11 02 2552
    uart 38600
    gyro ADIS16100
    Accelerometer ADXL345 Lis3lv02DQ
*/

```

```

#include<p30f4011.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>

```

```
char Flag_Capture = 0; // "0" is no flag , "1" is yes flag
```

```
char Capture_in_rang = 0; // "0" is out off rang , "1" is in rang
```

```
char Interrupt_Count_Capture;
```

```
unsigned long reference; ;
```

```
unsigned long Capture_period;
```

```
unsigned long period;
```

```
unsigned long max ,min;
```

```
int count ;
```

```
int Error_cap ;
```

```
long sum;
```

```
char calibrate,mm;
```

```
char str[12];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//int x,y;
float v;
unsigned int adc0,adc1;
//////////////////////////////////// Capture //////////////////////////////////////
void __attribute__((__interrupt__)) _IC1Interrupt(void) // แคลปเจอร์ที่ขา RD0
{
    if(PORTDbits.RD0 == 1)
    {
        WriteTimer23(0);
        PORTDbits.RD1 = 0; // ทดสอบว่าอินเทอร์รัพหรือเปล่า
        Flag_Capture = 0;
    }
    if(PORTDbits.RD0 == 0)
    {
        Capture_period = ReadTimer23(); //อ่านค่า
        Error_cap = reference - Capture_period;
        if(mm == 1)
        {
            max = reference+3;
            min = reference-2;
        }
        if(mm == 0)
        {
            max = reference+5;
            min = reference-5;
        }
        ////////////////////////////////////// ฟังก์ชันอินทิเกรต //////////////////////////////////////
        if((Capture_period>max)||((Capture_period<min))) //กรณีที่ capture_period มีค่ามากกว่า
rangeที่กำหนด
        {
            sum = Error_cap+sum;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// if((Capture_period<max)&(Capture_period>min))//กรณีที่น้อยกว่า max แต่มากกว่า
min

else
{
    Error_cap = 0;
    sum = Error_cap+sum;
}
Flag_Capture = 1;
//////////ฟังก์ชันคาลิเบรต//////////
if(calibrate == 1)
{
    if(sum != 0) //sum ไม่เท่ากับ 0
    {
        sum = 0;
        reference = reference - Error_cap;
        count = 0;
    }
    else
    {
        count++;
        if(count == 100)
        {
            mm = 0;
            calibrate = 0; // End of calibrate function
            PORTDbits.RD3 = 1;
        }
    }
}

// LATDbits.LATD2 = !LATDbits.LATD2;

adc0 = analog(0); //0-1023
adc1 = analog(1);
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

////////////////////////////////// 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;
}

////////////////////////////////// IC2 ////////////////////////////////////////////
/*void __attribute__((__interrupt__)) _IC2Interrupt(void) // แคปเจอร์ที่ขา RD1
{
    IFS0bits.IC2IF = 0; // Clear External interrupt 0 flag
    status = 1;
    LATEbits.LATE8 = !LATEbits.LATE8;
}*/

////////////////////////////////// Main Program ////////////////////////////////////////////
int main(void)
{
    char Txdata []="r\nUART test Gyroscope sensor\r\n";
    float sum_zeta;

    unsigned int i1,i2;

    calibrate = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

reference = 60400;
mm= 1;
count = 0;
uart1_init();
PWM_init();
capture_init();

putsUART1((unsigned int *)Txdata);

printf("\r");
Open_capture();
TRISDbits.TRISD3 = 0;
TRISDbits.TRISD1 = 0;
TRISDbits.TRISD2 = 0;
while(1)
{
while (Flag_Capture); // รอให้มีการ เคปเจอร์ก่อน
Flag_Capture = 0;
sum_zeta = sum/1000.00;
i1 = 2.512*adc0;
i2 = 2.512*(1023.0-adc0);
PDC1 = i1 ; // (5KHz=0 - 514)(1KHz=0-2570)
PDC2 = i1 ;
PDC3 = i2 ;

printf("\rzeta=%.5f\t\t%.3d\t\t%d",sum_zeta,count,adc0);

}
}

```

ส่วนต่อไปนี้เป็น โปรแกรมย่อยซึ่งเป็นตัวกำหนดครีจิสเตอร์เพื่อกำหนดการทำงานทางด้านฮาร์ดแวร์ของตัวไมโครคอนโทรลเลอร์ ซึ่งประกอบไปด้วยส่วนของ PWM, Capture , Analog to digital และ Uart

1. PWM

เป็นส่วนของการกำหนดค่าต่างๆ ของ pulsewidth modulation เช่น การกำหนดความถี่ของสัญญาณที่ออก การกำหนดโหมคของ PWM เป็น โหมคอิสระ หรือ แบบคอมพลิเมตารี การกำหนดค่าของเวลาตาย(dead time) เป็นต้น ซึ่งมีโปรแกรมดังนี้

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการขงในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define FCY 3000000 // 30 MIPS
#define FPWM 20000 // 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 = (FCY/FPWM - 1) >> 1; // Compute Period for desired frequency
    PTPER = 0x0500; // XT x16 5 KHz (ก่อนคือ 6675 = 50Hz)(0100 = 5KHz)
    OVDCON = 0x0000; // Disable all PWM outputs.
    DTCON1 = 0x008F; // (5KHz = 0x0088)
    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.
    IFS2bits.PWMIF = 0; // Clear PWM Interrupt flag
    IEC2bits.PWMIE = 1; // Enable PWM Interrupts
    OVDCON = 0x3F00; // PWM outputs are controller by PWM module
    PTCONbits.PTCKPS = 2; // PWM Time Base Input Clock Prescale Select bits 16
    //PTCONbits.PTMOD = 0; // free running mode
    PTCONbits.PTMOD = 2; // 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

```
// Program : Test A/D for JX-dsPIC40 Board
```

```
// Description : in put RB0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Frequency      : 4 MHz at PLL 4x
// Filename       : AD.c
// C compiler     : C30 Compiler by Microchip Technology

#include<p30f4011.h> // Header file for dsPIC30F4011
#include<lcd4011.h> // Header file for dsPIC30F4011
#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 and starts converting

    ADCHS = ch; // Connect RB0/AN0 as CH0 input in this example RB0/AN0 is the input
    ADCSSL = 0;
    ADCON3 = 0x0005; // Manual Sample, Tad = internal 2 Tcy
    ADCON2 = 0;
    ADCON1bits.ADON = 1; // turn ADC ON
    ADCON1bits.SAMP = 1; // start sampling ...

    for(a=0;a<102;a++); // Loop for delay 1 millisecc 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()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* 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

    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 &

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือเป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UART_TX_ENABLE          &
UART_INT_RX_CHAR        &
UART_ADR_DETECT_DIS     &
UART_RX_OVERRUN_CLEAR;

```

```
OpenUART1( U1ModeValue, U1StatusValue, BaudRateValue );
```

```
SetPriorityIntU1RX(3);
```

```
EnableIntU1RX;
```

```
}
```

ผู้สนใจสามารถนำโปรแกรมนี้นี้มาใช้ได้เลย ซึ่ง โปรแกรมย่อยทั้ง 4 นั้น สามารถนำมารวมกับ โปรแกรมหลักได้เลย หรือ แยกกันก็ได้ โดยถ้าแยกกันจะต้องกำหนด directory ของโปรแกรมย่อย ใน Include Path ซึ่งอยู่ในส่วนของ Build option

```
/*
```

```
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<p30f4011.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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include<ACD.c>
#include<ports.h>

char Flag_Capture = 0; // "0" is no flag , "1" is yes flag
char Capture_in_rang = 0; // "0" is out off rang , "1" is in rang
char Interrupt_Count_Capture;
unsigned long reference; ;
unsigned long Capture_period;
unsigned long period;
unsigned long max ,min;
int count ;
int Error_cap ;
long sum;
char calibrate,mm;
char str[12];
//int x,y;
float v;
unsigned int adc0,adc1;
//////////////////////////////////// Capture //////////////////////////////////////
void __attribute__((__interrupt__)) _IC1Interrupt(void) // แคลปเจอร์ที่ขา RD0
{
    if(PORTDbits.RD0 == 1)
    {
        WriteTimer23(0);
        PORTDbits.RD1 = 0; // ทดสอบว่าอินเทอร์รัพหรือเปล่า
        Flag_Capture = 0;
    }
    if(PORTDbits.RD0 == 0)
    {
        Capture_period = ReadTimer23(); //อ่านค่า
        Error_cap = reference - Capture_period;
    }
    if(mm == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

max = reference+3;

        min = reference-2;
    }
    if(mm == 0)
    {
        max = reference+5;
        min = reference-5;
    }

    ////////////////////////////////////ฟังก์ชันอินทิเกรต////////////////////////////////////

    if((Capture_period>max)||((Capture_period<min))//กรณีที่ capture_period มีค่ามากกว่า
rangeที่กำหนด
    {
        sum = Error_cap+sum;
    }

//
min
    if((Capture_period<max)&(Capture_period>min))//กรณีที่น้อยกว่า max แต่มากกว่า
else
    {
        Error_cap = 0;
        sum = Error_cap+sum;
    }

    Flag_Capture = 1;

    ////////////////////////////////////ฟังก์ชันคาร์ริเบรต////////////////////////////////////

    if(calibrate == 1)
    {
        if(sum != 0) //sum ไม่เท่ากับ 0
        {
            sum = 0;
            reference = reference - Error_cap;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ count = 0; เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else
    {
        count++;
        if(count == 100)
        {
            mm = 0;
            calibrate = 0;          // End of calibrate function
            PORTDbits.RD3 = 1;
        }
    }
}

// LATDbits.LATD2 = !LATDbits.LATD2;
adc0 = analog(0); //0-1023
adc1 = analog(1);
}
IFS0bits.ICIIF = 0; // เคลียร์อินเตอร์พแคปเจอร์
}

//////////////////////////////////// PWM //////////////////////////////////////
void __attribute__((__interrupt__)) _PWMIInterrupt(void)
{
    IFS2bits.PWMIF = 0;
}

//////////////////////////////////// TX1 RX1 //////////////////////////////////////
void __attribute__((__interrupt__)) _UITXInterrupt(void)
{
    IFS0bits.U1TXIF=0;
}

void __attribute__((__interrupt__)) _UIRXInterrupt(void)
{

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
////////////////////////////////////////////////////////////////
/*void __attribute__((__interrupt__)) _IC2Interrupt(void) // แลปเจอร์ที่ขา RD1
{
    IFS0bits.IC2IF = 0;          // Clear External interrupt 0 flag
    status = 1;
    LATEbits.LATE8 = !LATEbits.LATE8;
}*/

```

```

//////////////////////////////////////////////////////////////// Main Program //////////////////////////////////////////////////////////////////

```

```

int main(void)
{
    char Txdata []="\\n\\nUART test Gyroscope sensor\\n\\n";
    float sum_zeta;

    unsigned int i1,i2;

    calibrate = 1;
    reference = 60400;
    mm= 1;
    count = 0;
    uart1_init();
    PWM_init();
    capture_init();

```

```

putsUART1((unsigned int *)Txdata );

printf("\\n");
    Open_capture();
    TRISDbits.TRISD3 = 0;
    TRISDbits.TRISD1 = 0;
    TRISDbits.TRISD2 = 0;
    while(1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (Flag_Capture); // รอให้มีการ เคปเจอร์ก่อน
    Flag_Capture = 0;
    sum_zeta = sum/1000.00;
    i1 = 2.512*adc0;
    i2 = 2.512*(1023.0-adc0);
    PDC1 = i1 ; // (5KHz=0 - 514)(1KHz=0-2570)
    PDC2 = i1 ;
    PDC3 = i2 ;
    printf("\rzeta=%.5f\t\t%.3d\t\t%d",sum_zeta,count,adc0);
}
}

```

ส่วนต่อไปนี้เป็นโปรแกรมย่อยซึ่งเป็นตัวกำหนดรีจิสเตอร์เพื่อกำหนดการทำงานทางด้านฮาร์ดแวร์ของตัวไมโครคอนโทรลเลอร์ ซึ่งประกอบไปด้วยส่วนของ PWM, Capture , Analog to digital และ Uart

1. PWM

เป็นส่วนของการกำหนดค่าต่างๆ ของ pulsewidth modulation เช่น การกำหนดความถี่ของสัญญาณที่ออก การกำหนดโหมคของ PWM เป็นโหมคอิสระ หรือ แบบคอมพลิเมนต์ การกำหนดค่าของเวลาดตาย(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 = (FCY/FPWM - 1) >> 1; // Compute Period for desired frequency
    PTPER = 0x0500; // XT x16 5 KHz (ก่อนคือ 6675 = 50Hz)(0100 = 5KHz)
    OVDCON = 0x0000; // Disable all PWM outputs.
    DTCON1 = 0x008F; // (5KHz = 0x0088)
    PWMCON1 = 0x00FF; // Enable PWM output pins and enable complementary mode 0077
    PDC1 = PTPER; /* 0 Volts on Phase A. This value corresponds to*/

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการขังลิขสิทธิ์ของสำนักงานคณะกรรมการกฤษฎีกา ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PDC2 = PTPER; // 0 Volts on Phase B.
PDC3 = PTPER; // 0 Volts on Phase C.
IFS2bits.PWMIF = 0; // Clear PWM Interrupt flag
IEC2bits.PWMIE = 1; // Enable PWM Interrupts
OVDCON = 0x3F00; // PWM outputs are controller by PWM module
PTCONbits.PTCKPS = 2; // PWM Time Base Input Clock Prescale Select bits 16
//PTCONbits.PTMOD = 0; // free running mode
PTCONbits.PTMOD = 2; // Center aligned PWM operation
PTCONbits.PTEN = 1; // Start PWM
return;
}

```

2. ส่วนของ ADC หรือ Analog to Digital

ส่วนนี้จะมีการกำหนดที่สำคัญอยู่ คือ การเซ็ตขาของไมโครคอนโทรลเลอร์ให้เป็นขาที่รับสัญญาณทางอนาล็อก เรื่องของการเริ่มการ conversion และการหยุด conversion ความเร็วในการ sample ของสัญญาณ ซึ่งมีโปรแกรมดังนี้

```

// Program           : Test A/D for JX-dsPIC40 Board
// Description        : in put RB0
// Frequency          : 4 MHz at PLL 4x
// Filename           : AD.c
// C compiler         : C30 Compiler by Microchip Technology

#include<p30f4011.h> // Header file for dsPIC30F4011
#include<lcd4011.h> // Header file for dsPIC30F4011
#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 and starts converting

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADC_HS = ch; // Connect RB0/AN0 as CH0 input in this example RB0/AN0 is the input
ADC_SSR = 0;

ADC_CON3 = 0x0005; // Manual Sample, Tad = internal 2 Tcy
ADC_CON2 = 0;

ADC_CON1bits.ADON = 1; // turn ADC ON

ADC_CON1bits.SAMP = 1; // start sampling ...

for(a=0;a<102;a++); // Loop for delay 1 millisecon per unit(7.3728 MHz@ PLL4x) 816

ADC_CON1bits.SAMP = 0; // start Converting

while (!ADC_CON1bits.DONE); // conversion done?

val1 = ADCBUF0; // yes then get ADC value

return(val1);
}

```

3. ส่วนของ Capture หรือ การจับสัญญาณ

เป็นส่วนใช้ในการจับสัญญาณ PWM ของไมโครคอนโทรลเลอร์ ซึ่งต้องกำหนดการทำงานของขาอินพุต ซึ่งในที่นี้ใช้ขา RD0 และกำหนดลำดับความสำคัญของการอินเทอร์รัพ

```

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()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    int U1ModeValue;
    int U1StatusValue;
    unsigned BaudRateValue = 65; // 38400 when PLL=16x, XTAL1=30MHz

    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;
}

```

ผู้สนใจสามารถนำโปรแกรมนี้อมาใช้ได้เลย ซึ่ง โปรแกรมน้อยทั้ง 4 นั้น สามารถนำมารวมกับ
โปรแกรมหลักได้เลย หรือ แยกกันก็ได้ โดยถ้าแยกกันจะต้องกำหนด Directory ของ โปรแกรมน้อย ใน
Include Path ซึ่งอยู่ในส่วนของ Build option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่เขียนบนแมทแลป

```

g = 9.81;
r = 0.035;
Mw = 0.03;
Mp = 0.8;
Iw = 0.000018;
Ip = 0.008;
l = 0.14;
Km = 0.4528;
Ke = 0.5674;
R = 14.5;
beta = (2*Mw+(2*Iw/r^2)+Mp);
alpha = (Ip*beta + 2*Mp*l^2*(Mw + Iw/r^2));
A = [0 1 0 0;
0 (2*Km*Ke*(Mp*l*r-Ip-Mp*l^2))/(R*r^2*alpha) (Mp^2*g*l^2)/alpha 0;
0 0 0 1;
0 (2*Km*Ke*(r*beta - Mp*l))/(R*r^2*alpha) (Mp*g*l*beta)/alpha 0];
B2 = [ 0;
(2*Km*(Ip + Mp*l^2 - Mp*l*r))/(R*r*alpha);
0;
(2*Km*(Mp*l-r*beta))/(R*r*alpha)];
C2 = [1 0 0 0;0 0 1 0];
D=[0;0];
OLPoles=eig(A)
[num , den ]=ss2tf(A,B2,C2,zeros(2,1),1)

figure(1)
plot(real(OLPoles),imag(OLPoles),'wx')
xlabel('Re-Axis')
ylabel('Im-Axis')

```

$T_s = 1/50$;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[Ad,Bd,Cd,Dd] = C2DM (A, B2, C2, D, Ts,'zoh')
```

```
OLPoles=eig(Ad)
```

```
[ num1 , den1 ]=ss2tf(Ad,Bd,Cd,zeros(2,1),1)
```

```
figure(2);
```

```
zplane(num1,den1);
```

```
title('z-Plane');
```

```
Ts = 1/100;
```

```
pend_d = ss(Ad,Bd,Cd,Dd,Ts);
```

```
co = ctrb(pend_d);
```

```
ob = obsv(pend_d);
```

```
Controllability = rank(co)
```

```
Observability = rank(ob)
```

```
Q= [ 1   0   0   0   ;
      0   0   0   0   ;
      0   0   1   0   ;
      0   0   0   0   ];
```

```
R=1;
```

```
K=dlqr(Ad,Bd,Q,R);
```

```
F=-K
```

```
Acl=(Ad+Bd*F);
```

```
Bcl=Bd;
```

```
Ccl=Cd;
```

```
Dcl=D;
```

```
H=Acl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLPoles=eig(H)
figure(3);
plot(real(CLPoles),imag(CLPoles),'w+')
xlabel('Re-Axis')
ylabel('Im-Axis')

```

```

R1=[1 0 ;0 1];
LN=dlqr(Ad',Cd',Q,R1)';
L=-LN
He=Ad+L*Cd

```

```

CLPoles2=eig(He)

figure(4);
plot(real(CLPoles2),imag(CLPoles2),'w+')
xlabel('Re-Axis')
ylabel('Im-Axis')

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์



dsPIC30F6010A/6015

dsPIC30F6010A/6015 Enhanced Flash 16-bit Digital Signal Controller (DSC)

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 nonvolatile 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)
 - 7.37 MHz internal RC 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
- ± 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 (dsPIC306010A)
- 1 CAN module, 2.0B compliant (dsPIC306015)

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

dsPIC30F6010A/6015

Analog Features:

- 10-bit Analog-to-Digital Converter (A/D) with 4 S/H Inputs:
 - 1 Msps conversion rate
 - 16 input channels
 - Conversion available during Sleep and Idle
- 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

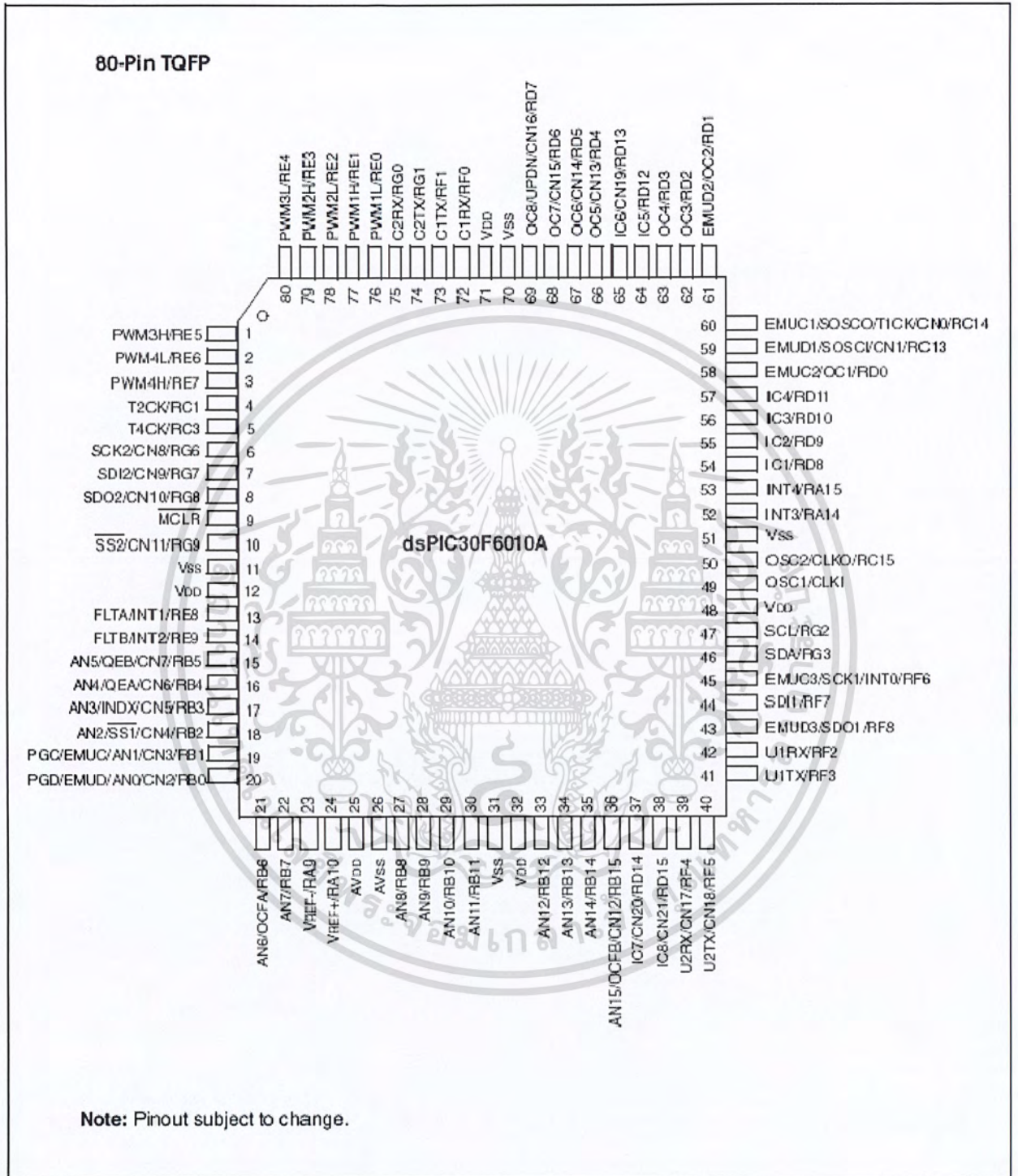
dsPIC30F Motor Control and Power Conversion Family*

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	A/D 10-bit 1 Msps	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
dsPIC30F5016	80	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010A	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2
dsPIC30F6015	64	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	1

* This table provides a summary of the dsPIC30F6010A/6015 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

dsPIC30F6010A/6015

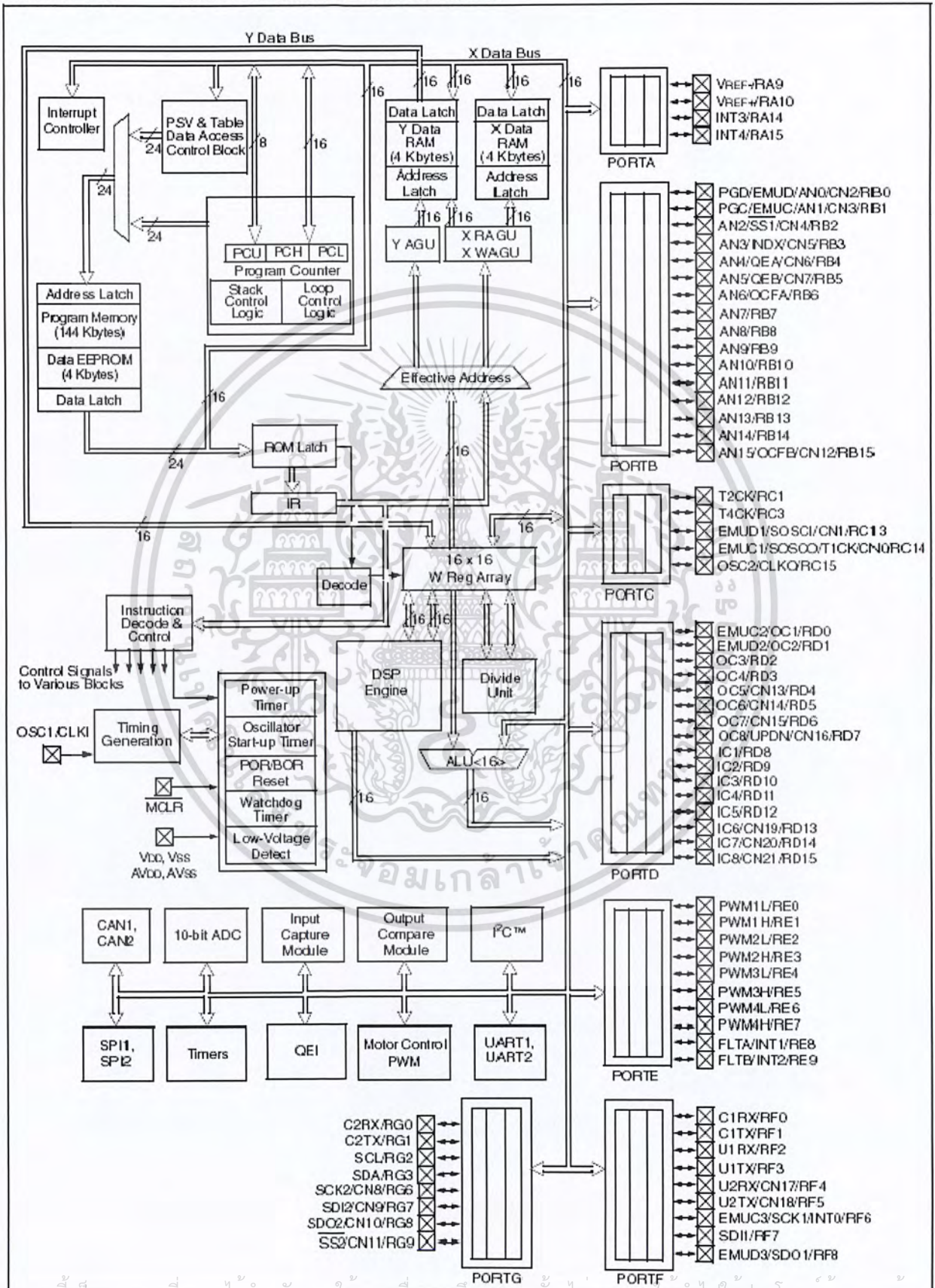
Pin Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

dsPIC30F6010A/6015

FIGURE 1-1: dsPIC30F6010A BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า

ไมโครชิพ เทคโนโลยี จำกัด ขอสงวนสิทธิ์ในข้อมูลนี้ และขอสงวนสิทธิ์ในข้อมูลนี้

dsPIC30F6010A/6015

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: dsPIC30F6010A/6015 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.
CLK1 CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLK0 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.
C1RX C1TX C2RX C2TX	I O I O	ST — ST —	CAN1 bus receive pin. CAN1 bus transmit pin. CAN2 bus receive pin. CAN2 bus transmit pin.
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2 EMUD3 EMUC3	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST ST ST	ICD Primary Communication Channel data input/output pin. ICD Primary Communication Channel clock input/output pin. ICD Secondary Communication Channel data input/output pin. ICD Secondary Communication Channel clock input/output pin. ICD Tertiary Communication Channel data input/output pin. ICD Tertiary Communication Channel clock input/output pin. ICD Quaternary Communication Channel data input/output pin. ICD Quaternary Communication Channel clock input/output pin.
IC1-IC8	I	ST	Capture inputs 1 through 8.
INDX QEA QEB	I I I	ST ST ST	Quadrature Encoder Index Pulse input. Quadrature Encoder Phase A input in QE1 mode. Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QE1 mode. Auxiliary Timer External Clock/Gate input in Timer mode.
UPDN	O	CMOS	Position Up/Down Counter Direction State.
INT0 INT1 INT2 INT3 INT4	I I I I I	ST ST ST ST ST	External interrupt 0. External interrupt 1. External interrupt 2. External interrupt 3. External interrupt 4.

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

dsPIC30F6010A/6015

TABLE 1-1: dsPIC30F6010A/6015 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 bidirectional I/O port.
RA14-RA15	I/O	ST	
RB0-RB15	I/O	ST	PORTB is a bidirectional I/O port.
RC1	I/O	ST	PORTC is a bidirectional I/O port.
RC3	I/O	ST	
RC13-RC15	I/O	ST	
RD0-RD15	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE9	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF8	I/O	ST	PORTF is a bidirectional I/O port.
RG0-RG3	I/O	ST	PORTG is a bidirectional 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.
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
T4CK	I	ST	Timer4 external clock input.

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

dsPIC30F6010A/6015

TABLE 1-1: dsPIC30F6010A/6015 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
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

ข้อมูลของไมโครคอนโทรลเลอร์รุ่นนี้ที่ได้นำเสนอ เป็นเพียงทางด้านคุณสมบัติที่สำคัญเท่านั้น จะไม่กล่าวถึงคุณลักษณะต่างๆ เช่น สถาปัตยกรรม โครงสร้างทางโปรแกรม หน่วยประมวลผลสัญญาณดิจิทัล เป็นต้น ซึ่งข้อมูลเหล่านี้ผู้สนใจสามารถค้นคว้าหาข้อมูลได้จากเว็บไซต์ของบริษัทไมโครชิฟ



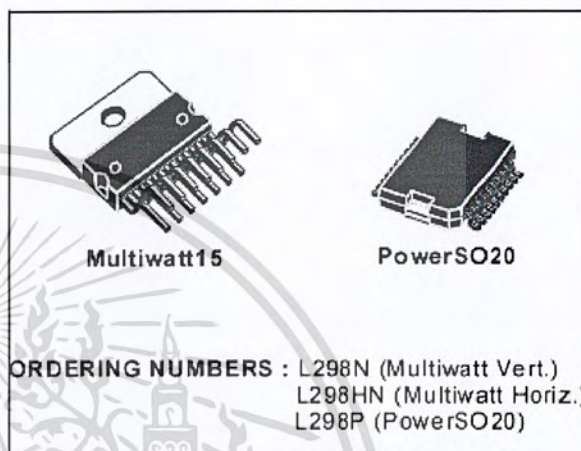
L298

DUAL FULL-BRIDGE DRIVER

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

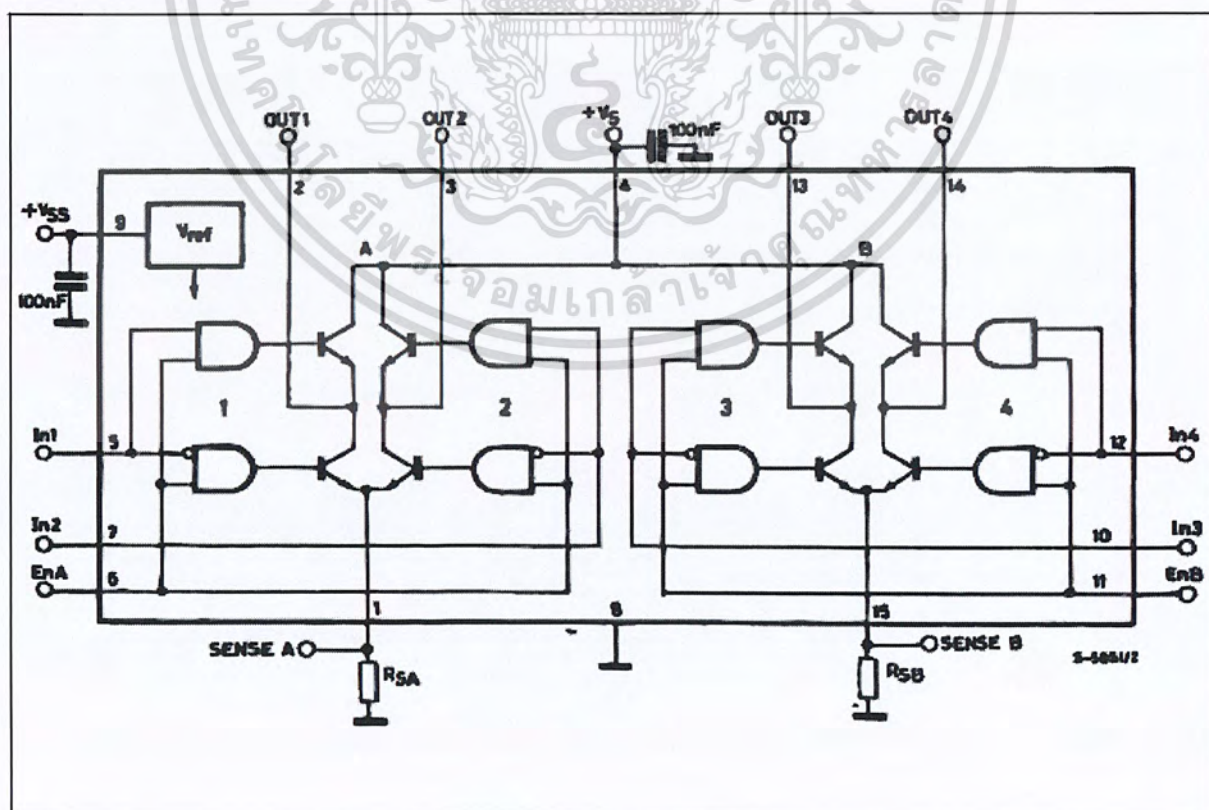
DESCRIPTION

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



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

BLOCK DIAGRAM

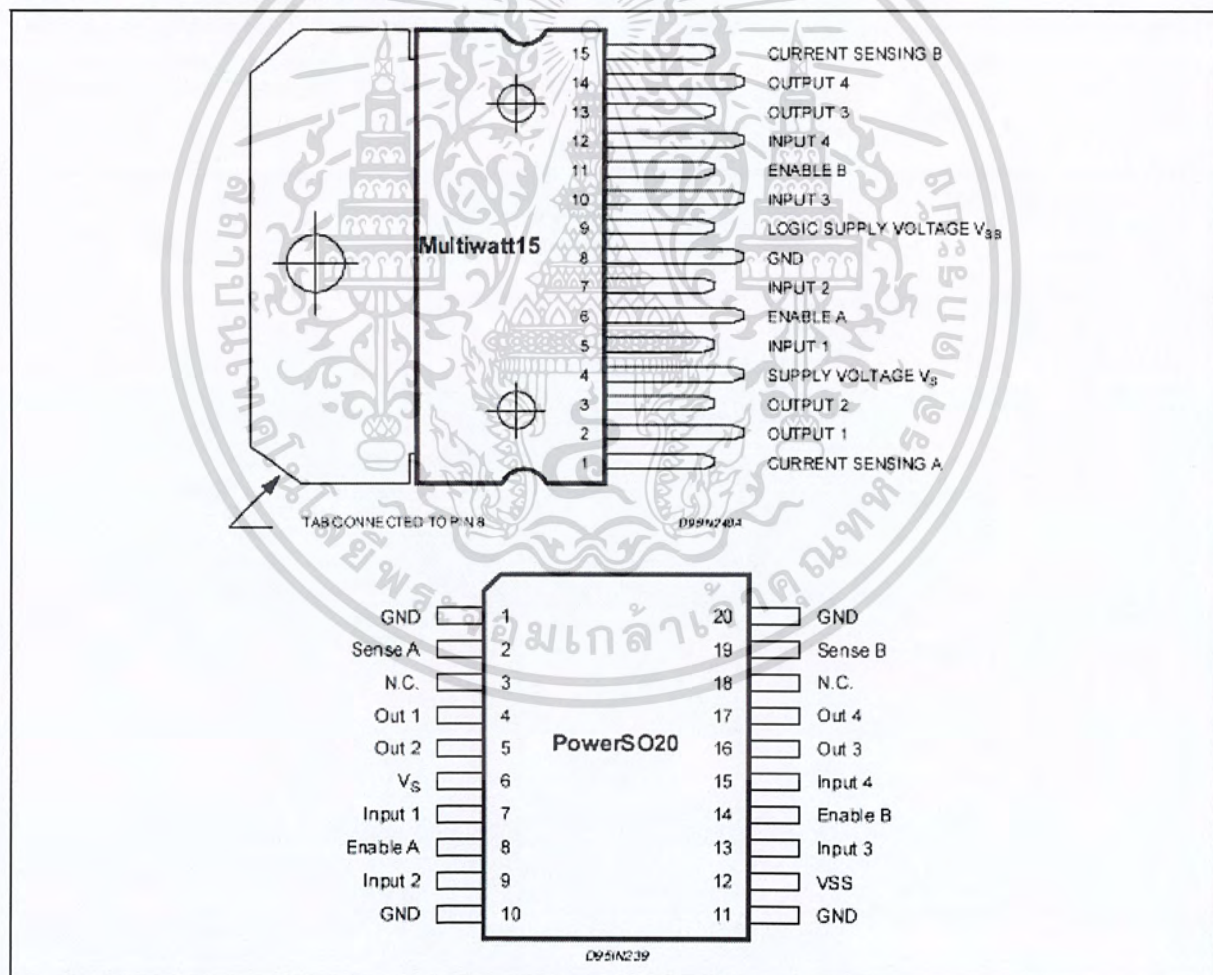


L298

ABSOLUTE MAXIMUM RATINGS

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

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{\theta j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{\theta j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

เอก (*) Mounted on aluminum substrate สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ 2/13 elli doq ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ 3/13

L298

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

1) Sensing voltage can be -1 V for t_s ≤ 50 μsec; in steady state V_{sen1} min ≥ -0.5 V.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

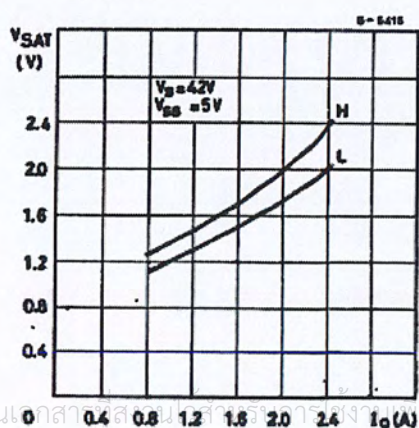
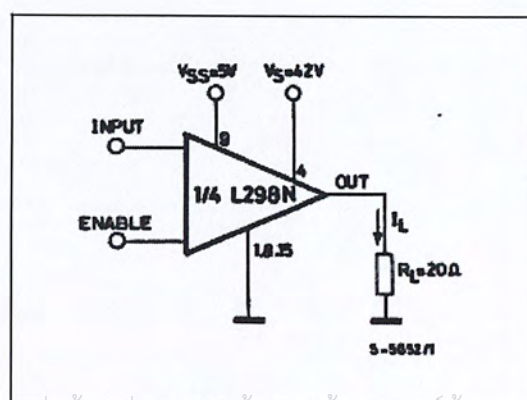


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H;
For ENABLE Switching, set IN = H

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ STMicroelectronics. การศึกษาและการใช้เอกสารนี้เป็นไปตามเงื่อนไขการใช้งานที่ระบุไว้ในเอกสารฉบับนี้. การทำซ้ำหรือการเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย. STMicroelectronics ขอสงวนสิทธิ์ในข้อมูลและข้อกำหนดที่ปรากฏในเอกสารนี้. ข้อมูลและข้อกำหนดอาจมีการเปลี่ยนแปลงโดยไม่ต้องแจ้งให้ทราบล่วงหน้า. โปรดตรวจสอบเอกสารฉบับล่าสุดเสมอ.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

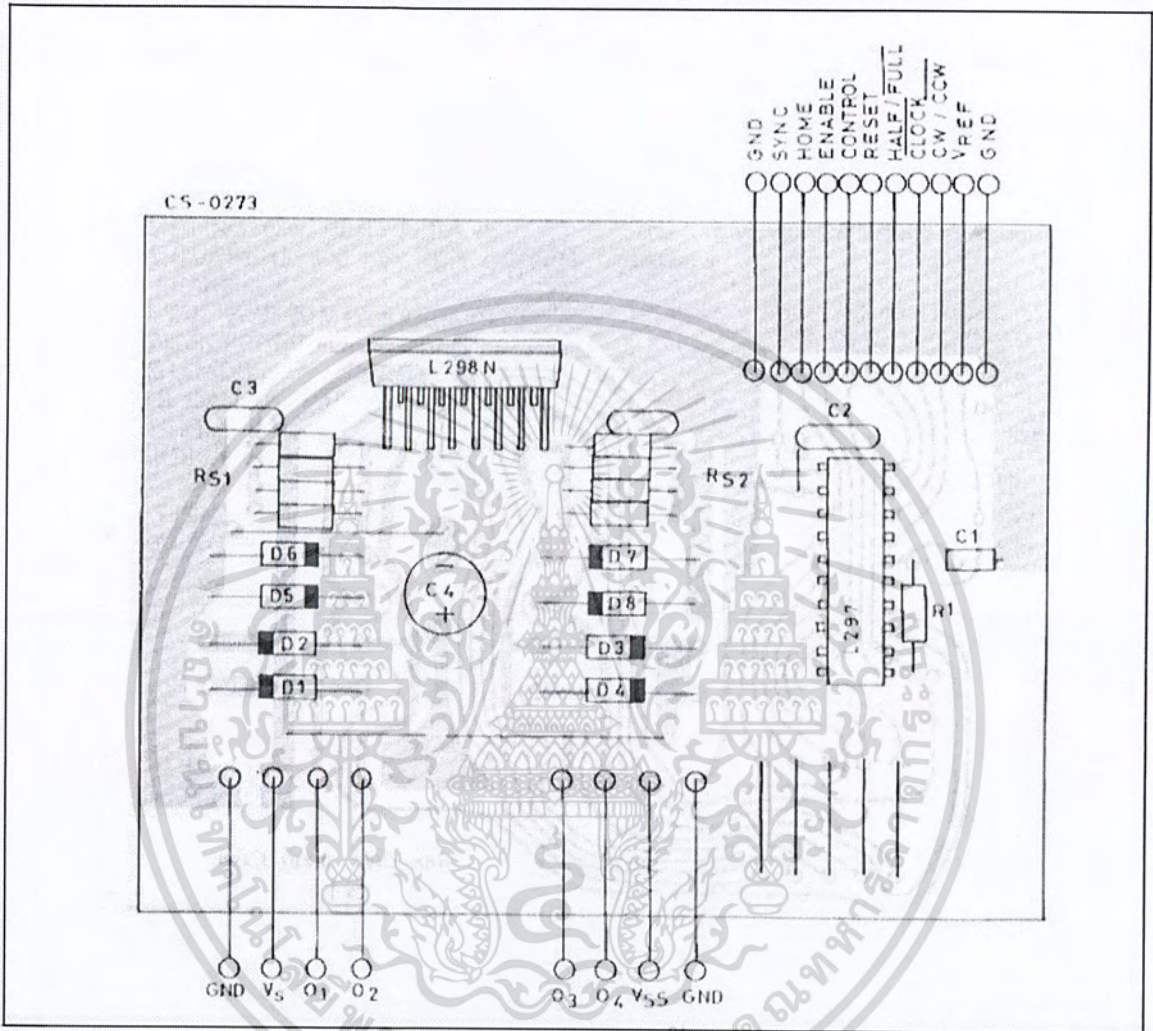
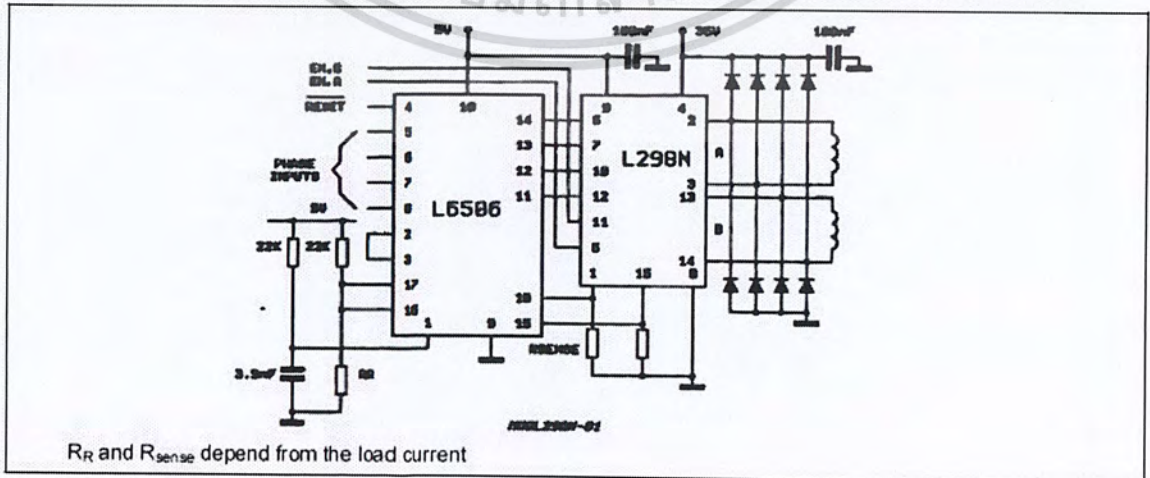


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.





$\pm 300^\circ/\text{sec}$ Yaw Rate Gyroscope with SPI

ADIS16100

FEATURES

- Complete angular rate gyroscope
- Z-axis (yaw rate) response
- SPI digital output interface
- High vibration rejection over wide frequency
- 2000 g-powered shock survivability
- Externally controlled self-test
- Internal temperature sensor output
- Dual auxiliary 12-bit ADC inputs
- Absolute rate output for precision applications
- 5V single-supply operation
- 8.2 mm × 8.2 mm × 5.2 mm package

APPLICATIONS

- Platform stabilization
- Image stabilization
- Guidance and controls
- Inertia measurement units
- Robotics

GENERAL DESCRIPTION

The ADIS16100 is a gyroscope that uses the Analog Devices, Inc., surface-micromachining process to make a functionally complete angular rate sensor with an integrated serial peripheral interface (SPI).

The digital data available at the SPI port is proportional to the angular rate about the axis that is normal to the top surface of the package (see Figure 20). A single external resistor can be used to increase the measurement range. An external capacitor can be used to lower the bandwidth.

Access to an internal temperature sensor measurement is provided through the SPI for compensation techniques. Two pins are available for the user to input analog signals for digitization. An additional output pin provides a precision voltage reference. Two digital self-test inputs electromechanically excite the sensor to test the operation of the sensor and the signal-conditioning circuits.

The ADIS16100 is available in an 8.2 mm × 8.2 mm × 5.2 mm, 16-terminal, peripheral land grid array (LGA) package.

FUNCTIONAL BLOCK DIAGRAM

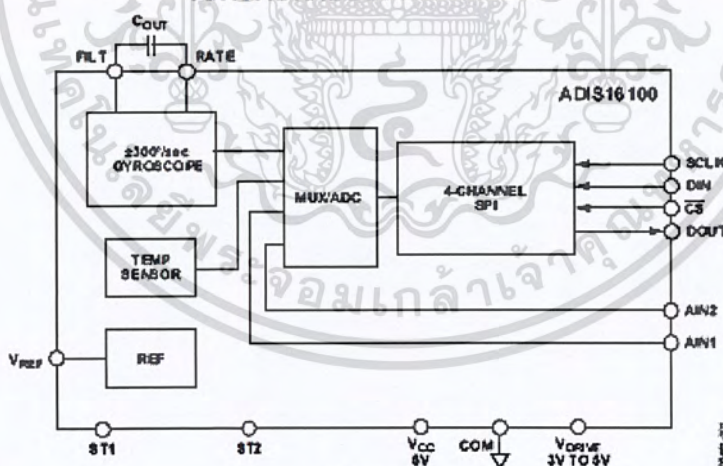


Figure 1.

Rev. D

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2006–2009 Analog Devices, Inc. All rights reserved.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADIS16100

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_{CC} = V_{DRIVE} = 5\text{ V}$, angular rate = $0^\circ/\text{sec}$, $C_{OUT} = 0\ \mu\text{F}$, $\pm 1\text{ g}$, unless otherwise noted.

Table 1.

Parameter	Conditions	Min ¹	Typ	Max ¹	Unit
SENSITIVITY					
Dynamic Range ²	Full-scale range over specifications range	± 300			$^\circ/\text{sec}$
Initial	Clockwise rotation is positive output, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$	0.2212	0.2439	0.2717	$^\circ/\text{sec}/\text{LSB}$
Change Over Temperature ³	$V_{CC} = V_{DRIVE} = 4.75\text{ V}$ to 5.25 V		± 5		$\%$
Nonlinearity	Best fit straight line		0.15		$\%FS$
NULL					
Initial	Nominal $0^\circ/\text{sec}$ output is 2048 LSB	-42		+42	$^\circ/\text{sec}$
Change Over Temperature ³	$V_{CC} = V_{DRIVE} = 4.75\text{ V}$ to 5.25 V		± 10		$^\circ/\text{sec}$
Turn-On Time	Power-on to $\pm 0.5^\circ/\text{sec}$ of final value		35		ms
Linear Acceleration Effect	Any axis		0.2		$^\circ/\text{sec}/\text{g}$
Voltage Sensitivity	$V_{CC} = V_{DRIVE} = 4.75\text{ V}$ to 5.25 V		± 1		$^\circ/\text{sec}/\text{V}$
NOISE PERFORMANCE					
Total Noise	0.1 Hz to 40 Hz no averaging		0.43		$^\circ/\text{sec rms}$
Rate Noise Density	@ 25°C		0.05		$^\circ/\text{sec}/\sqrt{\text{Hz}}$
FREQUENCY RESPONSE					
3 dB Bandwidth (User-Selectable) ⁴	$C_{OUT} = 0\ \mu\text{F}$		40		Hz
Sensor Resonant Frequency			14		kHz
SELF-TEST INPUTS					
ST1 Rateout Response ⁵	ST1 pin from Logic 0 to Logic 1	-121	-221	+376	LSB
ST2 Rateout Response ⁵	ST2 pin from Logic 0 to Logic 1	+121	+221	+376	LSB
Logic 1 Input Voltage	Standard high logic level definition	3.3			V
Logic 0 Input Voltage	Standard low logic level definition			1.7	V
Input Impedance	To common		50		k Ω
TEMPERATURE SENSOR					
Reading at 298 K			2048		LSB
Scale Factor	Proportional to absolute temperature		0.1453		K/LSB
2.5 V REFERENCE					
Voltage Value		2.45	2.5	2.55	V
Load Drive to Ground	Source		100		μA
Load Regulation	$0\ \mu\text{A} < I_{OUT} < 100\ \mu\text{A}$		5.0		mV/mA
Power Supply Rejection	$V_{CC} = V_{DRIVE} = 4.75\text{ V}$ to 5.25 V		1.0		mV/V
Temperature Drift	Delta from 25°C		5.0		mV
LOGIC INPUTS					
Input High Voltage, V_{IH}		$0.7 \times V_{DRIVE}$			V
Input Low Voltage, V_{IL}				$0.3 \times V_{DRIVE}$	V
Input Current, I_{IH}	Typically 10 nA	-1		+1	μA
Input Capacitance, C_{IH}			10		pF
ANALOG INPUTS					
Resolution	For $V_{IN} < V_{CC}$		12		Bits
Integral Nonlinearity		-2		+2	LSB
Differential Nonlinearity		-2		+2	LSB
Offset Error		-8		+8	LSB
Gain Error		-2		+2	LSB
Input Voltage Range		0		$V_{REF} \times 2$	V
Leakage Current		-1		+1	μA
Input Capacitance			20		pF
Full Power Bandwidth			8		MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADIS16100

Parameter	Conditions	Min ¹	Typ	Max ¹	Unit
DIGITAL OUTPUTS					
Output High Voltage, V_{OH}	$I_{SOURCE} = 200 \mu A$	$V_{DRIVE} - 0.2$			V
Output Low Voltage, V_{OL}	$I_{SINK} = 200 \mu A$			0.4	V
CONVERSION RATE					
Conversion Time	16 SCLK cycles with SCLK at 20 MHz			800	ns
Throughput Rate				1	MSPS
POWER SUPPLY					
V_{CC}	All at $T_A = -40^\circ C$ to $+85^\circ C$	4.75	5	5.25	V
V_{DRIVE}		2.7		5.25	V
V_{CC} Quiescent Supply Current	$V_{CC} = 5 V, f_{SCLK} = 50 \text{ kSPS}$		7.0	9.0	mA
V_{DRIVE} Quiescent Supply Current	$V_{DRIVE} = 5 V, f_{SCLK} = 50 \text{ kSPS}$		70	500	μA
Power Dissipation	$V_{CC} = V_{DRIVE} = 5 V, f_{SCLK} = 50 \text{ kSPS}$		40		mW
TEMPERATURE RANGE					
Operation		-40		+85	$^\circ C$

¹ All minimum and maximum specifications are guaranteed. Typical specifications are neither tested nor guaranteed.

² Dynamic range is the maximum full-scale measurement range possible, including output swing range, initial offset, sensitivity, offset drift, and sensitivity drift at 5 V supplies.

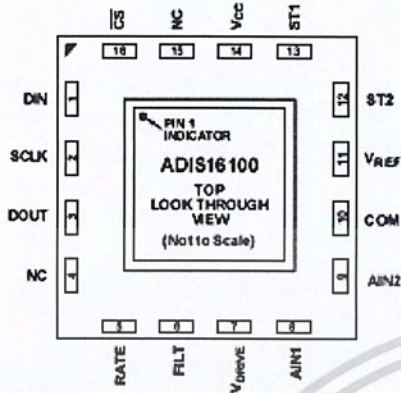
³ Defined as the output change from ambient to maximum temperature or ambient to minimum temperature.

⁴ Frequency at which the response is 3 dB down from dc response. Bandwidth = $1/(2 \times \pi \times 180 \text{ k}\Omega \times (22 \text{ nF} + C_{OUT}))$. For $C_{OUT} = 0 \mu F$, bandwidth = 40 Hz. For $C_{OUT} = 1 \mu F$, bandwidth = 0.87 Hz.

⁵ Self-test response varies with temperature.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
 1. NC = NO CONNECT
 2. THIS IS NOT AN ACTUAL TOP VIEW, AS THE PINS ARE NOT VISIBLE FROM THE TOP. THIS IS A LAYOUT VIEW, WHICH REPRESENTS THE PIN CONFIGURATION, IF THE PACKAGE IS LOOKED THROUGH FROM THE TOP. THIS CONFIGURATION IS PROVIDED FOR PCB LAYOUT PURPOSES.

Figure 4. Pin Configuration, Top Look Through View

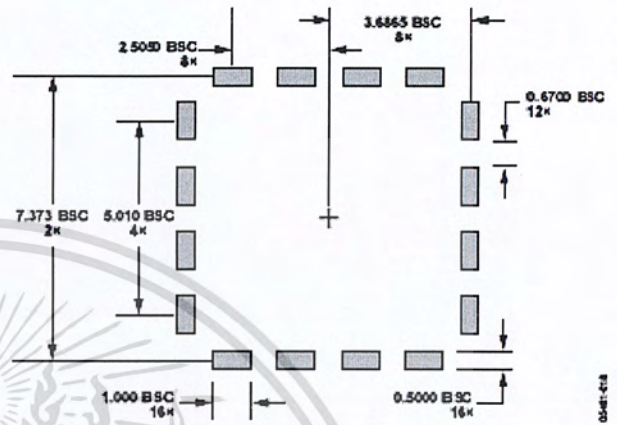


Figure 5. Second-Level Assembly Pad Layout

Table 4. Pin Function Descriptions

Pin No.	Mnemonic	Type ¹	Description
1	DIN	I	SPI Data Input.
2	SCLK	I	SPI Serial Clock.
3	DOUT	O	SPI Data Output.
4	NC		No Connect.
5	RATE	O	Buffered Analog Output. Represents the angular rate signal.
6	FILT	I	External Capacitor Connection to Control Band width.
7	V _{DRIVE}	S	Digital Interface Supply. To simplify interfacing, this can be the receive processing supply of the circuit.
8	AIN1	I	External Analog Input Channel 1. See ADD0 and ADD1 address bits in Table 5.
9	AIN2	I	External Analog Input Channel 2. See ADD0 and ADD1 address bits in Table 5.
10	COM	S	Common. Reference point for all circuitry in the ADIS16100.
11	V _{REF}	O	Precision 2.5 V Reference.
12	ST2	I	Self-Test Input 2.
13	ST1	I	Self-Test Input 1.
14	V _{CC}	S	Analog Power.
15	NC		No Connect.
16	\overline{CS}	I	Chip Select. Active low. This input frames the serial data transfer and initiates the conversion process.

¹ I = Input; O = output; S = power supply.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3-Axis, $\pm 2 g/\pm 4 g/\pm 8 g/\pm 16 g$ Digital Accelerometer

ADXL345

FEATURES

Ultralow power: as low as 40 μA in measurement mode and 0.1 μA in standby mode at $V_s = 2.5 V$ (typical)

Power consumption scales automatically with bandwidth

User-selectable resolution

Fixed 10-bit resolution

Full resolution, where resolution increases with g range, up to 13-bit resolution at $\pm 16 g$ (maintaining 4 mg/LSB scale factor in all g ranges)

Embedded, patent pending FIFO technology minimizes host processor load

Tap/double tap detection

Activity/inactivity monitoring

Free-fall detection

Supply voltage range: 2.0 V to 3.6 V

I/O voltage range: 1.7 V to V_s

SPI (3- and 4-wire) and I²C digital interfaces

Flexible interrupt modes mappable to either interrupt pin

Measurement ranges selectable via serial command

Bandwidth selectable via serial command

Wide temperature range ($-40^{\circ}C$ to $+85^{\circ}C$)

10,000 g shock survival

Pb free/RoHS compliant

Small and thin: 3 mm \times 5 mm \times 1 mm LGA package

APPLICATIONS

Handsets

Medical instrumentation

Gaming and pointing devices

Industrial instrumentation

Personal navigation devices

Hard disk drive (HDD) protection

Fitness equipment

GENERAL DESCRIPTION

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to $\pm 16 g$. Digital output data is formatted as 16-bit two's complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0° .

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm \times 5 mm \times 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

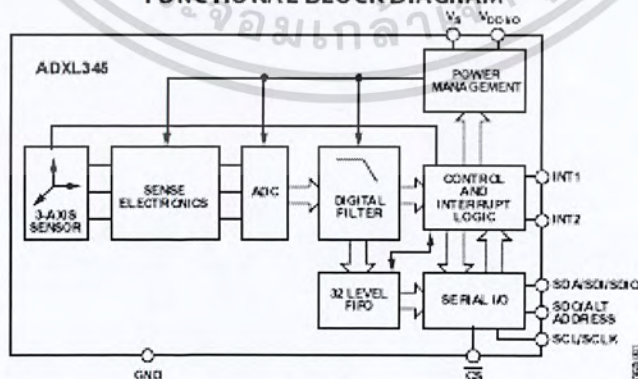


Figure 1.

Rev. 0

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. See the last page for disclaimers.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com

Fax: 781.461.3113

©2009 Analog Devices, Inc. All rights reserved.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPECIFICATIONS

T_A = 25°C, V_S = 2.5 V, V_{DDIO} = 1.8 V, acceleration = 0g, C_G = 1 μF tantalum, C_{IO} = 0.1 μF; unless otherwise noted

Table 1. Specifications¹

Parameter	Test Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis User selectable		±2, ±4, ±8, ±16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ²			±1		%
OUTPUT RESOLUTION					
All g Ranges	Each axis		10		Bits
±2 g Range	10-bit resolution		10		Bits
±4 g Range	Full resolution		11		Bits
±8 g Range	Full resolution		12		Bits
±16 g Range	Full resolution		13		Bits
SENSITIVITY					
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	Each axis		256	286	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±2 g, 10-bit or full resolution	23.2	3.9	4.3	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	116	128	143	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±4 g, 10-bit resolution	7.0	7.8	8.6	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	58	64	71	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±8 g, 10-bit resolution	14.0	15.6	17.2	mg/LSB
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	29	32	36	LSB/g
Scale Factor at X _{OUT} , Y _{OUT} , Z _{OUT}	±16 g, 10-bit resolution	28.1	31.2	34.3	mg/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 g BIAS LEVEL					
0g Output for X _{OUT} , Y _{OUT}	Each axis	-150	±40	+150	mg
0g Output for Z _{OUT}		-250	±80	+250	mg
0g Offset vs. Temperature for x-, y-Axes			±0.8		mg/°C
0g Offset vs. Temperature for z-Axis			±4.5		mg/°C
NOISE PERFORMANCE					
Noise (x-, y-Axes)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.0		LSB rms
Noise (z-Axis)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.5		LSB rms
OUTPUT DATA RATE AND BANDWIDTH					
Measurement Rate ³	User selectable	6.25		3200	Hz
SELF-TEST⁴					
	Data rate ≥ 100 Hz, 2.0 V ≤ V _S ≤ 3.6 V				
Output Change in x-Axis		0.20		2.10	g
Output Change in y-Axis		-2.10		-0.20	g
Output Change in z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V _S)		2.0	2.5	3.6	V
Interface Voltage Range (V _{DDIO})	V _S ≤ 2.5V	1.7	1.8	V _S	V
	V _S ≥ 2.5V	2.0	2.5	V _S	V
Supply Current	Data rate > 100 Hz		145		μA
	Data rate < 10 Hz		40		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time ⁵	Data rate = 3200 Hz		1.4		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C
WEIGHT					
Device Weight			20		mg

¹ All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

² Cross-axis sensitivity is defined as coupling between any two axes.

³ Bandwidth is half the output data rate.

⁴ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (in the DATA_FORMAT register) minus the output (g) when the SELF_TEST bit = 0 (in the DATA_FORMAT register). Due to device filtering, the output reaches its final value after 4 × τ when enabling or disabling self-test, where τ = 1/(data rate).

⁵ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately τ + 1.1 in milliseconds, where τ = 1/(data rate).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADXL345

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration	
Any Axis, Unpowered	10,000 g
Any Axis, Powered	10,000 g
V _S	−0.3 V to +3.6 V
V _{DDIO}	−0.3 V to +3.6 V
Digital Pins	−0.3 V to V _{DDIO} + 0.3 V or 3.6 V, whichever is less
All Other Pins	−0.3 V to +3.6 V
Output Short-Circuit Duration (Any Pin to Ground)	Indefinite
Temperature Range	
Powered	−40°C to +105°C
Storage	−40°C to +105°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

Table 3. Package Characteristics

Package Type	θ_{JA}	θ_{JC}	Device Weight
14-Terminal LGA	150°C/W	85°C/W	20 mg

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.