

อินเวอร์ทเพนดูลัม

INVERTED PENDULUM



โดย

นางสาว กนิษฐา วิจิตรเมฆทอง รหัส 38014004

นาย พงษ์ ณีรุพล พากเพียร รหัส 38014145

นาย นวเทพ ชนาพรรณ รหัส 38014232

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขที่.....

เลขทะเบียน 33932

วัน, เดือน, ปี 20 ก.ย. 2542

ข้าพเจ้าขอแจ้งให้ทราบว่า ข้าพเจ้าขอรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเวอร์ทเพนดูลัม
INVERTED PENDULUM

โดย

นางสาว กนิษฐา วิจิตรเมฆทอง รหัส 38014004

นาย ณัฐพล พากเพียร รหัส 38014145

นาย นวเทพ ชนาพรธณ รหัส 38014232

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.จกมล งามวิวิทย์

อาจารย์สุมิตร พนาอุดมทรัพย์

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญาโทปีการศึกษา 2541

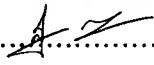
ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อินเวิร์ทเพนดูลัม
INVERTED PENDULUM

ผู้จัดทำ กนิษฐา วิจิตรเมฆทอง
ณัฐพล พากเพียร
นวเทพ ชนาพรรณ

.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.จงกล งามวิวิทย์)

.....อาจารย์ที่ปรึกษา
(อาจารย์สุมิตร พนาอุคมทรัพย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเวอร์ทเพนดูลัม
INVERTED PENDULUM

โดย

นางสาวกนิษฐา วิจิตรเมฆทอง

รหัสประจำตัวนักศึกษา 38014004

นายณัฐพล พากเพียร

รหัสประจำตัวนักศึกษา 38014145

นายนวเทพ ชนาพรธม

รหัสประจำตัวนักศึกษา 38014232

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.จกต งามวิวิทย์

อาจารย์สุมิตร พนาอุดมทรัพย์

บทคัดย่อ

ปริญญานิพนธ์นี้ได้กล่าวถึงการควบคุมและการออกแบบอินเวอร์ทเพนดูลัมที่จะทำให้ก้าน
ลูกตุ้มทรงตัวได้อย่างสมดุลย์ โดยได้ทำการหาโมเดลทางคณิตศาสตร์ของระบบจากโครงสร้างที่ได้
ประกอบขึ้น และทำการควบคุมระบบด้วยโปรแกรมคอมพิวเตอร์ สำหรับผลการทดลองได้นำ
เสนอไว้ในปริญญานิพนธ์ฉบับนี้

ABSTRACT

This thesis describes the design of inverted pendulum system and control algorithms to
control the vertical balance of rotational inverted pendulum. The mathematical model of the
system is determined from its structure and control the system by computer. The experimental
results also are shown in the thesis.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลงได้ด้วยดี เพราะได้รับความเมตตาจาก รองศาสตราจารย์ ดร.จกมล งามวิวิทย์ และ อาจารย์สุมิตร พนาอุดมทรัพย์ อาจารย์ที่ปรึกษา ที่ได้ให้ความกรุณาแนะนำ พร้อมด้วยอุปกรณ์ในการทำงานแก่ผู้จัดทำตลอดมา คณะผู้จัดทำขอกราบขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย และยังขอขอบพระคุณคณาจารย์ทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้แก่คณะผู้จัดทำ

ขอกราบขอขอบพระคุณ คุณพ่อ คุณแม่ ของผู้จัดทำที่ได้อุปการะผู้จัดทำ ทั้งยังเป็นผู้ให้กำลังใจแก่ผู้จัดทำตลอดมา

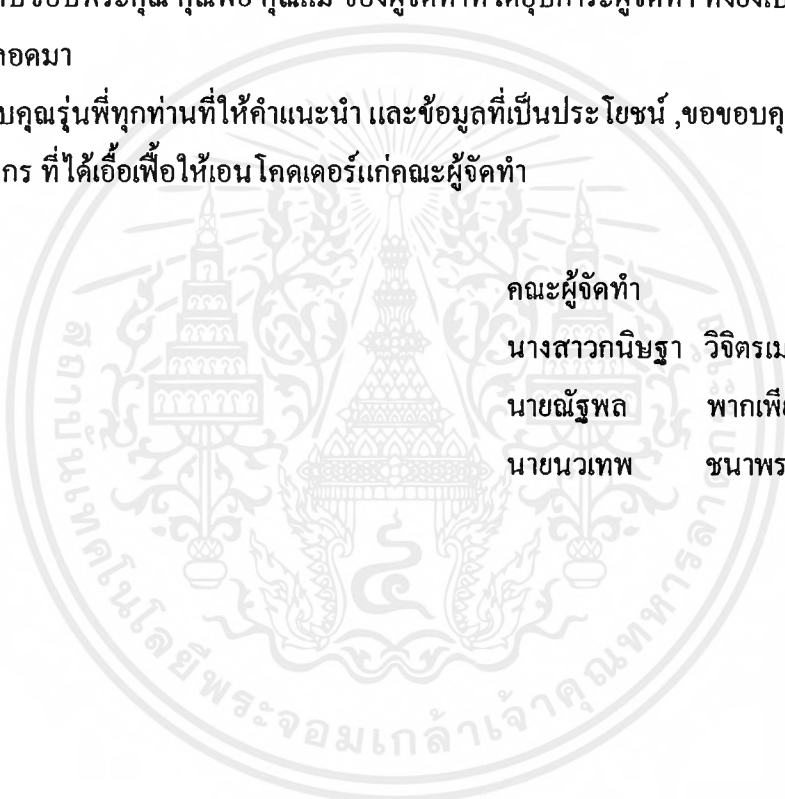
ขอขอบคุณรุ่นพี่ทุกท่านที่ให้คำแนะนำ และข้อมูลที่เป็นประโยชน์ ,ขอขอบคุณเพื่อนคือนายคอน อิศรากร ที่ได้เอื้อเพื่อให้เอน โคคเคอร์แก่คณะผู้จัดทำ

คณะผู้จัดทำ

นางสาวกนิษฐา วิจิตรเมฆทอง

นายณัฐพล พากเพียร

นายนวเทพ ชนาพรรณ



สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 จุดประสงค์	1
1.2 ประเภทของอินเวิร์ทเพนคูล์ม	1
1.3 หลักการทั่วไป	1
1.4 โครงสร้างระบบโดยรวม	2
บทที่ 2 ส่วนประกอบสำคัญของระบบ	3
2.1 โครงสร้างทางเครื่องกล	3
2.2 การติดต่อกับเครื่องคอมพิวเตอร์	5
2.3 วงจรอิเล็กทรอนิกส์	7
2.4 ส่วนควบคุมและประมวลผล	9
บทที่ 3 รูปแบบทางคณิตศาสตร์และการออกแบบระบบควบคุม	13
3.1 สมการและโมเดลของระบบ	13
3.2 การออกแบบระบบควบคุม	17
บทที่ 4 ผลการทดลอง	21
4.1 การหาพารามิเตอร์ของระบบ	22
4.2 การทดลองหาฟังก์ชันถ่ายโอนของวงจรควบคุมแรงดัน	25
4.3 การทดลองระบบหลังการควบคุม	31
บทที่ 5 สรุปและวิเคราะห์ผลการดำเนินการ บรรณานุกรม ภาคผนวก	35

สารบัญรูปภาพ

รูป	หน้า
รูป 1.1 : การจัดระบบควบคุมการตั้งของลูกตุ้ม	2
รูป 2.1 : แสดงโครงสร้างทางเครื่องกล	5
รูป 2.2 : การแสดงตำแหน่งของการเลือกคิปสวิทช์	5
รูป 2.3 : แสดงวงจรการทำงานของ IC74HC688ร่วมกับคิปสวิทช์	6
รูป 2.4 : แสดงโครงสร้างภายในส่วนของPIC16C55และ16C56	7
รูป 2.5 : สัญญาณที่ได้จากเอน โทคเตอร์	7
รูป 2.6 : ไดอะแกรมสถานที่เขียนได้จากรูป 2.5	8
รูป 2.7 : แสดงวงจรคาร์ดิอินเตอร์เฟส	10
รูป 2.8 : วงจรควบคุมแรงคั้นมอเตอร์กระแสตรง	11
รูป 2.9 : แหล่งจ่ายไฟตรง 50 โวลท์	12
รูป 3.1 : บล็อกไดอะแกรมของระบบทั้งหมด	13
รูป 3.2 : แสดงแบบจำลองของมอเตอร์และแกน	14
รูป 3.3 : แสดงแบบจำลองของลูกตุ้ม	15
รูป 3.4 : แสดงแบบจำลองโมเดลของระบบ	17
รูป 3.5 : ระบบก่อนการควบคุม	19
รูป 3.6 : ระบบหลังการควบคุม	19
รูป 3.7 : บล็อกไดอะแกรมของระบบหลังถูกควบคุม	20
รูป 4.1 : ผลที่ได้จากการเขียนแบบโดยโปรแกรมMATLABเพื่อหาค่า J_m และ b_1	22
รูป 4.2 : แสดงแบบจำลองการทดลองหาค่า J_2 และ b_2	23
รูป 4.3 : ผลที่ได้จากการทดลองเพื่อหาค่า J_2 และ b_2	25
รูป 4.4 : กราฟของ $\theta_1(t)$ ที่ได้จากการคำนวณด้วยคอมพิวเตอร์	26
รูป 4.5 : กราฟของ $\dot{\theta}_1(t)$ ที่ได้จากการใช้โปรแกรมMATLAB	27
รูป 4.6 : กราฟของ $\ddot{\theta}_1(t)$ ที่ได้จากการคำนวณด้วยคอมพิวเตอร์	29
รูป 4.7 : กราฟของ $\dot{\theta}_1(t)$ ที่ได้จากการใช้โปรแกรมMATLAB	30
รูป 4.8 : กราฟแสดงผลตอบสนองของระบบหลังการควบคุมแบบPID	32
รูป 4.9 : กราฟแสดงผลตอบสนองของระบบหลังการควบคุมแบบป้อนกลับ ของ $\theta_1(t)$ หรือแกน	33
รูป 4.10 : กราฟแสดงผลตอบสนองของระบบหลังการควบคุมแบบป้อนกลับ ของ $\theta_2(t)$ หรือก้านลูกตุ้ม	34

บทที่ 1

บทนำ

การควบคุมเสถียรภาพของอินเวอร์ทเพนดูลัมเป็นชุดควบคุมก้านลูกตุ้ม (Pendulum) บน แขน ซึ่งจากเดิมเป็นระบบที่ไม่เสถียร การที่จะทำระบบให้เสถียรนั้น เราต้องมีการควบคุม

1.1 จุดประสงค์

1. ศึกษาทฤษฎีของการควบคุมเสถียรภาพและทดลองภายใต้การควบคุมแบบ PID
2. นำความรู้ที่ได้รับจากการศึกษา และทดลองไปเป็นพื้นฐานในการศึกษา พร้อมทั้งการประยุกต์ใช้ในอนาคต
3. ทราบถึงปัญหาที่เกิดขึ้น และแนวทางแก้ไข

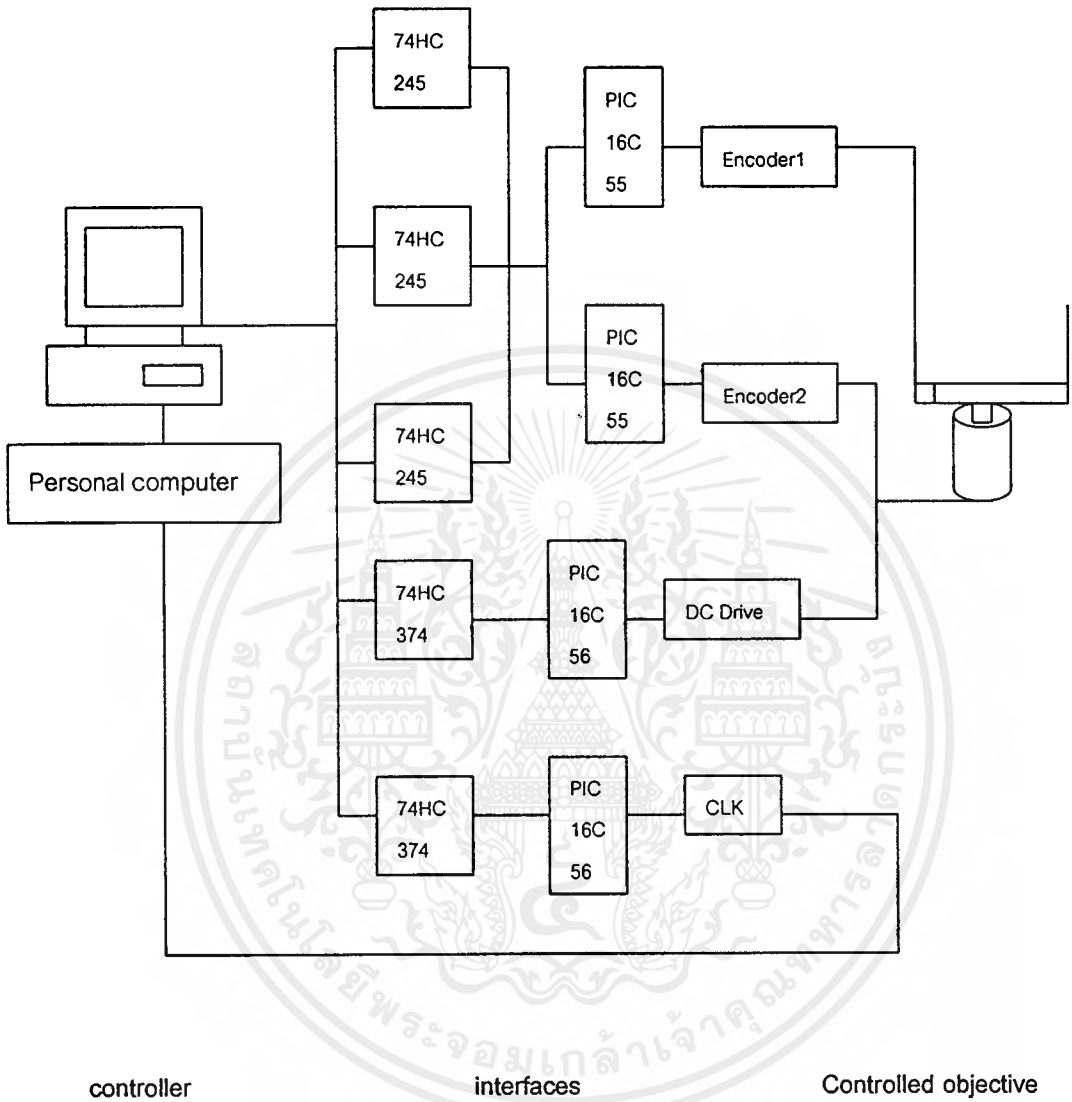
1.2 ประเภทของอินเวอร์ทเพนดูลัม

ที่ใช้ทำการศึกษามีสองประเภทใหญ่ๆ คือ แบบรถ (cart and stick) และแบบประยุกต์ ซึ่งแบบประยุกต์ที่ได้ทำการค้นคว้าคือแบบหมุน ซึ่งมีข้อดีคือ ประหยัดเนื้อที่ และประมาณการคำนวณเป็นเชิงเส้นได้

1.3 หลักการทั่วไป

ส่วนประกอบทั้งหมดได้แก่ ชุดลูกตุ้ม (rotational inverted pendulum) วงจรติดต่อ คอมพิวเตอร์ (interface circuit) และชุดควบคุม (controller) เราควบคุมการตั้งของก้านลูกตุ้มด้วย ตำแหน่งแขนซึ่งจะยึดติดกับเพลลาของมอเตอร์ โดยแกนมอเตอร์จะมีตัววัดตำแหน่งแบบแสง (optical encoder) ติดอยู่เพื่อวัดมุมของแขน (θ_1) เพลลาของมอเตอร์ และแขนจะหมุนได้รอบส่วนตัว มอเตอร์ จะยึดแน่นกับฐาน ทางด้านก้านลูกตุ้ม จะวัดมุม (θ_2) ได้ โดยตัววัดตำแหน่งแบบแสง เช่นเดียวกัน โดยระบบทั้งหมดจะต้องรักษาสภาพสมดุลของก้านลูกตุ้มไว้ ($\theta_2 \Rightarrow 0$)

1.4 โครงสร้างระบบโดยรวม



รูปที่ 1.1 การจักระบบควบคุมการตั้งของลูกตุ้ม

บทที่ 2

ส่วนประกอบสำคัญของระบบ

เมื่อเราพิจารณาโครงสร้างของระบบต้องมีวงจรสำหรับควบคุมสัญญาณจากอินพุทของระบบสำหรับมอเตอร์ และเราได้ตัดสินใจเลือกวงจรควบคุมแรงดันที่จ่ายให้แก่มอเตอร์ซึ่งเป็นการควบคุมความเร็ว เนื่องจากเป็นวงจรที่เราพบ โดยส่วนมาก และเคยได้ทดลองใช้งานใน pre-project มาแล้ว จึงสะดวกมากกว่า ที่จะทำวงจรควบคุมกระแสซึ่งมีความยุ่งยากกว่ามาก

เมื่อเราได้วงจรควบคุมแรงดันซึ่งเป็นอินพุทของระบบแล้ว ต่อไปเราจึงพิจารณาตัวแปรสเกลซึ่งเป็นค่าที่เราจะต้องวัดจากระบบ ณ เวลาต่างๆ ได้แก่ ค่า θ_1 , $\dot{\theta}_1$, $\ddot{\theta}_1$, θ_2 , $\dot{\theta}_2$ สำหรับตัวแปร $\dot{\theta}_1$ และ $\dot{\theta}_2$ นั้น เราสามารถใช้โปรแกรมคอมพิวเตอร์หาค่าได้จาก อัตราการเปลี่ยนแปลงของมุมเท่ากับมุมที่เปลี่ยนไปหารด้วยเวลา คือเอาค่าปัจจุบันลบด้วยค่าในอดีตแล้วหารด้วยคาบเวลาในการสุ่ม (sampling period) และในทำนองเดียวกันกับตัวแปร $\ddot{\theta}_1$ คือ อัตราการเปลี่ยนแปลงของความเร็วของมอเตอร์ เราจึงเลือกใช้ตัววัดเป็นเอนโคเดอร์ที่ติดมากับมอเตอร์ ได้มุมที่เปลี่ยนไปอยู่ในรูปของจำนวนพัลส์ ซึ่งจะต้องผ่านวงจรนับอีกทีจึงจะใช้ได้ ส่วนมุม θ_2 เป็นมุมที่ข้อต่อระหว่างชุดปลายแขนของเพลากับก้านลูกตุ้มทำกับแนวตั้ง เราจึงใช้เอนโคเดอร์อีกตัวหนึ่งมาเป็นตัววัดมุม ได้มุมที่เปลี่ยนไปอยู่ในรูปของจำนวนพัลส์ เช่นกัน

สำหรับวงจรควบคุมแรงดันนั้นก่อนที่เราจะออกแบบเราจะต้องกำหนดลักษณะการทำงานที่เราต้องการก่อน โดยการพิจารณาจากขนาดพิกัดของมอเตอร์ที่เราจะใช้ คือเราสามารถที่จะใช้ค่าความเร็วต่างๆ ได้ ในช่วงที่ไม่เกินพิกัดของมอเตอร์ และสามารถควบคุมให้เคลื่อนที่ได้ 2 ทิศทางอย่างรวดเร็ว เมื่อได้ค่าการทำงานและลักษณะการทำงานของวงจรควบคุมแล้ว จึงหาทางต่อวงจรให้ได้ตามต้องการ

ระบบทั้งหมดแบ่งออกเป็น 4 ส่วน ได้แก่

1. โครงสร้างทางเครื่องกล
2. การติดต่อกับเครื่องคอมพิวเตอร์
3. วงจรอิเล็กทรอนิกส์
4. ส่วนควบคุมและประมวลผล

2.1 โครงสร้างทางเครื่องกล

เมื่อเราได้หัวข้อของโครงงาน เราได้เริ่มทำการศึกษารูปแบบต่างๆ ของอินเวอร์ทเพนดูลัมเปรียบเทียบข้อดี ข้อเสีย และความเป็นไปได้ เราได้เลือกอินเวอร์ทเพนดูลัมแบบหมุนได้ ซึ่งข้อดีคือ

ชุดอินเวิร์ทเพนดูลัมดังกล่าวสามารถหมุนได้รอบจึงไม่การสิ้นสุดของระยะทาง และใช้พื้นที่น้อย ในการติดตั้งขณะทำงาน พร้อมทั้งความง่ายของโครงสร้างทางเครื่องกลด้วย

ส่วนอุปกรณ์สำคัญที่ใช้ได้แก่ มอเตอร์กระแสตรงมีเอน โคคเคอร์ติด และเอน โคคเคอร์ที่ยึด ติดกับก้านลูกตุ้ม เราจึงต้องหาอุปกรณ์ดังกล่าว แล้ววัดขนาด น้ำหนัก และรูปแบบต่างๆ แล้วจึงทำ โครงสร้าง เพื่อให้มีความเหมาะสมเป็นระเบียบ แข็งแรง และสวยงาม

จากการศึกษาคุณสมบัติของโลหะต่างพบว่า อลูมิเนียม มีน้ำหนักเบา ทนทาน ไม่เป็นสนิม และง่ายต่อการทำชิ้นงาน ดังนั้นเราจึงเลือกอลูมิเนียมในการทำโครงสร้างในส่วนของแขนซึ่งจะยึด ติดกับเพลลาของมอเตอร์ ในส่วนของฐานและเสาที่ยึดกับมอเตอร์ ต้องมีความแข็งแรง มั่นคงและ สามารถรับน้ำหนักของส่วนประกอบต่างๆ ในโครงสร้างพร้อมกับมอเตอร์ จึงเลือกเหล็กต้นในการ ทำส่วนโครงสร้างนี้ ในส่วนของก้านลูกตุ้มเราเลือกใช้ก้านธนู ด้วยรูปทรงที่ตรงตลอดทั้งความยาว ก้าน น้ำหนักเบา และมีความคงทน เกิดการ โค้งงอเปลี่ยนรูปทรง ได้ยาก พร้อมทั้งเลือกใช้ชนิดที่มีความ แข็ง เพื่อใช้ในการประกอบชิ้นงานเข้าด้วยกัน

ในการทำและประกอบชิ้นงานเราเน้นที่ความแข็งแรง ความเป็นไปได้ และความสวยงามจึง เสียเวลาพอสมควรแต่เมื่อชิ้นงานออกมาและประกอบเสร็จก็เป็นที่น่าพอใจ ดังรูปที่ 2.1 จะเห็น ได้ว่าเราออกแบบความยาวของก้านลูกตุ้ม ให้น้อยกว่าความสูงของฐานทั้งหมด เพื่อป้องกันลูกตุ้ม กระทบกับพื้นซึ่งจะเป็นอันตรายต่อเอน โคคเคอร์ตัวที่ยึดกับก้านลูกตุ้มอย่างมาก

ลักษณะและการประกอบอุปกรณ์

ฐาน เราใช้แผ่นเหล็กขนาด 20x20 cm และยึดแผ่นกับเสา โดยนอต

เสา เป็นเหล็กต้นขนาด \varnothing 0.75 นิ้ว สูง 1.5 ฟุต ทำรูขั้วนอตตรงกลางทั้งสี่เสายึดให้แน่น กับฐานแทนยึดมอเตอร์

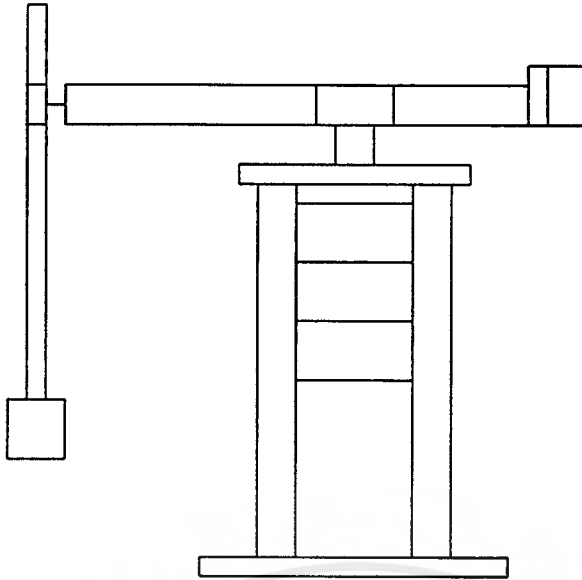
แทนยึดมอเตอร์ เป็นเหล็กขนาด 13x13 cm ยึดตัวมอเตอร์ให้แข็งแรงแล้วจึงนำมายึดกับเสา อีกครั้งหนึ่งอย่างแน่นหนา โดยส่วนของเพลลามอเตอร์จะสูงเหนือแทนยึดมอเตอร์ขึ้นมาส่วนที่ปลาย มอเตอร์จะมีเอน โคคเคอร์ติดอยู่

ตัวยึดเพลลามอเตอร์กับแขน ใช้เหล็กขนาด 6x8x3.5 cm โดยยึดเพลลาของมอเตอร์แบบบีบ รัคจะมีความแข็งแรงมาก โดยส่วนนี้สามารถขันปรับน็อตตัวบนเพื่อปรับความยาวของแขนได้

แขน เป็นอลูมิเนียมสี่เหลี่ยมจัตุรัสกลวงบางขนาด 2.5x2.5x66.5 cm โดยแขนนี้จะมีน้ำ หนักเบาและแข็งแรงจะส่งแรงจากเพลลามอเตอร์ไปยังตำแหน่งฐานของก้านลูกตุ้ม

ส่วนยึดฐานของก้านลูกตุ้ม ที่ปลายของแขนจะมีที่ยึดแขนกับฐานก้านลูกตุ้มจะมีเอน โคค เคอร์ติดอยู่ เพื่อวัดมุมเปลี่ยนแปลง

ก้านลูกตุ้ม เป็นแท่งก้านธนูกลมกลวงขนาด \varnothing 0.6 cm น้ำหนักเบาที่ปลายของก้านลูกตุ้ม จะมีลูกตุ้มติดอยู่



รูปที่ 2.1 แสดง โครงสร้างทางเครื่องกล

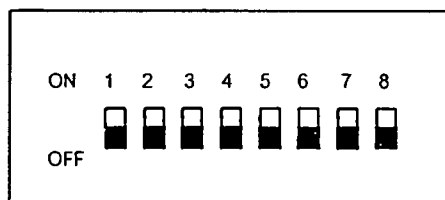
2.2 การติดต่อกับเครื่องคอมพิวเตอร์

วงจรอินเทอร์เฟสดังรูปที่ 2.7 เป็นวงจรที่ออกแบบขึ้นมาเพื่อใช้ในการติดต่อระหว่างวงจรควบคุมอุปกรณ์ภายนอกกับเครื่องคอมพิวเตอร์ ซึ่งประกอบด้วย 3 ส่วนคือ

2.2.1 วงจร decode จาก Address Bus เพื่อใช้ในการเลือกรับหรือส่งข้อมูล

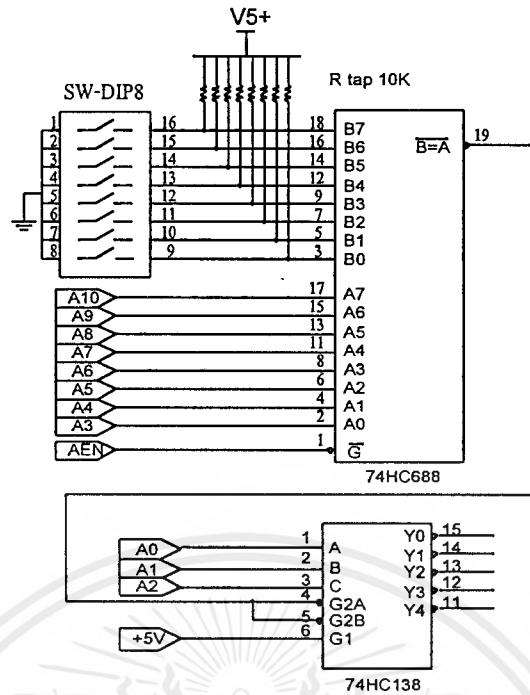
การติดต่อกับเครื่องคอมพิวเตอร์นั้นใช้การติดต่อพอร์ตแบบตายตัว (fix decode port) ดังนั้น อุปกรณ์ IC74HC688 จึงถูกนำมาใช้งานร่วมกับคิปสวิทช์ ทำให้เราสามารถเลือกใช้แอดเดรสเพื่อใช้ติดต่อกับวงจรควบคุมได้จากตำแหน่งการปิด เปิด ของคิปสวิทช์

การกำหนดแอดเดรสจะถูกกำหนดโดยขา A0-A10 ดังนั้นในการติดต่อกับวงจรควบคุม ถ้าขาสัญญาณคิปสวิทช์ตรงกับขาสัญญาณ IC 74HC688 (A3-A10) จะทำให้เกิดการทำงานของ IC 74HC138 ซึ่งจะ decode A0-A2 ทำให้เราสามารถติดต่อเพิ่มขึ้นได้เป็น 8 ทาง โดยการอ้างจาก A0-A2 แต่ในวงจรนี้เราต้องการใช้ติดต่อกับ IC74HC245 3 ตัว และ IC 74HC374 2 ตัว ดังวงจรรูปที่ 2.3



A10 A9 A8 A7 A6 A5 A4 A3

รูปที่ 2.2 การแสดงตำแหน่งของการเลือกคิปสวิทช์



รูปที่ 2.3 แสดงวงจรการทำงานของ IC 74HC688 ร่วมกับคิปสวิตช์และ IC 74HC138

2.2.2 Multiplex Data Bus ซึ่งจะแยกเป็น 2 ส่วนคือ Input Port และ Output Port

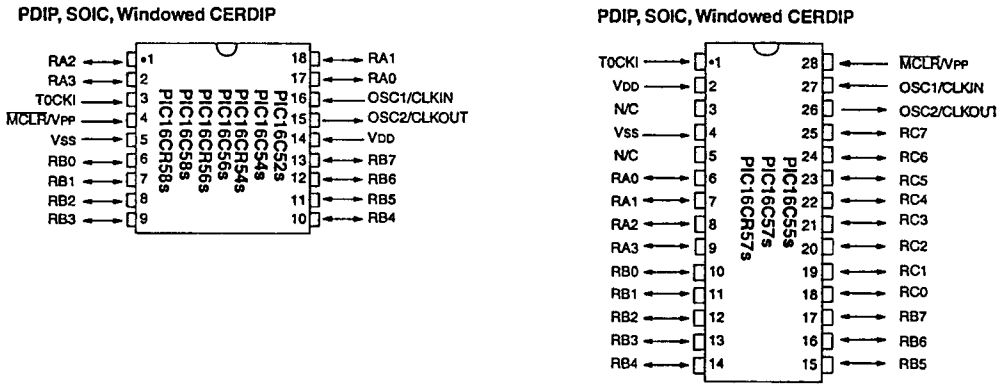
โดยที่ Input Port จะใช้ 74HC245 ซึ่งเป็น Tri-state Buffer 8 bit 3 ตัว รวมเป็น 24 บิต เพื่อรับข้อมูลจากเอนโคเดอร์ทั้ง 2 ชุด และ Output Port จะใช้ 74HC374 ซึ่งเป็นวงจร Latch 8 bit 3 ตัว รวมเป็น 24 บิต แต่เราจะใช้เพียง 20 บิตเท่านั้น โดยที่ชุดแรกจะใช้ 8 บิตในการส่งข้อมูลความเร็วมอเตอร์ 7 บิต และทิศทางอีก 1 บิต ชุดที่ 2 ใช้ 8 บิตในการส่งข้อมูลในการกำหนดสัญญาณ Clock ที่จะใช้เป็นสัญญาณ Sampling ข้อมูล ชุดที่ 3 ใช้ 4 บิตในการ Enable หรือ Disable Microcontroller

2.2.3 Microcontroller บนการ์ด Interface จะมีทั้งหมด 2 เบอร์

คือ PIC 16C55 จำนวน 2 ตัว ซึ่งแต่ละตัวใช้ในการรับสัญญาณ Phase A,B จาก Encoder แล้วทำการนับขึ้นหรือลงตามทิศทางการหมุนของ Encoder แล้วส่งค่าที่นับได้ออกสู่ Port 12 bit ซึ่งจะมีการทำงานดังหัวข้อที่ 2.3.1

PIC 16C56 จำนวน 2 ตัว ตัวแรกจะทำหน้าที่เป็น D/A Converter โดยรับข้อมูล 7 บิต แล้วแปลงเป็นสัญญาณ PWM ในการส่งไปควบคุมความเร็วมอเตอร์ ตัวที่ 2 จะรับข้อมูล 8 บิต แล้วแปลงเป็นสัญญาณ Clock ที่ความถี่ตามข้อมูลที่ได้รับ แล้วส่งสัญญาณ Clock กลับไปที่ขา IRQ 5 ของ Personal Computer เพื่อใช้เป็นความถี่ในการ Sampling ข้อมูล

ในส่วนของโปรแกรมวงจรมับและ PWM แสดงอยู่ในภาคผนวก



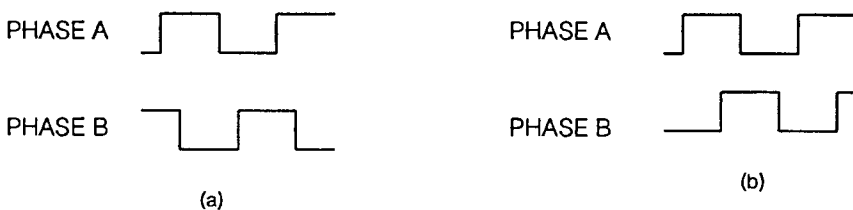
รูปที่ 2.4 แสดงโครงสร้างภายในส่วนของ PIC 16C55 และ 16C56

2.3 วงจรอิเล็กทรอนิกส์

2.3.1 เอนโคเดอร์และวงจรมับ

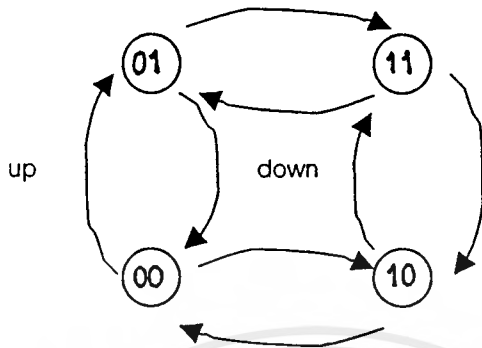
เอนโคเดอร์จะสร้างสัญญาณพัลส์ที่แปรผันตรงกับการหมุนของเพลาของมอเตอร์ซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วของเพลาของมอเตอร์ในรูปของอัตราจำนวนพัลส์หรือ การรับรู้ตำแหน่งโดยการนับจำนวนพัลส์ได้โดยสำหรับเอนโคเดอร์ที่ใช้ทั้ง 2 ตัวมีความละเอียด 1000 พัลส์ต่อรอบ เพราะฉะนั้นการเปลี่ยนแปลง 1 พัลส์ จะเท่ากับ $360/1000 = 0.36$ องศา แต่เนื่องจากเราสามารถเขียน โปรแกรมให้ทำการนับทุกครั้งที่มีการเปลี่ยนแปลงสถานะค่าเฟส A และ B ซึ่งมีอยู่ด้วยกัน 4 สถานะทำให้วัดค่าได้ถึง 4000 พัลส์ สำหรับขาสัญญาณที่ต้องต่อให้กับเอนโคเดอร์ตัวนี้จะมี 4 ขา คือ ขาไฟเลี้ยง 5V, ขากราวด์, ขาเอาต์พุต A และ B ซึ่งขา A และขา B นี้จะมีเอาต์พุตเป็นพัลส์ที่มีขนาดต่ำ เวลาใช้งานจะต้องมีการต่อเข้ากับตัวต้านทานและ ไฟเลี้ยง 5V เพื่อการดึงสัญญาณ (pull up) ส่วนการใช้งานจะดูการนำและการตามของพัลส์ที่ได้จากขาทั้งสองนี้ เป็นการตรวจสอบทิศทางการหมุนของมอเตอร์ ดังนั้นเราจึงนำขาทั้ง 2 ขานี้มาต่อป้อนให้กับวงจรมับเพื่อการควบคุมการนับขึ้นและนับลงได้

การต่อวงจรในส่วนนี้เราจำเป็นต้องทราบการทำงานของเอนโคเดอร์แบบใช้แสงซึ่งจะให้เอาต์พุตออกมาเป็นคลื่นรูปเหลี่ยม 2 สัญญาณ คือ เฟส A และเฟส B โดยตัวตรวจวัดวางให้ค่าเฟสทั้งสองต่างกัน 90 องศา เพื่อใช้บอกทิศทางการหมุนดังรูปที่ 2.5



รูปที่ 2.5 สัญญาณที่ได้จากเอนโคเดอร์ (a) นับขึ้น (b) นับลง

จากรูปที่ 2.5 เราสามารถนำมาเขียนเป็นไคอะแกรมสถานะ (State Diagram) เพื่อวิเคราะห์ ออกแบบได้ดังรูปที่ 2.6



รูปที่ 2.6 ไคอะแกรมสถานะที่เขียนได้จากรูปที่ 2.5

ในตอนแรกที่เราทำการออกแบบวงจร เราพิจารณาแบบง่ายๆ คือ เมื่อสัญญาณจากเฟส A เข้ามา 1 ลูก ก็ให้วงจรนับนับหนึ่งครั้ง โดยพิจารณาจากการเปลี่ยนแปลงสถานะค่าเฟส A และ B ว่าเป็นไปในทิศทางใดของรูป 2.6 เช่นถ้าสถานะเดิมเป็น 01 แล้ว หากสถานะถัดมาเปลี่ยนไปเป็น 10 แล้ว ก็จะนับขึ้น แต่ถ้าหากสถานะถัดมาเปลี่ยนไปเป็น 00 แล้ว ก็ให้นับลงแทน เป็นต้น

สำหรับในที่นี้เราใช้ PIC 16C55 ซึ่งเป็นไมโครคอนโทรลเลอร์ชนิดหนึ่ง ทำการนับพัลส์ จากเอนโคเดอร์ทั้ง 2 ตัว โดยการเขียนโปรแกรมให้ทำการนับทุกครั้งที่มีการเปลี่ยนแปลงของสถานะค่าเฟส A และ B ซึ่งมีอยู่ด้วยกัน 4 สถานะ ทำให้มีสัญญาณนับ 4 ครั้ง ด้วยจากสัญญาณจากเอนโคเดอร์ 1 คาบ ซึ่งเอนโคเดอร์แต่ละตัวมีความละเอียด 1000 พัลส์ ทำให้สามารถนับได้ถึง 4000 ค่า ก็คือใช้ PIC 12 bit (4096) ก็เพียงพอต่อความต้องการแล้วเราใช้ 74HC245 3 ตัวในการ Multiplex สายสัญญาณจำนวน 24 บิต จากวงจรนับสัญญาณเอนโคเดอร์ทั้ง 2 ชุด ส่งไปให้กับคอมพิวเตอร์ เพื่อทำการประมวลผลผ่านทาง data bus d0-d7 ซึ่งมีความสะดวกมากในการใช้งาน วงจรมีขนาดเล็ก ไม่ต้องใช้ D/A, A/D และสามารถ Reset ค่าที่นับได้โดยการ Reset Microcontroller

2.3.2 วงจรควบคุมแรงดันมอเตอร์กระแสตรง

มอเตอร์ที่เราใช้เป็นมอเตอร์กระแสตรง ซึ่งง่ายต่อการควบคุม มีขนาดพิกัดที่แรงดัน 100V แรงบิด 8.8 LB-IN เพื่อมอเตอร์สามารถให้แรงได้เพียงพอต่อความต้องการของระบบ

เนื่องจากลักษณะของมอเตอร์ที่ใช้มีการเปลี่ยนแปลงอย่างรวดเร็ว เราจึงเลือกใช้อุปกรณ์พวก Mosfet ซึ่งสามารถทำงานลักษณะนี้ได้ดี ด้วยระบบของเราใช้แหล่งจ่ายไฟตรง 50V ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ทำกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิกัดที่สูงมาก มีข้อจำกัด และปัญหายุ่งยากมาก ต้องพิจารณาออกแบบวงจรที่เหมาะสมถึงจะใช้งานได้ จากการศึกษาที่เราได้ทดลองใช้วงจรควบคุมกับระบบจึงได้พบกับปัญหาต่างๆและทำการแก้ไขปัญหาเหล่านั้นอยู่หลายครั้งจนได้วงจรควบคุมแรงดันที่เหมาะสม ด้วยข้อจำกัดในการหาอุปกรณ์ไม่ได้ ต้องใช้อุปกรณ์อื่นที่มีคุณสมบัติใกล้เคียงทดแทนกัน และข้อจำกัดในคุณสมบัติของอุปกรณ์บางตัวในการใช้งานจึงต้องเพิ่มอุปกรณ์บางอย่างเข้าไปในวงจร เช่น การใช้ Mosfet MTP 50N06 แทน MTP 50N06EL จึงจำเป็นต้องคิดแปลงวงจรขับ Mosfet ใหม่ โดยเพิ่มแรงดันสำหรับจ่ายให้วงจร Logic ให้สูงกว่าเดิม แล้วเปลี่ยนอุปกรณ์ Logic ทั้งหมดเป็น High speed CMOS ระบบจำเป็นต้องมีการเปลี่ยนแปลงทิศทางการหมุนของมอเตอร์อย่างรวดเร็วตลอดเวลา โดยเราจะใช้เพียงแค่แหล่งจ่ายไฟ แค่เพียง ไฟบวกกับกราวด์(เราจะไม่ใช่ไฟลบ) จึงใช้การต่อวงจรในลักษณะ H-Bridge

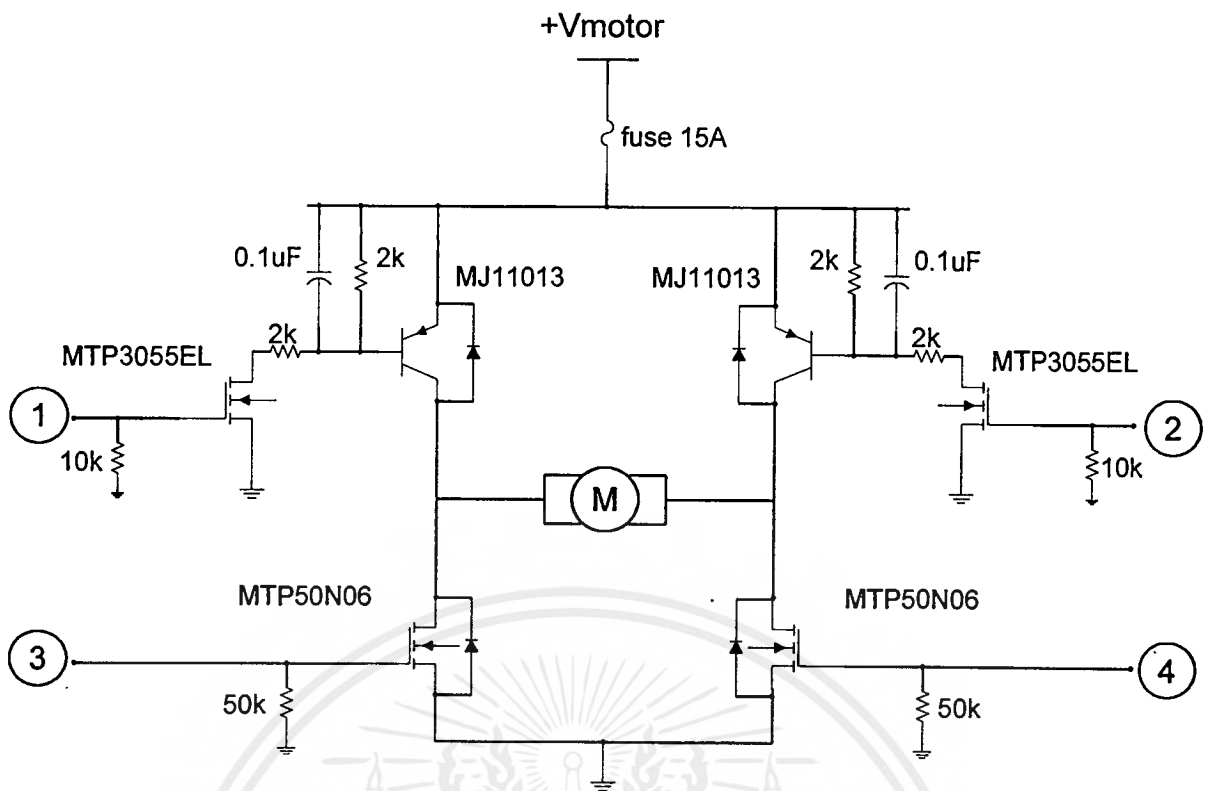
การทำงานของวงจร คือ วงจรจะมีขาสัญญาณ Input อยู่ 2 ขาคือ Direct และ PWM ซึ่งจะกระทำขบวนการทาง Logic ดังรูป 2.8 ภายในวงจร Logic จะประกอบไปด้วยวงจร 2 ส่วน

ส่วนที่ 1 วงจรขับ Motor แบบ H-Bridge ซึ่งมีคุณสมบัติในการควบคุมความเร็วและทิศทางของ Motor

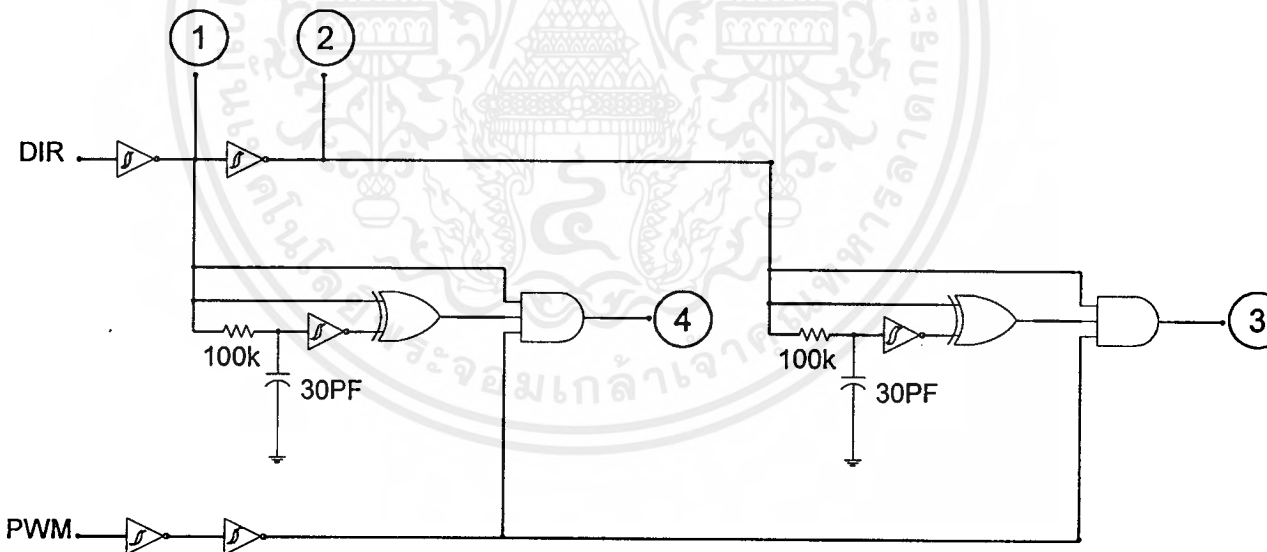
ส่วนที่ 2 ทำหน้าที่เปลี่ยนสัญญาณ Input ทั้ง 2 ไปเป็นสัญญาณที่จะนำไปขับ Mosfet ทั้ง 4 ตัว เพื่อกำหนดทิศทางของกระแสที่ไหลผ่านมอเตอร์และปริมาณแรงดันเฉลี่ยที่ตกคร่อมมอเตอร์ แล้วผ่านวงจร Delay สัญญาณเมื่อมีการเปลี่ยนทิศทางการไหลของกระแสมอเตอร์ เนื่องจากวงจร H-Bridge ที่ใช้ขับมอเตอร์มีส่วนประกอบของ Transistor ซึ่งมีคุณสมบัติให้การ Switching ต่ำกว่า Mosfet จะทำงานตัดสัญญาณ PWM เมื่อช่วงขณะที่มีการเปลี่ยนทิศทางการไหลของกระแสมอเตอร์

2.4 ส่วนควบคุมและประมวลผล

เรานำคอมพิวเตอร์มารับค่าจากส่วนอินเตอร์เฟสแล้วนำค่าที่ได้มาคำนวณ กำหนดค่าพารามิเตอร์ต่างๆ แล้วจึงส่งค่าที่ได้ไปยังส่วนอินเตอร์เฟสอีกทีหนึ่งเพื่อควบคุมระบบต่อไป ส่วนควบคุมเราจะแสดงในบทที่ 3 และ โปรแกรมในภาคผนวก



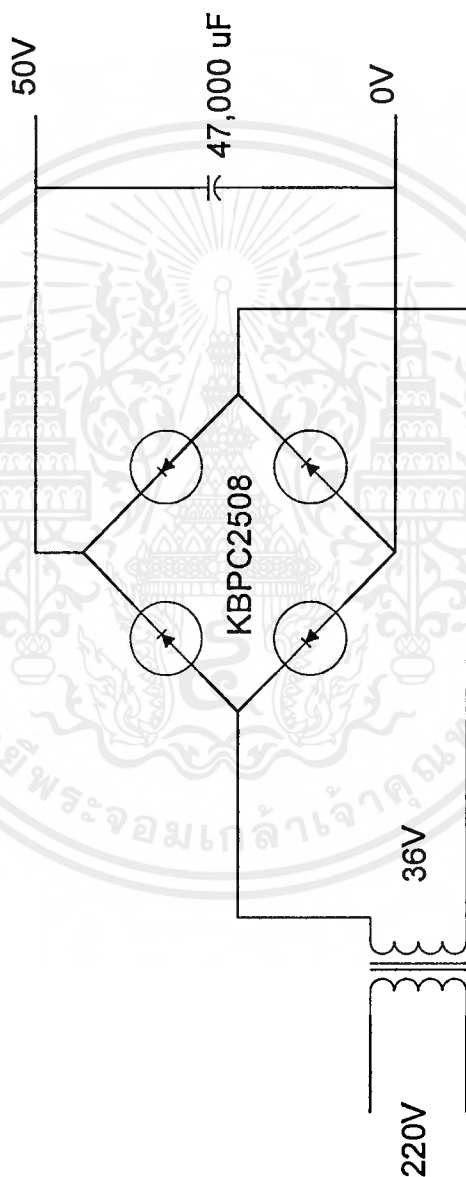
Part1



Part2

รูปที่ 2.8 วงจรควบคุมแรงดันมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

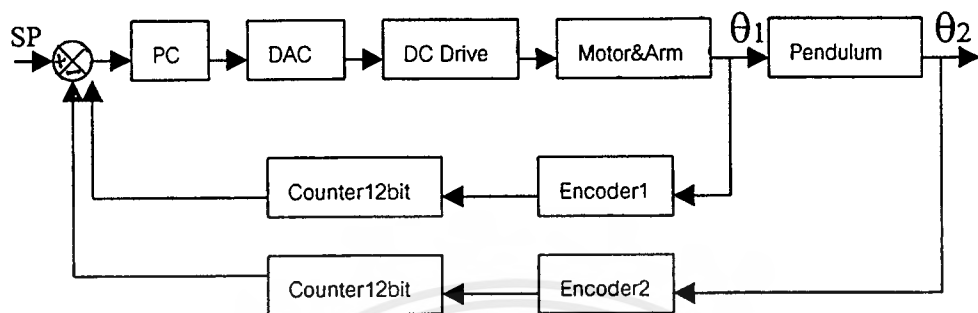


รูปที่ 2.9 แหล่งจ่ายไฟตรง 50 โวลท์

บทที่ 3

รูปแบบทางคณิตศาสตร์และการออกแบบระบบควบคุม

เราสามารถเขียนเป็นบล็อกไดอะแกรม(Block Diagram) ได้ดังนี้



รูปที่ 3.1 บล็อกไดอะแกรมของระบบทั้งหมด

3.1 สมการและโมเดลของระบบ

จากรูปที่ 3.1 เราสามารถหาฟังก์ชันถ่ายโอนของระบบ DC Motor ได้จากหนังสือ “ Modern Control Engineering ” ของ Ogata หน้าที่ 118 ถึงหน้าที่ 121 จะได้แบบจำลองทางคณิตศาสตร์ดังนี้

$$\frac{\Theta_1(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \tag{3.1}$$

หรือจัดรูปสมการใหม่ได้

$$\frac{\dot{\Theta}_1(s)}{E_a(s)} = \frac{K_m}{(T_m s + 1)} \tag{3.2}$$

ค่าตัวแปรต่าง ๆ นิยามได้ดังนี้

$$K_m = \frac{K}{R_a b_1 + K K_b} \quad \text{: ค่าคงที่ของมอเตอร์ (motor gain constant)}$$

$$T_m = \frac{R_a J_1}{R_a b_1 + K K_b} \quad \text{: ค่าคงที่เวลาของมอเตอร์ (motor time constant)}$$

K : ค่าคงที่แรงบิดของมอเตอร์ (motor torque constant)

K_b : ค่าคงที่แรงดันย้อนกลับ (back emf constant)

R_a : ความต้านทานอาร์เมเจอร์ (armature resistance)

b_1 : แรงเสียดทาน (equivalent visous-friction coefficient of the motor and load referred to the motor shaft)

J_1 : โมเมนต์ความเฉื่อย (equivalent moment of inertia of the motor and load referred to the motor shaft)

M : มวลของแขน

l_1 : ความยาวของแขน

J_a : โมเมนต์ความเฉื่อยของแขนรอบจุดหมุน

J_m : โมเมนต์ความเฉื่อยของมอเตอร์

θ_1 : มุมของแขนที่เปลี่ยนไป (rad)

จากการคำนวณและข้อมูลของมอเตอร์ (ภาคผนวก)

$$K = 0.181 \text{ N-m/A}$$

$$K_b = 19 \text{ V/KRPM} = 0.1814 \text{ V-sec/rad}$$

$$R_a = 1.2 \Omega, \quad M = 0.293 \text{ kg}$$

$$b_1 = 0.00351 \text{ N-m/rad/sec}, \quad l_1 = 0.74 \text{ m}$$

$$J_m = 0.00028 \text{ kg-m}^2$$

$$J_a = (1/12)Ml_1^2 = 0.01337 \text{ kg-m}^2$$

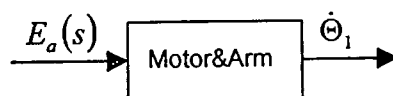
$$J_1 = J_m + J_a = 0.01365 \text{ kg-m}^2$$

$$K_m = \frac{0.181}{(1.2)(0.00351) + (0.181)(0.1814)} = 4.886$$

$$T_m = \frac{(1.2)(0.01365)}{(1.2)(0.00351) + (0.181)(0.1814)} = 0.442$$

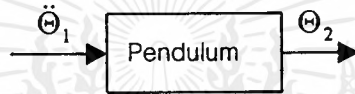
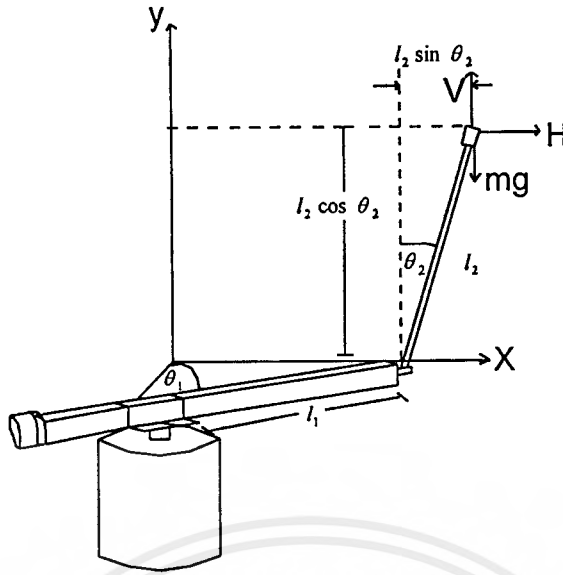
เมื่อแทนค่าต่างๆจะได้

$$\frac{\dot{\theta}_1(s)}{E_a(s)} = \frac{4.886}{0.442s + 1} = \frac{11.0543}{s + 2.262} \quad \text{rad/volt} \quad (3.3)$$



รูปที่ 3.2 แสดงแบบจำลองของมอเตอร์และแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงแบบจำลองของลูกตุ้ม

l_1 : ความยาวของแขนนับจากจุดหมุน(m)

l_2 : ความยาวของก้านลูกตุ้ม(m)

m : น้ำหนักของก้านลูกตุ้ม(kg)

θ_1 : มุมของแขนที่เปลี่ยนไป(rad)

θ_2 : มุมของก้านลูกตุ้มที่เปลี่ยนไป(rad)

V : แรงในแนวตั้งที่ทำกับก้านลูกตุ้ม(N)

H : แรงในแนวนอนที่ทำกับก้านลูกตุ้ม(N)

จากรูปที่ 3.3 เราสามารถหาฟังก์ชันถ่ายโอนได้ดังนี้

จากกฎข้อ 2 ของนิวตัน จะได้

$$\text{พิจารณาในแกน Y; } V - mg = m \frac{d^2}{dt^2} (l_2 \cos \theta_2) \quad (3.4)$$

$$\text{พิจารณาในแกน X; } H = m \frac{d^2}{dt^2} (l_2 \sin \theta_2 + \theta_1 l_1) \quad (3.5)$$

เมื่อ $\theta_1 l_1$ คือระยะทางที่แขนเคลื่อนที่ไปเป็นวงกลมด้วยมุม θ_1

จากหนังสือ "Modern Control Engineering" หน้า 103 เราจะได้ความสัมพันธ์ คือ

$$J_2 \ddot{\theta}_2 + b_2 \dot{\theta}_2 = H l_2 \cos \theta_2 - V l_2 \sin \theta_2 \quad (3.6)$$

นำสมการที่ (3.4),(3.5) แทนลงใน(3.6)

$$J_2 \ddot{\theta}_2 + b_2 \dot{\theta}_2 = \left[m \frac{d^2}{dt^2} (l_2 \sin \theta_2 + l_1 \theta_1) \right] l_2 \cos \theta_2 - \left[mg + m \frac{d^2}{dt^2} (l_2 \cos \theta_2) \right] l_2 \sin \theta_2$$

และ

$$\frac{d^2}{dt^2} \sin \theta = -(\sin \theta) \dot{\theta}^2 + (\cos \theta) \ddot{\theta}$$

$$\frac{d^2}{dt^2} \cos \theta = -(\cos \theta) \dot{\theta}^2 - (\sin \theta) \ddot{\theta}$$

$$\begin{aligned} J_2 \frac{d^2}{dt^2} \theta_2 + b_2 \frac{d}{dt} \theta_2 &= ml_2 \cos \theta_2 \left[l_2 (\cos \theta_2 (\ddot{\theta}_2) - \sin \theta_2 (\dot{\theta}_2)^2) + l_1 \ddot{\theta}_1 \right] - mgl_2 \sin \theta_2 \\ &\quad - ml_2^2 \sin \theta_2 \left[-\sin \theta_2 (\ddot{\theta}_2) - \cos \theta_2 (\dot{\theta}_2)^2 \right] \\ &= ml_2^2 (\cos \theta_2)^2 - ml_2^2 \cos \theta_2 \sin \theta_2 (\dot{\theta}_2)^2 + ml_1 l_2 \cos \theta_2 (\ddot{\theta}_1) \\ &\quad - mgl_2 \sin \theta_2 + ml_2^2 (\sin \theta_2)^2 \ddot{\theta}_2 + ml_2^2 \cos \theta_2 \sin \theta_2 (\dot{\theta}_2)^2 \\ &= ml_2^2 \ddot{\theta}_2 (\cos^2 \theta_2 + \sin^2 \theta_2) - mgl_2 \sin \theta_2 + ml_1 l_2 \cos \theta_2 (\ddot{\theta}_1) \end{aligned}$$

จะได้

$$(J_2 - ml_2^2) \ddot{\theta}_2 + b_2 \dot{\theta}_2 + mgl_2 \sin \theta_2 = \ddot{\theta}_1 ml_1 l_2 \cos \theta_2$$

เราประมาณค่า $\sin \theta_2 \sim 0$ และ $\cos \theta_2 \sim 1$ เมื่อ $\theta_2 \sim 0$

จะได้

$$(J_2 - ml_2^2) \ddot{\theta}_2 + b_2 \dot{\theta}_2 = \ddot{\theta}_1 ml_1 l_2$$

Laplace transform จะได้

$$\frac{\Theta_2(s)}{\Theta_1(s)} = \frac{ml_1 l_2}{(J_2 - ml_2^2) s^2 + b_2 s} \quad (3.7)$$

เมื่อแทนค่าพารามิเตอร์ต่างๆ ดังนี้

$$m = 0.05 \text{ kg} \quad , l_1 = 0.47 \text{ m} \quad , l_2 = 0.48 \text{ m}$$

$$J_2 = 0.02536 \text{ kg-m}^2$$

$$b_2 = 0.002768 \text{ N-m/rad/sec}$$

จะได้

$$\begin{aligned} \frac{\Theta_2(s)}{\Theta_1(s)} &= \frac{0.01128}{0.01384s^2 + 0.002768s} \\ &= \frac{0.815}{s(s+0.2)} \end{aligned} \quad (3.8)$$

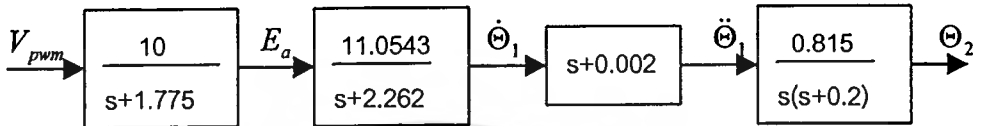
จากสมการที่ 4.12 ฟังก์ชันถ่ายโอนของวงจรถวลุมแรงดันคือ

$$\frac{E_a(s)}{V_{pwm}(s)} = \frac{10}{s+1.775}$$

และจากสมการที่ 4.13

$$\frac{\ddot{\Theta}_1(s)}{V_{pwm}(s)} = \frac{110.543(s+0.002)}{(s+1.775)(s+2.262)}$$

จากสมการที่ (3.3),(3.8),(4.12),(4.13) เราจะได้ความสัมพันธ์ของระบบดังรูปที่ 3.4



รูปที่ 3.4 แสดงแบบจำลองโมเดลของระบบ

3.2 การออกแบบระบบควบคุม

ในการออกแบบระบบควบคุม สามารถทำการออกแบบได้ 2 วิธีคือ ระบบควบคุมด้วยLQR และระบบควบคุมด้วย State Feedback Control System

3.2.1 ระบบควบคุมแบบป้อนกลับด้วยการป้อนกลับแบบสเกต

จากสมการฟังก์ชันถ่ายโอนที่(4.11)

$$\frac{\ddot{\Theta}_1(s)}{V_{pwm}(s)} = \frac{110.543(s+0.002)}{(s+1.775)(s+2.262)} = \frac{110.543}{s^2 + 4.037s + 4.015}$$

จะได้

$$s^3\Theta_1(s) + 4.037s^2\Theta_1(s) + 4.015\Theta_1(s) = 110.543V_{pwm}(s)$$

อินเวอร์สลาปลาซจะได้

$$\ddot{\Theta}_1(t) = -4.037\dot{\Theta}_1(t) - 4.015\Theta_1(t) + 110.543V_{pwm}(t) \quad (3.9)$$

จากสมการฟังก์ชันถ่ายโอนที่ (3.8)และ(4.13)

$$\frac{\Theta_2(s)}{\dot{\Theta}_1(s)} = \frac{0.815(s+0.002)}{s^2 + 0.2s}$$

จัดรูปสมการได้

$$s^2\Theta_2(s) + 0.2\Theta_2(s) = 0.815s\dot{\Theta}_1(s) + 0.00163\dot{\Theta}_1(s)$$

อินเวอร์สลาปลาซจะได้

$$\ddot{\Theta}_2(t) = 0.815\ddot{\Theta}_1(t) + 0.00163\dot{\Theta}_1(t) - 0.2\dot{\Theta}_2(t) \quad (3.10)$$

เรานิยามตัวแปรสแตทตัวใหม่ x ดังต่อไปนี้

$$\begin{aligned}x_1 &= \theta_1 \\x_2 &= \dot{\theta}_1 \\x_3 &= \ddot{\theta}_1 \\x_4 &= \theta_2 \\x_5 &= \dot{\theta}_2\end{aligned}$$

จากสมการ (3.9), (3.10) จะได้

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -4.015x_2 - 4.037x_3 + 110.543v_{pwm} \\ \dot{x}_4 &= x_5 \\ \dot{x}_5 &= 0.00163x_2 + 0.815x_3 - 0.2x_5\end{aligned}$$

สามารถจัดสมการสแตทเป็น 5 มิติ ได้ดังนี้

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

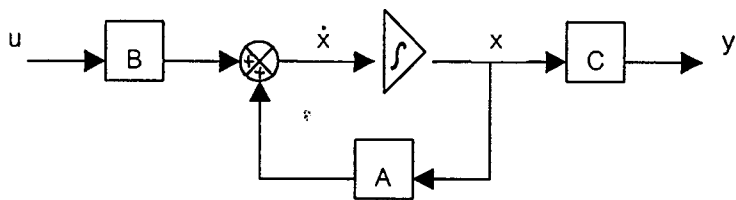
โดยที่

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

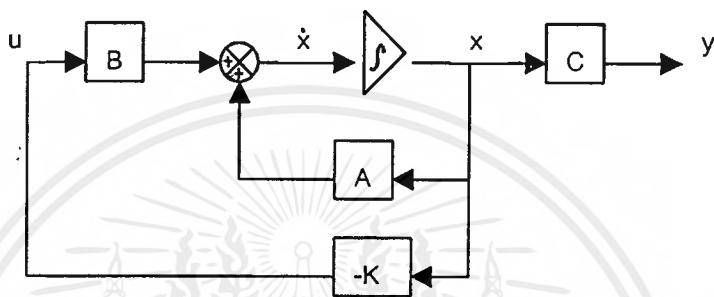
$u = v_{pwm}$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -4.015 & -4.037 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0.00163 & 0.815 & 0 & -0.2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 110.543 \\ 0 \\ 0 \end{bmatrix} v_{pwm}$$

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



รูปที่3.5 ระบบก่อนการควบคุม



รูปที่3.6 ระบบหลังการควบคุม

เราสามารถออกแบบค่าเกน(K feedback) ป้อนกลับได้จากการใช้โปรแกรมMATLABได้

$$K = [k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5]$$

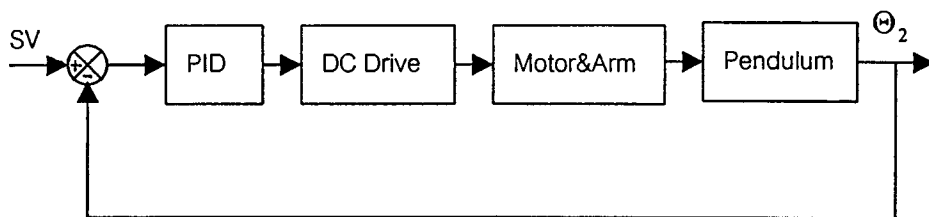
$$= [8.87 \quad 13.04 \quad 0.464 \quad -5.93 \quad -9.48]$$

และระบบหลังการควบคุมคือ

$$\dot{x} = (A - BK)x$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -980.516 & -1445.5 & -55.33 & 655.52 & 1047.94 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0.00163 & 0.815 & 0 & -0.2 \end{bmatrix}$$

3.2.2 ระบบควบคุมที่ใช้ตัวควบคุมแบบ PID



รูปที่ 3.7 บล็อกไดอะแกรมของระบบหลังถูกควบคุม

ซึ่งเราสามารถออกแบบการควบคุมแบบ PID ได้จากการใช้โปรแกรม MATLAB ได้ Controller ของระบบดังนี้

$$\frac{V_{pwm}(s)}{E_{ror}(s)} = \frac{0.07812(s+0.15)}{(s+2.86)(s+0.002)} \quad (3.11)$$

เราต้องการทำการแปลงเป็นคิสมิทก่อนนำไปเขียน โปรแกรมได้ดังนี้ จากสมการ(3.8)จะได้

$$\frac{V_{pwm}(s)}{E_a(s)} = \frac{0.0741}{s+2.86} + \frac{0.00405}{s+0.002}$$

จาก Z transform ของ $\frac{1}{s+a} = \frac{1}{1-e^{-aT}Z^{-1}}$

เมื่อ T คือ Sampling time ซึ่งเท่ากับ 0.005วินาที

$$\begin{aligned} \text{จะได้ว่า } \frac{V_{pwm}(Z)}{E_{ror}(Z)} &= \frac{0.0741}{1-e^{-2.86(0.005)}Z^{-1}} + \frac{0.00405}{1-e^{-0.002(0.005)}Z^{-1}} \\ &= \frac{0.07815-0.07815Z^{-1}}{1-2Z^{-1}+Z^{-2}} \end{aligned}$$

$$V_{pwm}(Z) = 2Z^{-1}V_{pwm}(Z)-Z^{-2}V_{pwm}(Z)+0.07815E_{ror}(Z)-0.07815Z^{-1}E_{ror}(Z)$$

$$V_{pwm}(n) = 2V_{pwm}(n-1)-V_{pwm}(n-2)+0.07815E_{ror}(n)-0.07815E_{ror}(n-1)$$

****ข้อสังเกต** จะเห็นว่าการควบคุมแบบPID นั้น สามารถควบคุมได้เพียงแคให้ก้านลูกตุ้มสามารถทรงตัวอยู่ได้บนแกนโดยไม่หกล่น($\Theta_2 \sim 0$) เท่านั้น ไม่สามารถควบคุมตำแหน่งของแกนให้อยู่นิ่ง

บทที่ 4

ผลการทดลอง

จากระบบที่เราได้ออกแบบและได้ประกอบขึ้นตามแบบที่ผ่านมา จะเห็นว่าเรามีความจำเป็นที่จะต้องหาพารามิเตอร์คุณลักษณะของระบบก่อน แล้วเราจึงจะนำค่าเหล่านั้นมาแทนในโมเดลของระบบเพื่อใช้ในการควบคุมต่อไป

4.1 การหาพารามิเตอร์ของระบบ

4.1.1 โมเมนต์ความเฉื่อยของแขน(J_1)และค่าสัมประสิทธิ์ความเสียดทานของข้อต่อ(b_1)

ความยาวแขนนับจากจุดหมุนถึงข้อต่อลูกตุ้ม(l_1) = 0.47 m

น้ำหนักแขน(M) = 0.293 kg

J_1 = ค่าโมเมนต์ความเฉื่อยของส่วนแขน+ ค่าโมเมนต์ความเฉื่อยของมอเตอร์

ค่าโมเมนต์ความเฉื่อยของส่วนแขน(J_a) = $(1/2)Ml_1^2$

ค่าโมเมนต์ความเฉื่อยของมอเตอร์(J_m) คำนวณได้ดังนี้

จากสมการของมอเตอร์กระแสตรง(DC motor)

$$J\ddot{\theta} + b\dot{\theta} = \tau \quad (4.1)$$

ให้สภาวะเริ่มต้น $\theta(0) = 0$ และ $\dot{\theta}(0) = 0$ แล้วทำการแปลงลาปลาซ

$$Js^2\Theta(s) + bs\Theta(s) = \tau(s) \quad (4.2)$$

จาก(4.2)เขียนได้เป็น

$$\dot{\Theta}(s) = \frac{\tau(s)}{Js + b} \quad (4.3)$$

เมื่อให้ $\tau(s) = T * U(s)$

$$\begin{aligned} \dot{\Theta}(s) &= \frac{T * U(s)}{Js + b} \\ &= \frac{T/b}{(J/b)s + 1} * U(s) \end{aligned} \quad (4.4)$$

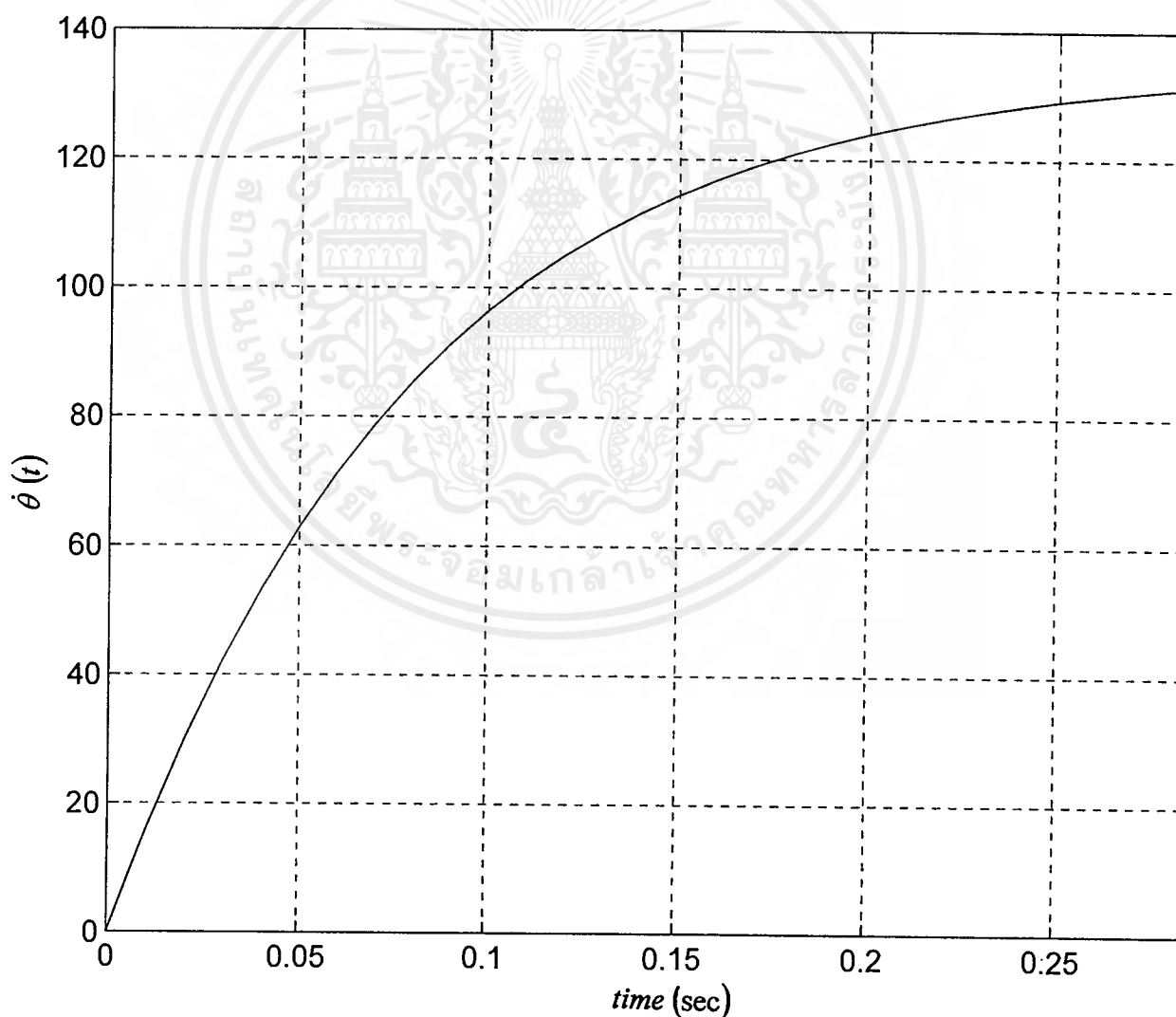
ดังนั้นเมื่อเราทำการป้อนแรงบิดคงที่ค่าหนึ่ง (Step Torque) ให้กับมอเตอร์ แล้วนำกราฟ ณ ช่วงเริ่มหมุนจากการทดลองมาเปรียบเทียบกับ (fit curve) กับกราฟที่ได้จากการเขียนแบบของสมการ (4.4) ที่ค่า J และ b ที่เหมาะสม (โดยการทดลองเปลี่ยนค่าทั้งสองแล้วนำมาเขียนแบบไปจนได้ค่าใกล้เคียง) ก็สามารถหาค่าของ J และ b ของมอเตอร์ออกมาได้ ดังรูปที่ 4.1 จะได้ว่า J และ b มีค่าเป็น

$$\frac{\dot{\theta}(s)}{U(s)} = \frac{T/b}{(J/b)s + 1} = \frac{135.14}{0.08s + 1}$$

โดยที่ $T = P/w = 0.47376$

จะได้ $J_m = 0.00028 \text{ kg-m}^2$

$b_1 = 0.00351 \text{ kg-m}^2/\text{s}$



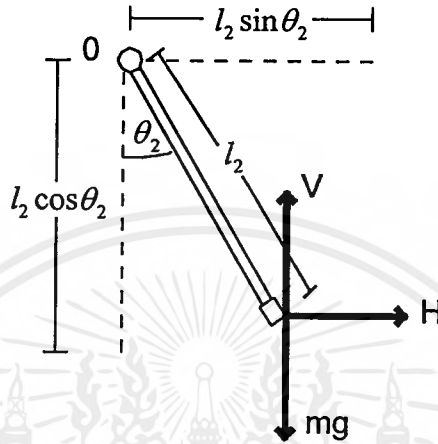
รูปที่ 4.1 ผลที่ได้จากการเขียนแบบโดยโปรแกรม MATLAB เพื่อหาค่า J_m และ b_1

4.1.2 โมเมนต์ความเฉื่อยของก้านลูกตุ้ม(J_2)และค่าสัมประสิทธิ์ความเสียดทานของข้อต่อ(b_2)

$$\text{ความยาวก้านลูกตุ้ม}(l_2) = 0.48 \text{ m}$$

$$\text{น้ำหนักของก้านลูกตุ้ม}(m) = 0.05 \text{ kg}$$

เมื่อเราขีดแกนไว้แล้วปล่อยให้ก้านลูกตุ้มแกว่งอย่างอิสระ จะได้ความสัมพันธ์ดังรูปที่ 4.2



รูปที่ 4.2 แสดงแบบจำลองการทดลองหาค่า J_2 และ b_2

จากกฎข้อ 2 ของนิวตัน

พิจารณาในแนวแกน Y จะได้

$$V - mg = m \frac{d^2}{dt^2} (l_2 \cos \theta_2) \quad (4.5)$$

พิจารณาในแนวแกน X จะได้

$$H = m \frac{d^2}{dt^2} (l_2 \sin \theta_2) \quad (4.6)$$

จากหนังสือ “Modern Control Engineering “ ของ Ogata หน้า 103 เราจะได้ความสัมพันธ์คือ

$$J_2 \ddot{\theta}_2 + b_2 \dot{\theta}_2 = H l_2 \cos \theta_2 - V l_2 \sin \theta_2 \quad (4.7)$$

นำสมการที่ (4.5), (4.6) แทนลงใน (4.7) จะได้

$$J_2 \ddot{\theta}_2 + b_2 \dot{\theta}_2 = \left[m \frac{d^2}{dt^2} (l_2 \sin \theta_2) \right] l_2 \cos \theta_2 - \left[mg + m \frac{d^2}{dt^2} (l_2 \cos \theta_2) \right] l_2 \sin \theta_2 \quad (4.8)$$

และ

$$\frac{d^2}{dt^2} \sin \theta = -(\sin \theta) \dot{\theta}^2 + (\cos \theta) \ddot{\theta}$$

$$\frac{d^2}{dt^2} \cos \theta = -(\cos \theta) \dot{\theta}^2 - (\sin \theta) \ddot{\theta}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้

$$\begin{aligned}
 J_2 \frac{d^2}{dt^2} \theta_2 + b_2 \frac{d}{dt} \theta_2 &= ml_2 \cos \theta_2 \left[l_2 (\cos \theta_2 (\ddot{\theta}_2) - \sin \theta_2 (\dot{\theta}_2)^2) \right] - mgl_2 \sin \theta_2 \\
 &\quad - ml_2^2 \sin \theta_2 \left[-\sin \theta_2 (\ddot{\theta}_2) - \cos \theta_2 (\dot{\theta}_2)^2 \right] \\
 J_2 \ddot{\theta}_2 + b_2 \dot{\theta}_2 &= ml_2^2 \ddot{\theta}_2 - mgl_2 \sin \theta_2
 \end{aligned} \tag{4.9}$$

เราประมาณค่า $\sin \theta_2 \sim \theta_2$ จากการกระจายอนุกรมเทเลอร์จะได้

$$\ddot{\theta}_2 = \frac{-mgl_2 \theta_2}{J_2 - ml_2^2} - \frac{b_2}{J_2 - ml_2^2} \dot{\theta}_2 \tag{4.10}$$

เปลี่ยนตัวแปรสแตทใหม่ x ได้ดังนี้

$$\begin{aligned}
 x_1 &= \theta_2 \\
 x_2 &= \dot{x}_1 = \dot{\theta}_2
 \end{aligned}$$

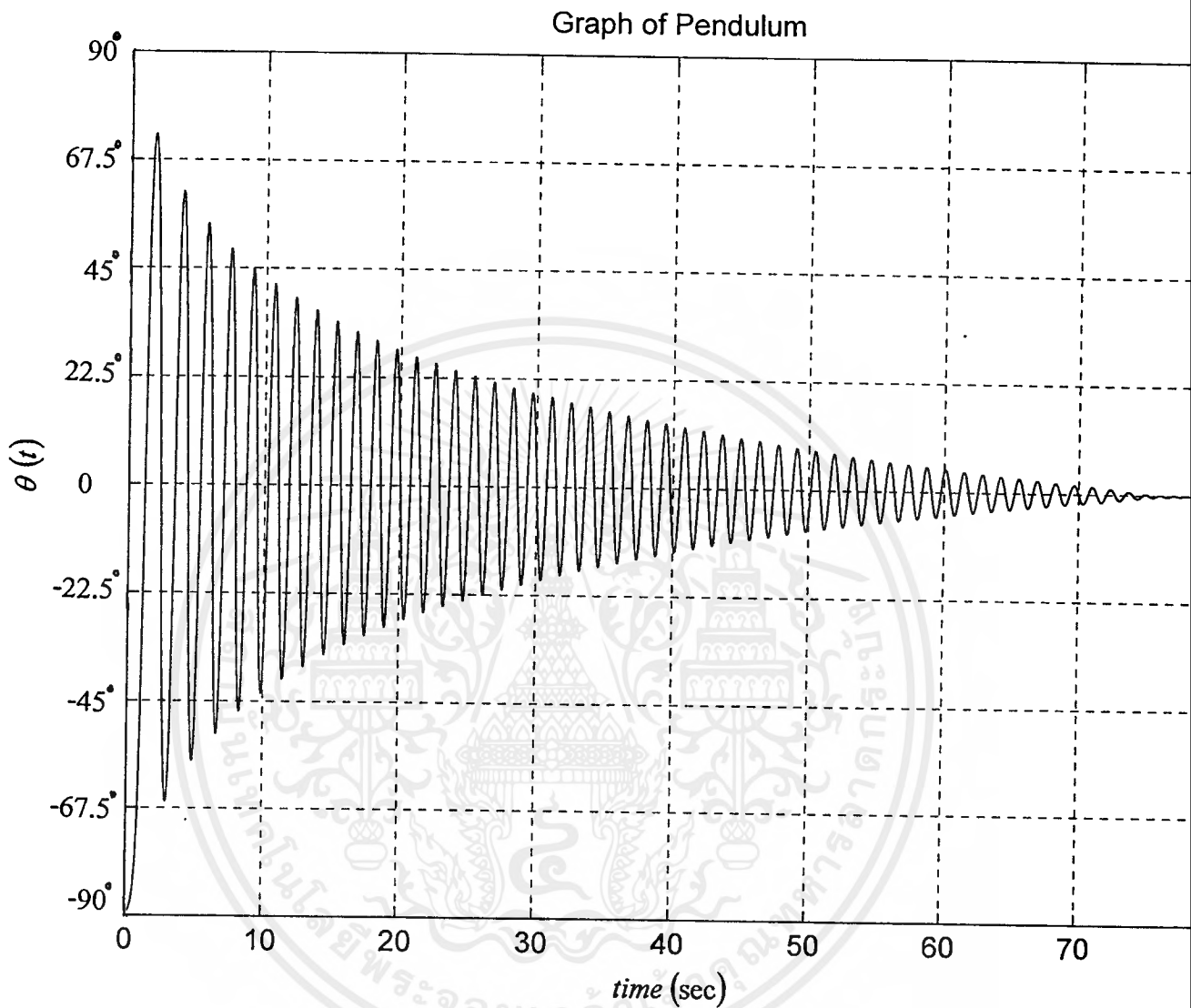
จะได้สมการ state space จากการคำนวณคือ

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-mgl_2}{J_2 - ml_2^2} & \frac{-b_2}{J_2 - ml_2^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

เราทดสอบระบบโดยปล่อยให้ก้านลูกตุ้มแกว่งโดยอิสระด้วยมุมเริ่มต้น 90 องศา หรือ 1.57 เรเดียน ขณะยึดแขนให้อยู่กับที่ในแนวระดับ นั่นคือ $x_1(0) = \theta_2(0) = 1.57$ เรเดียน และ $x_2(0) = \dot{\theta}_2(0) = 0$ แล้วใช้โปรแกรม MATLAB ในการเขียนแบบตามสมการสแตท ค่า b_2 และ J_2 ที่ค่าต่างๆกัน เพื่อนำมาเปรียบเทียบกับผลที่ได้จากการทดลองก็จะได้ค่า b_2 และ J_2 เท่ากับ

$$J_2 = 0.02536 \text{ kg-m}^2$$

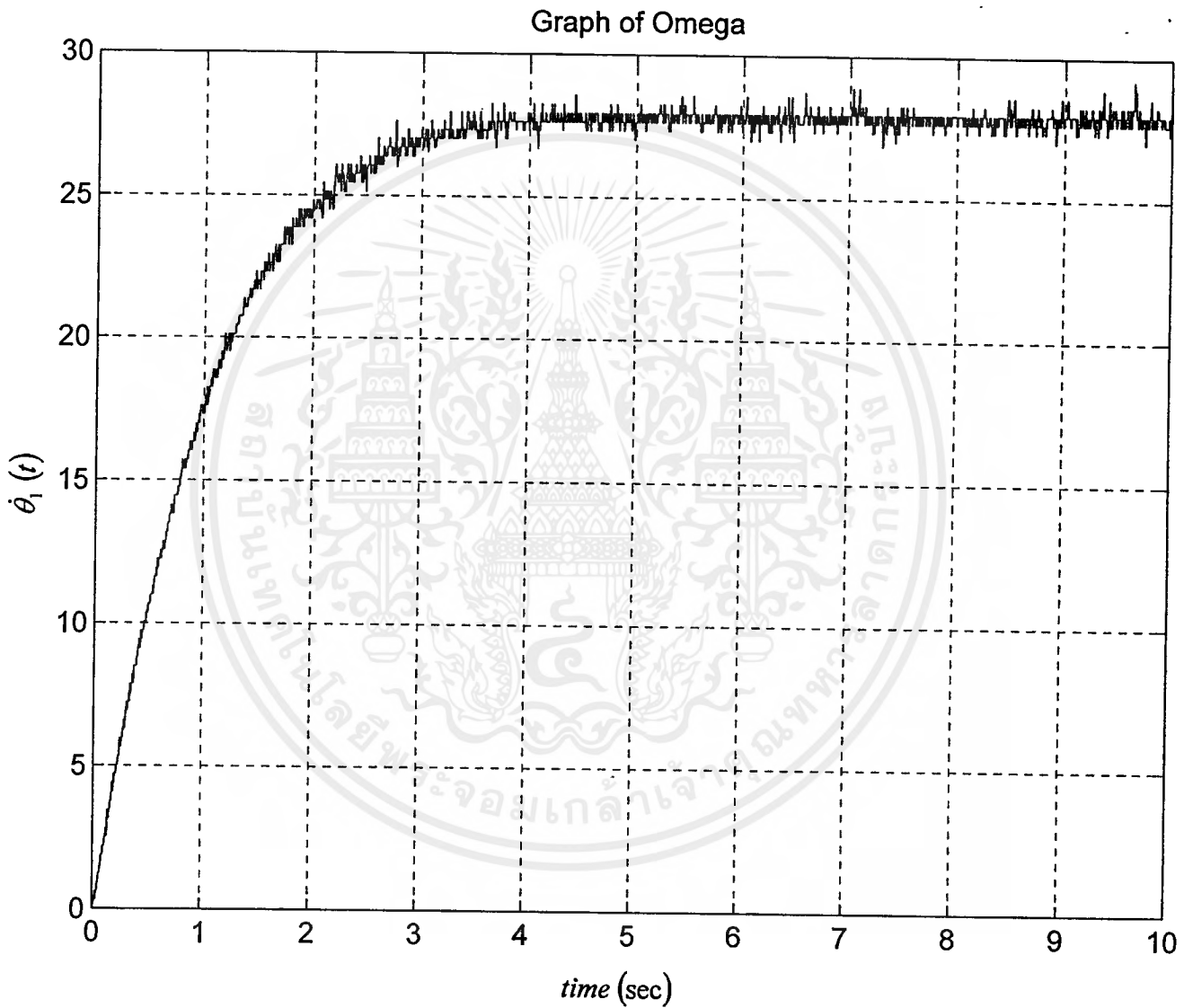
$$b_2 = 0.002768 \text{ kg-m}^2/\text{s}$$



รูปที่ 4.3 ผลที่ได้จากการทดลองเพื่อหาค่า J_2 และ b_2

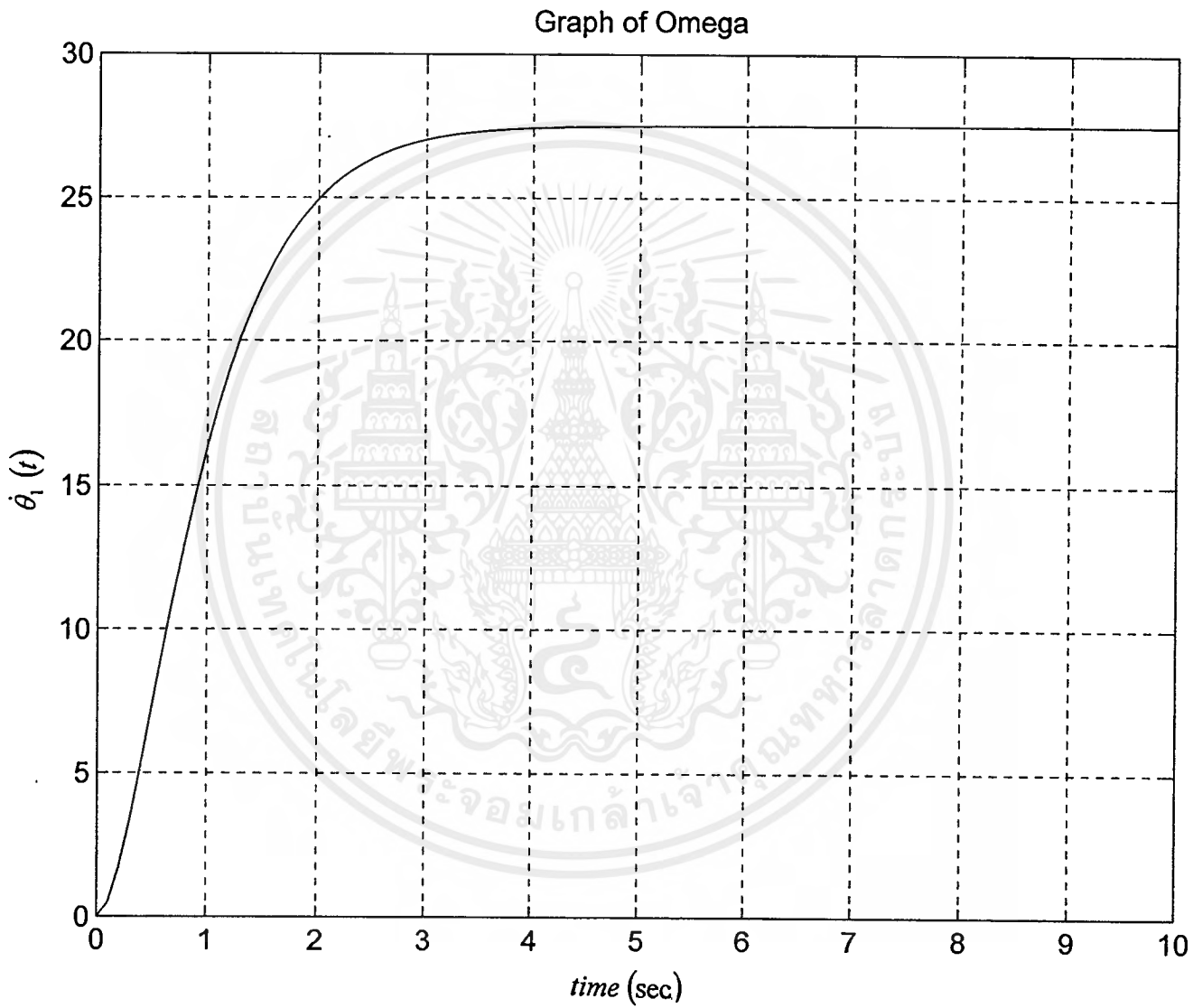
4.2 การทดลองหาฟังก์ชันถ่ายโอนของวงจรถวลความถี่

เนื่องด้วยวงจรถวลความถี่มีคุณลักษณะระหว่างความเร็วและ V_{pwm} ที่ป้อนให้ไม่เป็นเส้นตรง ซึ่งเราสามารถหาฟังก์ชันถ่ายโอนของวงจรถวลความถี่ได้ โดยการต่อวงจรผ่านโหลด คือ มอเตอร์ที่ประกอบแขนแล้ว และทำการป้อน V_{pwm} ด้วย Unit Step นำกราฟที่ได้จากการทดลองโดยใช้โปรแกรมคำนวณหาค่า $\theta_1(t)$ ดังที่ได้กล่าวผ่านมาแล้วในบทที่ 2 มาเปรียบเทียบกับ เอกสารที่ได้จากการใช้โปรแกรม MATLAB ดังรูปที่ 4.4 และ 4.5 ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 กราฟของ $\theta_1(t)$ ที่ได้จากการคำนวณด้วยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟของ $\theta_1(t)$ ที่ได้จากการใช้โปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะได้ฟังก์ชันถ่ายโอนของวงจรถูกควบคุมร่วมกับมอเตอร์และแกน คือ

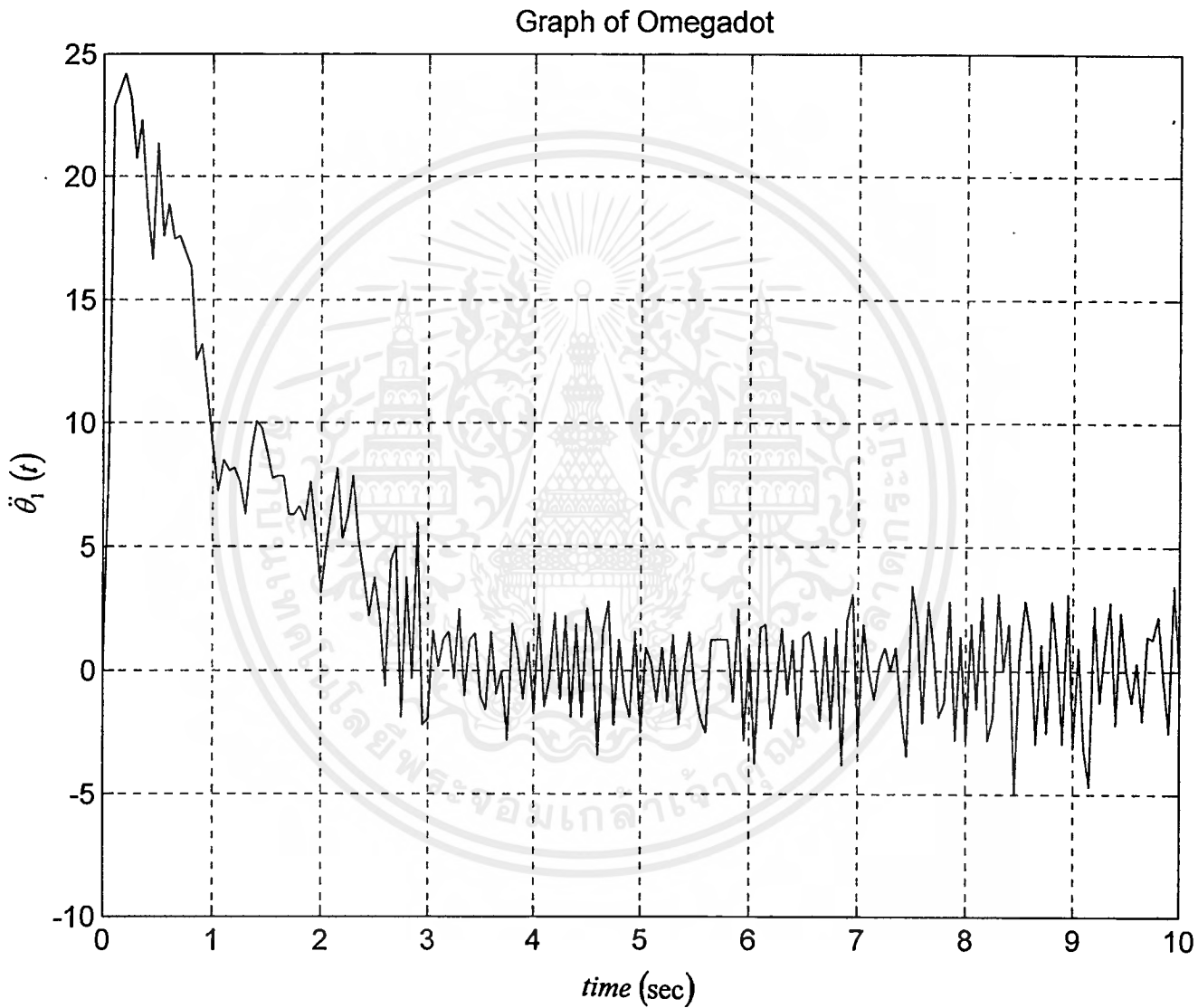
$$\frac{\theta_1(s)}{V_{pwm}(s)} = \frac{110.543}{(s+1.775)(s+2.262)} \quad (4.11)$$

ดังนั้น เราจะได้ฟังก์ชันถ่ายโอนของวงจรถูกควบคุมแรงดัน คือ

$$\frac{E_a(s)}{V_{pwm}(s)} = \frac{10}{(s+1.775)} \quad (4.12)$$

และในทำนองเดียวกันเราสามารถหาค่ากราฟผลตอบสนองของ $\theta_1(t)$ จากการป้อน V_{pwm} ด้วย Unit Step ได้กราฟจากการทดลองนำมาเปรียบเทียบกับกราฟจากการใช้โปรแกรม MATLAB ดังรูปที่ 4.6 และ 4.7

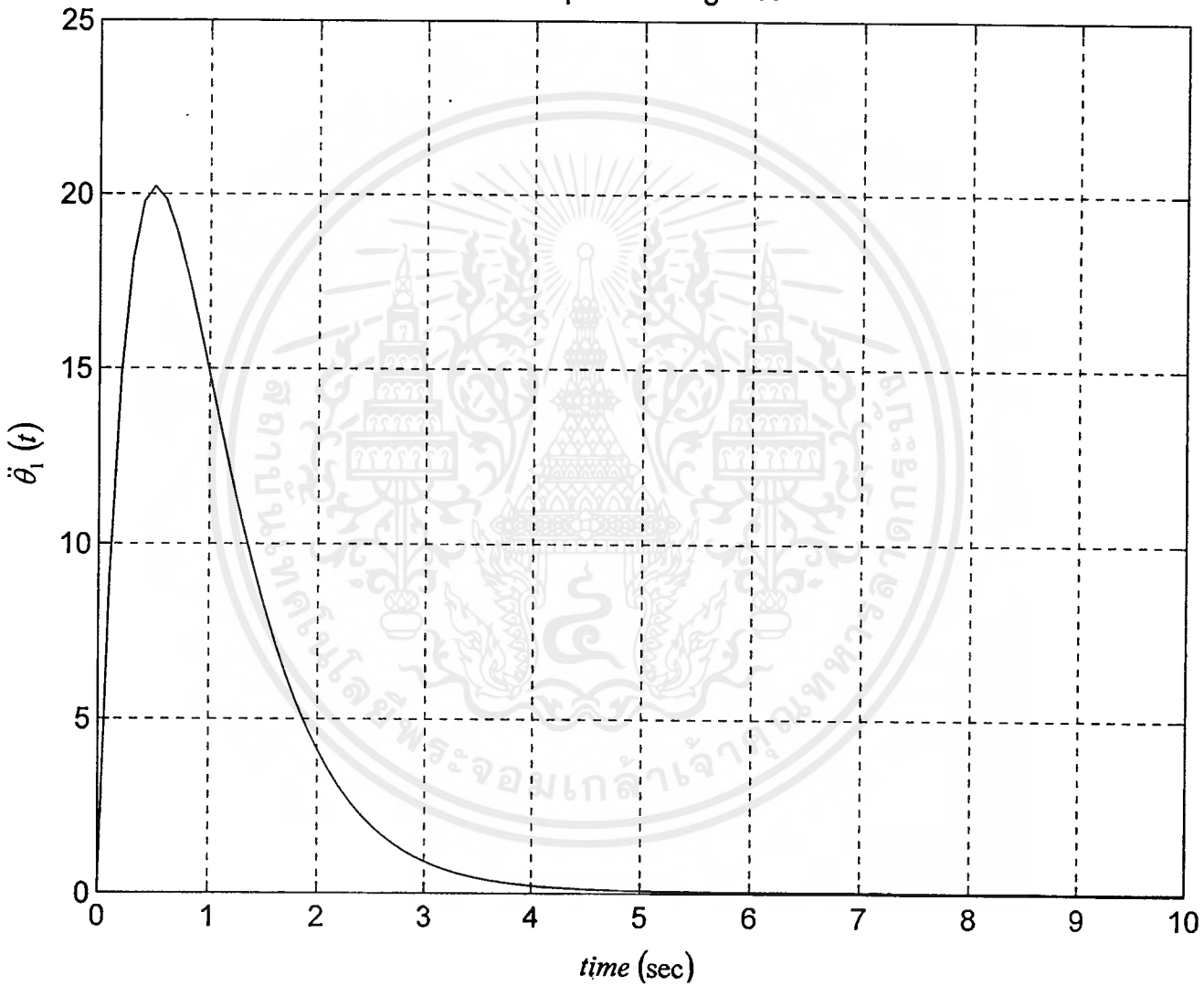




รูปที่ 4.6 กราฟของ $\dot{\theta}_1(t)$ ที่ได้จากการคำนวณด้วยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Graph of Omegadot

รูปที่ 4.7 กราฟของ $\dot{\theta}_1(t)$ ที่ได้จากการใช้โปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะได้ฟังก์ชันถ่ายโอนจากโปรแกรม MATLAB คือ

$$\frac{\theta_1(s)}{V_{pwm}(s)} = \frac{110.543(s+0.002)}{(s+1.775)(s+2.262)} \quad (4.13)$$

ดังนั้นแสดงว่าเราสามารถหาความสัมพันธ์ระหว่าง $\theta_1(s)$ และ $V_{pwm}(s)$ ในระบบของเราได้คือ

$$\frac{\theta_1(s)}{V_{pwm}(s)} = \frac{110.543(s+0.002)}{(s+1.775)(s+2.262)}$$

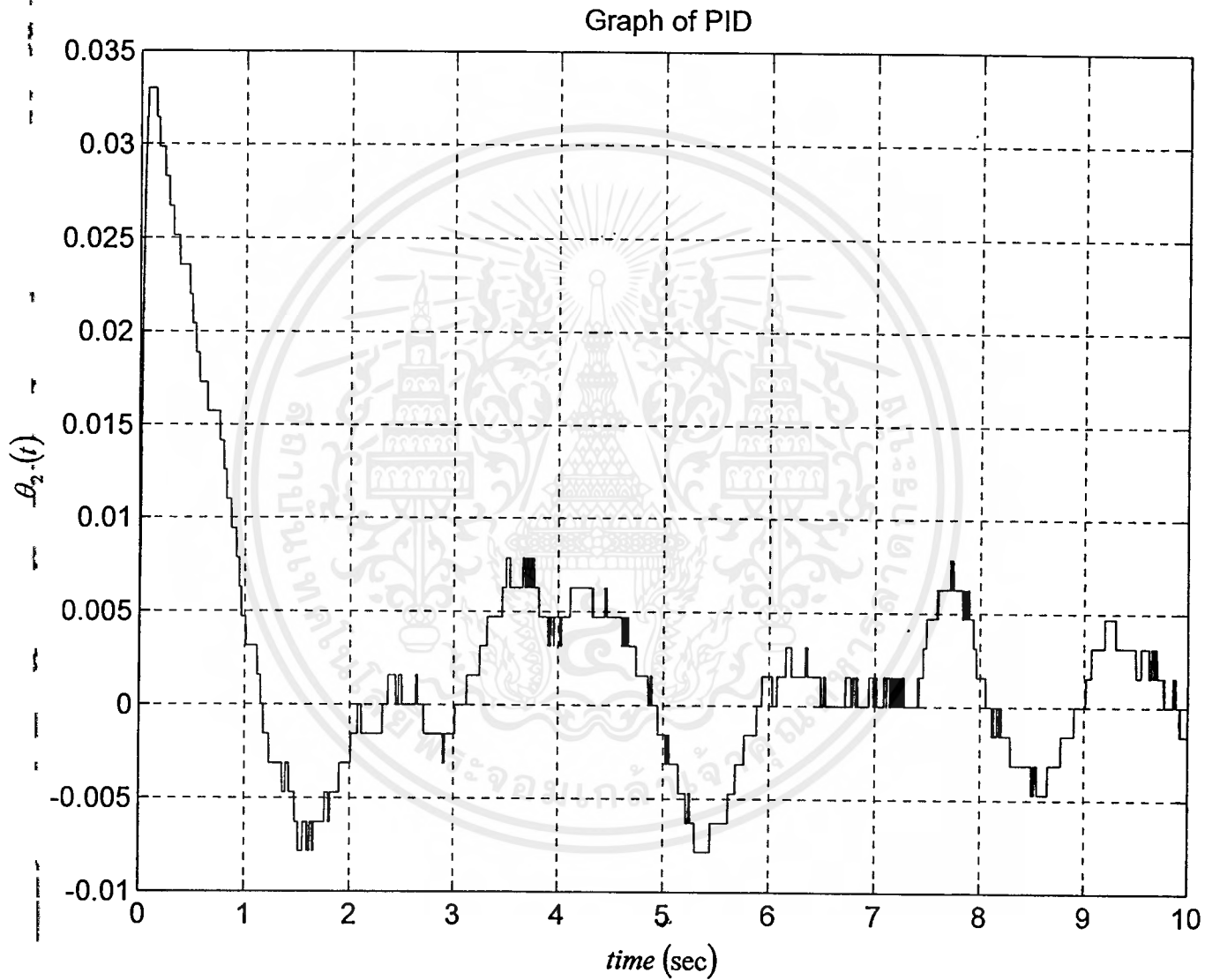
4.3 การทดลองระบบหลังการควบคุม

เนื่องจากเราต้องการทดลองซอฟต์แวร์ ก่อนว่ามีปัญหาหรือไม่สอดคล้องกับระบบโครงสร้างทางเครื่องกลของเราหรือไม่ ดังนั้นเราจึงทำการแบ่งขั้นตอนการพัฒนาการควบคุมดังนี้

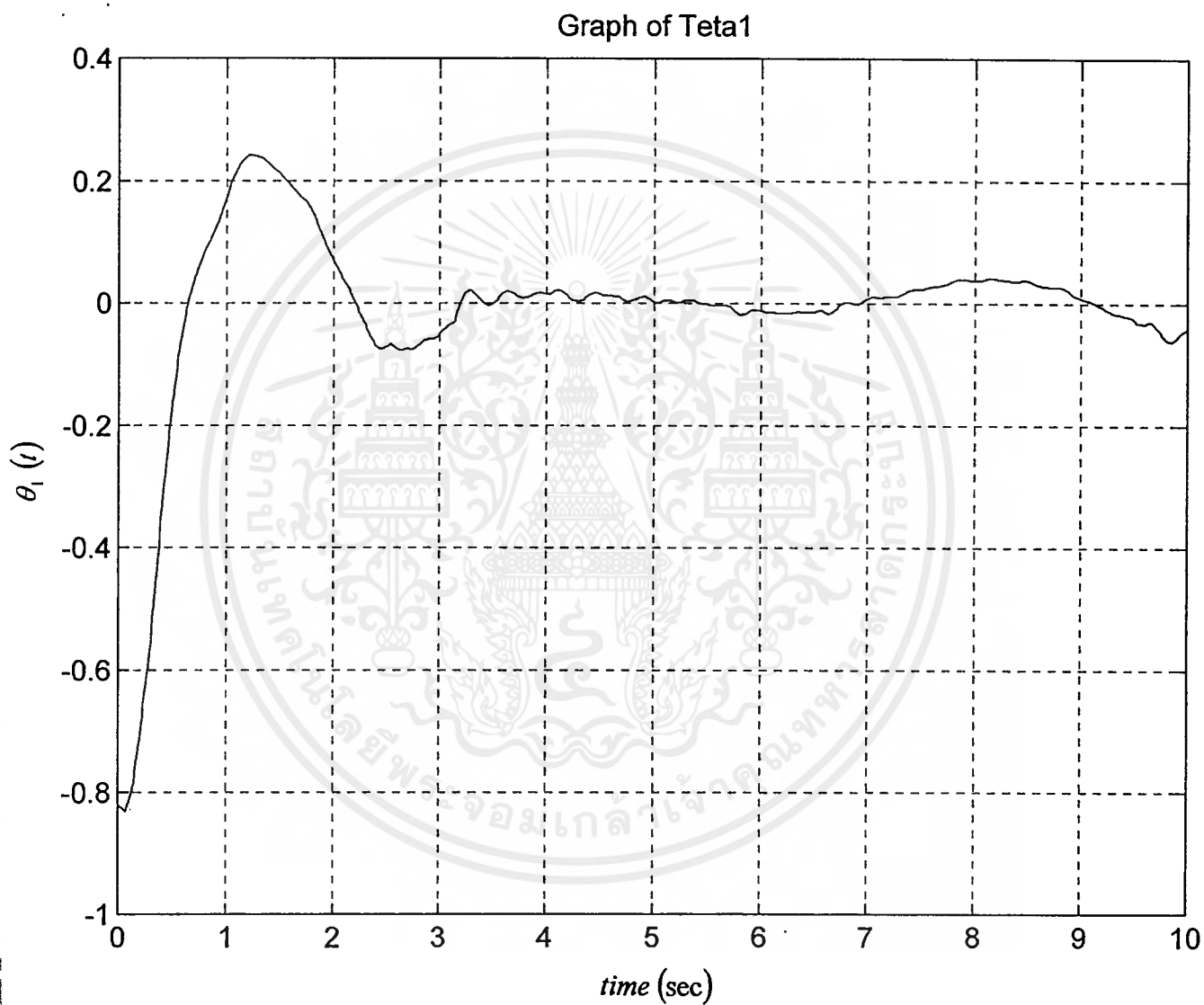
ขั้นแรก ทำการทดลองควบคุมขั้นพื้นฐานอย่างง่าย โดยการทดลองควบคุมแขนที่แยกก้านลูกค้อนออกมาให้เคลื่อนที่ไปหยุด ณ ตำแหน่งที่เรากำหนดใดๆ ได้ ด้วยการใช้อุปกรณ์ควบคุมแบบ PID ซึ่งให้ผลตอบสนองค่อนข้างถูกต้อง มีค่าความผิดพลาดน้อยมาก

จากนั้นจึงเริ่มขั้นตอนที่สอง ทำการทดลองควบคุมก้านลูกค้อนให้ทรงตัวบนแขนได้จากการออกแบบที่ผ่านมาแล้วในบทที่ 3 ด้วยการใช้อุปกรณ์ควบคุมแบบ PID ปรับแต่งค่า K_p , K_i , K_d ให้เหมาะสมและให้ผลตอบสนองที่ดีที่สุด แต่ในระดับนี้เรายังไม่สามารถกำหนดตำแหน่งแขนให้อยู่ ณ ตำแหน่งจุดสมดุลใดๆ ที่เราต้องการได้

เมื่อทดลองควบคุมก้านลูกค้อนแบบ PID สำเร็จแล้ว เราจึงทำการทดลองควบคุมก้านลูกค้อน ด้วยการใช้อุปกรณ์ควบคุมแบบป้อนกลับ จากสมการที่ออกแบบในบทที่ 3 แล้วได้ผลการทดลองดังรูป



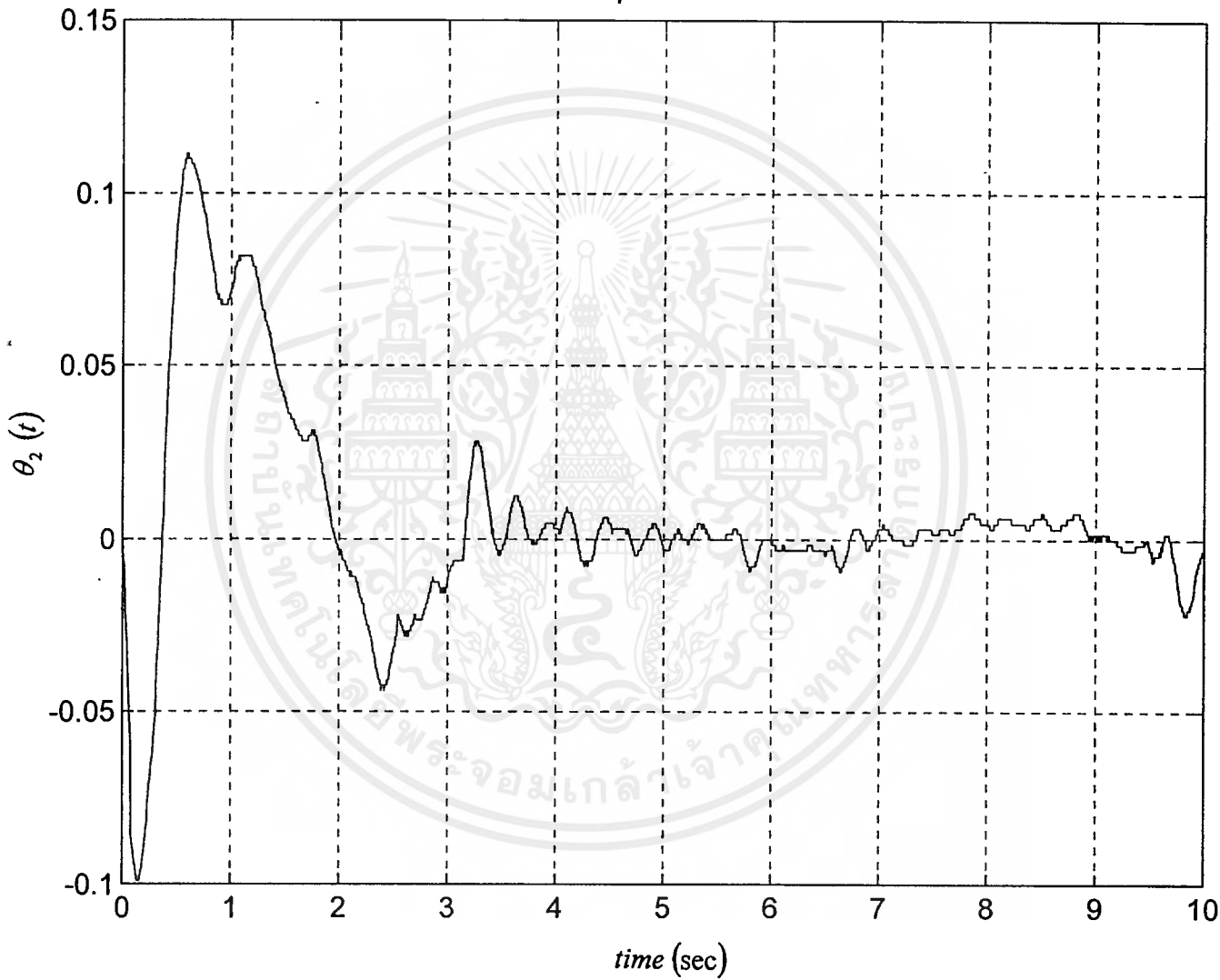
รูปที่ 4.8 กราฟแสดงผลตอบสนองของระบบหลังการควบคุมแบบ PID



รูปที่ 4.9 กราฟแสดงผลตอบสนองของระบบหลังการควบคุม
แบบป้อนกลับของ $\theta_1(t)$ หรือแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Graph of Teta2



รูปที่ 4.10 กราฟแสดงผลตอบสนองของระบบหลังการควบคุม
แบบป้อนกลับของ $\theta_2(t)$ หรือก้านลูกตุ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิเคราะห์ผลการดำเนินการ

5.1 สรุปการดำเนินงาน

จากการศึกษาและทดลองของโครงการ ในส่วน โครงสร้างทางเครื่องกล ได้ออกแบบไว้ อย่างเหมาะสม ซึ่งเป็นผลดีต่อระบบอย่างยิ่งเนื่องจากเกิดปัญหาน้อยมาก ด้านวงจรอิเล็กทรอนิกส์ และการติดต่อกับคอมพิวเตอร์ เราเลือกใช้อุปกรณ์และออกแบบวงจร เพื่อให้สามารถรับและส่งข้อมูลที่ใช้ควบคุมได้อย่างสะดวก โดยใช้ Microcontroller ในการแปลงข้อมูลถึง 4 ตัว ส่วนควบคุม คือ โปรแกรมเราใช้โปรแกรมภาษาซี ด้วยโครงสร้างโปรแกรมที่ตรงไปตรงมาทำงานง่าย ไม่ซับซ้อน ง่ายต่อการตรวจสอบ การศึกษาและการพัฒนาต่อไปในอนาคต

จากการได้ทดลองการควบคุมเสถียรภาพของ Inverted Pendulum ซึ่งเราได้มีใช้การควบคุม ระบบอยู่ด้วยกัน 2 แบบ คือ

- 1) การควบคุม $\theta_2(t)$ โดยใช้ PID Controller ซึ่งเราสามารถปรับแต่งค่า K_p , K_i และ K_d เพื่อใช้ค่า error ที่ $\theta_2(t)$
- 2) การควบคุมทั้ง $\theta_1(t)$ และ $\theta_2(t)$ โดยใช้ state feedback ซึ่งการใช้ state feedback นั้น เราจำเป็นต้องหาค่าเกน K feedback เพื่อมาควบคุมเสถียรภาพของ Inverted Pendulum โดยเราต้องพิจารณาค่าพารามิเตอร์ของระบบถึง 5 ค่า คือ θ_1 , $\dot{\theta}_1$, $\ddot{\theta}_1$, θ_2 , $\dot{\theta}_2$ ซึ่งเราสามารถหาค่าเกน $K = [k_1 \ k_2 \ k_3 \ k_4 \ k_5]$ ได้โดยการคำนวณ ดังที่ได้กล่าวมาแล้ว แต่ค่าที่ได้จากการคำนวณนั้น เมื่อนำมาทดลองพบว่าไม่สามารถที่จะทำให้ระบบเข้าสู่เสถียรภาพได้จึงจำเป็นต้องมีการปรับแต่งค่าเกน K โดยพิจารณาถึง ลักษณะของระบบโดยที่เราพิจารณาว่าค่า θ_2 มีความสำคัญมากต่อระบบ และได้ทำการปรับแต่งให้ค่าเกน K ของ θ_2 (หรือ k_4) มีค่าน้อยลง ซึ่งมีผลทำให้ Input ของระบบมีค่ามากขึ้น ซึ่งเมื่อสามารถทำให้ θ_2 มี error น้อยมากแล้วจึงทำการปรับแต่งพารามิเตอร์ตัวอื่น เพื่อให้ระบบสามารถเข้าสู่เสถียรภาพได้เร็วที่สุด

5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

ตัวมอเตอร์ ดีซีมอเตอร์ที่เราใช้ในตอนแรกเป็นแบบขับโดยตรงขนาด 24 โวลต์ เอน โคด เดอร์ขนาด 432 พัลส์ ไม่พอกับความต้องการสำหรับการควบคุมและเอน โคด เดอร์ มีความละเอียดค่อนข้างต่ำไม่พอกับความต้องการ จึงต้องเปลี่ยนมอเตอร์ใหม่เป็นมอเตอร์แบบขับ โดยตรงขนาด 100 โวลต์ เอน โคด เดอร์ขนาด 1000 พัลส์ ทำให้เป็นการเสียเวลาและวัสดุที่ต้องนำมาแก้ไข ออก

แบบใหม่ ในส่วนประกอบที่เกี่ยวข้องกับรูปทรงของมอเตอร์ที่เปลี่ยนไป ได้แก่ ส่วนฐาน แทนยึดมอเตอร์ และส่วนล่างของตัวยึดเฟลมอเตอร์กับแขน

วงจรมอเตอร์ เมื่อเริ่มต้นเราได้ออกแบบวงจรมอเตอร์ และทดสอบวงจรมอเตอร์ จึงได้พบปัญหาที่ผลการนำมามีความไม่แน่นอน และวงจรมอเตอร์มีขนาดใหญ่เกินไป จึงแก้ไขโดยการนำ PIC แทน ซึ่งทำให้วงจรมอเตอร์มีขนาดเล็กและมีเสถียรภาพสูง เนื่องจาก PIC มีความเร็วสูง และเราใช้การพันสายตัวนำในวงจรมอเตอร์แทนการเชื่อมสายด้วยตะกั่ว เพื่อความสะดวก และง่ายในการเชื่อมวงจรมอเตอร์ และหาจุดบกพร่องเพื่อแก้ไข ทำให้มีข้อผิดพลาดน้อยกว่า และลดปัญหาการฉีกขาดของสายไฟ

วงจรมอเตอร์เคลื่อนที่ เนื่องจากมอเตอร์ที่นำมาใช้มีขนาดใหญ่ จึงกระแสมากจึงทำให้เกิดปัญหาด้านความร้อน ซึ่งก็ได้ทำการแก้ไขไปแล้วบางส่วน โดยติด cooling fan แต่ก็สามารถช่วยได้เพียงบางส่วนเท่านั้น ความร้อนที่เกิดขึ้นทำให้ค่า electrical characteristic ของอุปกรณ์ semiconductor เปลี่ยนไป เช่น Mosfet จะมีค่า Ron มากขึ้น ทำให้มอเตอร์มีความเร็วลดลง จึงมีผลต่อการควบคุมอย่างมาก แนวทางแก้ไขต่อไปคือ ต้องมีการชดเชยค่า input ของระบบเมื่ออุณหภูมิเกิดการเปลี่ยนแปลง

ข้อเสนอแนะเพิ่มเติม

จากการตัดสินใจเลือกใช้วงจรมอเตอร์ควบคุมแรงดัน เนื่องจากความง่าย ความสะดวกในการทำวงจรมอเตอร์ แต่เราจะพบปัญหาในการหาโมเดลของระบบ ที่ไม่ค่อยสอดคล้องกับโครงสร้างของระบบ เกิดความยุ่งยากตามมาในการออกแบบระบบควบคุมแบบป้อนกลับ ซึ่งควรจะใช้วงจรมอเตอร์กระแส เพื่อควบคุมแรงบิดแทน

บรรณานุกรม

1. ชานินทร์ ถาวรศาสนวงศ์, “ การอินเทอร์เน็ตเฟส ”, สำนักพิมพ์ฟิสิกส์เซ็นเตอร์, กรุงเทพฯ, พศ.2536.
2. KATSUHIKO OGATA , “ MODERN CONTROL ENGINEERING ” , SECOND EDITION, PRENTICE-HALL INTERNATIONAL, 1990.
3. ผศ.น.ท.ดร.สุธรรม ศรีเกษม, “ MATLAB เพื่อการแก้ปัญหาทางวิศวกรรม ” , สำนักพิมพ์ศรีอนันต์การพิมพ์, กรุงเทพฯ, พศ.2521.
4. SHAHIAN HASSUL, “ CONTROL SYSTEM DESIGN USING MATLAB ” , PRENTICE-HALL INTERNATIONAL, 1993.
5. บัณฑิต จามรภูติ, “ ดอสอินเทอร์เน็ตเฟส ” , บริษัท ซีเอ็ดดูเคชั่น จำกัด, พศ.2539.
6. ชันวา ศรีประโมง, “ การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม ” , พิมพ์ครั้งที่สี่, จัดพิมพ์โดยมหาวิทยาลัยเทคโนโลยีมหานคร, พศ.2539.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมควบคุมระบบแบบ State Feedback

```
#include <DOS.H>
#include <CONIO.H>
#include <MATH.H>
#include <STDIO.H>

#define TCK 0x0d
#define DATA 3333

void interrupt (*Old_ISR)();
float PID(float);

volatile int port300,port301,port302,port303;
volatile float T=0.005,Mi,Mp,Kp,Ki,Md,Kd,y0,y1;

void interrupt My_ISR()
{
    disable();
    outportb(0x303,port303);
    port300 = inportb(0x300);
    port301 = inportb(0x301);
    port302 = inportb(0x302);
    outportb(0x20,0x20);    /* End of Interrupt */
    enable();
}

void main()
{
    int    sp1,sp2,encoder1,encoder2,encoder2_t,bin,i;
    float  teta1,teta1_t,teta1dot,teta1dot_t,teta1dotdot,teta2,teta2_t,teta2dot;
    float  Kteta1,Kteta1dot,Kteta1dotdot,Kteta2,Kteta2dot;
```

```

float   fb1,fb1dot,fb1dotdot,fb2,fb2dot,output,err;
FILE    *fp1,*fp2;

i=0;
sp1=2500;
encoder2_t=0;
outportb(0x305,0x00); /* Clear All PIC Microcontroller */
delay(5);
outportb(0x305,0x0f); /* Enable All PIC Microcontroller */
outportb(0x303,0x00); /* Clear PWM */
outportb(0x304,0xff); /* Set Samping Period */
Old_ISR = getvect(TCK); /* Recieve Interrupt Vector */
setvect(TCK,My_ISR); /* Set Interrupt Vector to My ISR */
outportb(0x21,(inportb(0x21) & 0xdf)); /* Enable IRQ5 */

if((fp1 = fopen("teta1.m","w"))==NULL)
{
    puts("cannot open file teta1.m\n");
    exit(1);
}
fprintf(fp1,"a = [ ");

if((fp2 = fopen("teta2.m","w"))==NULL)
{
    puts("cannot open file teta2.m\n");
    exit(1);
}
fprintf(fp2,"a = [ ");

while (!kbhit())
{
    encoder2=((port301&0xf)*256)+port300;
    if (encoder2<2000) encoder2=encoder2+2000;
    else encoder2=encoder2-2000;
    encoder1=(port302*16)+((port301&0xf0)/16);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (encoder1<2000) encoder1=encoder1+2000;
        else encoder1=encoder1-2000;
        printf("Encoder1 = %d Encoder2 = %d\n",encoder1,encoder2);
    }

    getch();

    Mi=0.0;
    Kp=5.0;
    Ki=0.25;
    Kd=2.25;
    while ( !kbhit() & encoder2>25 & encoder2<3974 )
    {
        encoder2_t=encoder2;
        encoder2=((port301&0xf)*256)+port300;
        if (encoder2<2000) encoder2=encoder2+2000;
        else encoder2=encoder2-2000;
        encoder1=(port302*16)+((port301&0xf0)/16);
        if (encoder1<2000) encoder1=encoder1+2000;
        else encoder1=encoder1-2000;
        err=((float)(sp1-encoder1))*0.001571;
        if (encoder2>2000 & (encoder2-encoder2_t)<0) sp1=1500;
        if (encoder2<2000 & (encoder2-encoder2_t)>0) sp1=2500;
        if (encoder2>3500 | encoder2<500) Kp=3.25;
        output=PID(err);
        printf("Encoder1 = %d Encoder2 = %d\n",encoder1,encoder2);
        if (output<0)
            {
                output=0-output;
                bin=(128/5)*output;
                port303=bin&0x7f;
            }
        else
            {
                bin=(128/5)*output;
                port303=bin|0x80;
            }
    }
}

```

```

    }

    }

port303=0x00;
sp1=2000;
sp2=2000;
teta1_t=0;
teta1dot_t=0;
teta2_t=0;
Kteta1=3.0;
Kteta1dot=3.0;
Kteta1dotdot=0;
Kteta2=-17.0;
Kteta2dot=-4.5;
encoder2=((port301&0xf)*256)+port300;

while (!kbhit() & encoder2!=2000) encoder2=((port301&0xf)*256)+port300;

while (!kbhit() & encoder2<2200 & encoder2 >1800 & i<DATA)
{
    encoder1=(port302*16)+((port301&0xf0)/16);
    if (encoder1<2000) encoder1=encoder1+2000;
    else encoder1=encoder1-2000;
    encoder2=((port301&0xf)*256)+port300;
    teta1=((float)(sp1-encoder1))*0.001571;
    teta1dot=(teta1-teta1_t)/T;
    teta1dotdot=(teta1dot-teta1dot_t)/T;
    teta2=((float)(sp2-encoder2))*0.001571;
    teta2dot=(teta2-teta2_t)/T;
    teta1_t=teta1;
    teta1dot_t=teta1dot;
    teta2_t=teta2;

    fprintf(fp1,"%f %f;\n",0.003*i,teta1);
    fprintf(fp2,"%f %f;\n",0.003*i,teta2);
    i=i++;
}

```

```

printf("teta1=%f teta2=%f\n",teta1,teta2);

fb1=Kteta1*teta1;

fb1dot=Kteta1dot*teta1dot;

fb1dotdot=Kteta1dotdot*teta1dotdot;

fb2=Kteta2*teta2;

fb2dot=Kteta2dot*teta2dot;

output=-fb1-fb1dot-fb1dotdot-fb2-fb2dot;

if (output<0)
{
output=0-output;
bin=(128/5)*output;
port303=bin&0x7f;
}
else
{
bin=(128/5)*output;
port303=bin|0x80;
}
}

port303=0x00;
outportb(0x303,0x00); /* Clear PWM */
outportb(0x21,(inportb(0x21) | 0x20)); /* Disable IRQ5 */
setvect(TCK,Old_ISR); /* Return Interrupt Vector to Old ISR */

fprintf(fp1,"%f %f\n",0.003*i,teta1);
fprintf(fp2,"%f %f\n",0.003*i,teta2);

fprintf(fp1,"x = a(:, 1);\n");
fprintf(fp2,"x = a(:, 1);\n");

fprintf(fp1,"y = a(:, 2);\n");
fprintf(fp2,"y = a(:, 2);\n");

fprintf(fp1,"plot(x,y), ... \n");
fprintf(fp2,"plot(x,y), ... \n");

fprintf(fp1,"xlabel ('t secs');\n");
fprintf(fp1,"ylabel ('rad');\n");

```

```

fprintf(fp1,"title('Graph of Teta1'), ...\n");
fprintf(fp2,"xlabel ('t secs');\n");
fprintf(fp2,"ylabel ('rad');\n");
fprintf(fp2,"title('Graph of Teta2'), ...\n");
fprintf(fp1,"grid\n");
fprintf(fp2,"grid\n");
fclose(fp1);
fclose(fp2);
}

```

```

float PID(float y1)
{
    Mp=Kp*y1;
    Mi=Mi+Ki*y1*T;
    Md=Kd*(y1-y0)/T;
    y0=y1;
    return(Mp+Mi+Md);
}

```



โปรแกรมควบคุมระบบแบบ PID

```
#include <DOS.H>
#include <CONIO.H>
#include <MATH.H>
#include <STDIO.H>

#define TCK 0x0d
#define DATA 4000

void interrupt (*Old_ISR)();
float PID(float);

volatile int port300,port301,port302,port303;
volatile float T=0.005,Mi,Mp,Kp,Ki,Md,Kd,y0,y1;
void interrupt My_ISR()
{
    disable();
    outportb(0x303,port303);
    port300 = inportb(0x300);
    port301 = inportb(0x301);
    port302 = inportb(0x302);
    outportb(0x20,0x20);    /* End of Interrupt */
    enable();
}

void main()
{
    int    sp1,sp2,encoder1,encoder2,encoder2_t,bin,i;
    float  output,temp,y1,err,m;
    FILE  *fp1;

    i=0;
```

```

sp1=2500;
encoder2_t=0;

outputb(0x305,0x00); /* Clear All PIC Microcontroller */
delay(5);
outputb(0x305,0x0f); /* Enable All PIC Microcontroller */
outputb(0x303,0x00); /* Clear PWM */
outputb(0x304,0xff); /* Set Sampling Period */
Old_ISR = getvect(TCK); /* Recieve Interrupt Vector */
setvect(TCK,My_ISR); /* Set Interrupt Vector to My ISR */
outputb(0x21,(inportb(0x21) & 0xdf)); /* Enable IRQ5 */

if((fp1 = fopen("pid.m","w"))==NULL)
{
puts("cannot open file pid.m\n");
exit(1);
}
fprintf(fp1,"a = [ *);

while (!kbhit())
{
encoder2=((port301&0xf)*256)+port300;
if (encoder2<2000) encoder2=encoder2+2000;
else encoder2=encoder2-2000;
encoder1=(port302*16)+((port301&0xf0)/16);
if (encoder1<2000) encoder1=encoder1+2000;
else encoder1=encoder1-2000;
printf("Encoder1 = %d Encoder2 = %d\n",encoder1,encoder2);
}

getch();

Mi=0.0;
Kp=5.0;
Ki=0.25;
Kd=2.25;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while ( !kbhit() & encoder2>25 & encoder2<3974 )
{
    encoder2_t=encoder2;
    encoder2=((port301&0xf)*256)+port300;
    if (encoder2<2000) encoder2=encoder2+2000;
    else encoder2=encoder2-2000;
    encoder1=(port302*16)+((port301&0xf0)/16);
    if (encoder1<2000) encoder1=encoder1+2000;
    else encoder1=encoder1-2000;
    err=((float)(sp1-encoder1))*0.001571;
    if (encoder2>2000 & (encoder2-encoder2_t)<0) sp1=1500;
    if (encoder2<2000 & (encoder2-encoder2_t)>0) sp1=2500;
    if (encoder2>3500 | encoder2<500) Kp=3.25;
    output=PID(err);
    printf("Encoder1 = %d Encoder2 = %d\n",encoder1,encoder2);
    if (output<0)
    {
        output=0-output;
        bin=(128/5)*output;
        port303=bin&0x7f;
    }
    else
    {
        bin=(128/5)*output;
        port303=bin|0x80;
    }
}

port303=0x00;
sp2=2000;
Mi=0;
Kp=5.5;
Ki=4.0;
Kd=4.0;

encoder2=((port301&0xf)*256)+port300;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while (!kbhit() & encoder2!=2000) encoder2=((port301&0xf)*256)+port300;
```

```
while (!kbhit() & encoder2<2100 & encoder2>1900 & i<DATA)
```

```
{
```

```
    encoder2=((port301&0xf)*256)+port300;
```

```
    err=((float)(sp2-encoder2))*0.001571;
```

```
    fprintf(fp1,"%f %f;\n",0.0025*i,err);
```

```
    i=i++;
```

```
    printf("Encoder2=%d error=%f\n",encoder2,err);
```

```
    output=PID(err);
```

```
    if (output<0)
```

```
    {
```

```
        output=0-output;
```

```
        bin=(128/5)*output;
```

```
        port303=bin&0x7f;
```

```
    }
```

```
    else
```

```
    {
```

```
        bin=(128/5)*output;
```

```
        port303=bin|0x80;
```

```
    }
```

```
}
```

```
outportb(0x303,0x00); /* Clear PWM */
```

```
outportb(0x21,(inportb(0x21) | 0x20)); /* Disable IRQ5 */
```

```
setvect(TCK,Old_ISR); /* Return Interrupt Vector to Old ISR */
```

```
fprintf(fp1,"%f %f]\n",0.0025*i,err);
```

```
fprintf(fp1,"x = a(:, 1);\n");
```

```
fprintf(fp1,"y = a(:, 2);\n");
```

```
fprintf(fp1,"plot(x,y), ...;\n");
```

```
fprintf(fp1,"xlabel ('t secs');\n");
```

```
fprintf(fp1,"ylabel ('rad');\n");
```

```
fprintf(fp1,"title('Graph of PID'), ...;\n");
```

```
fprintf(fp1,"grid\n");
```

```
fclose(fp1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float PID(float y1)
{
    Mp=Kp*y1;
    Mi=Mi+Ki*y1*T;
    Md=Kd*(y1-y0)/T;
    y0=y1;
    return(Mp+Mi+Md);
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสร้างสัญญาณ PWM

device pic16c56,hs_osc,wdt_off,protect_off

```
register1    =    08h
register2    =    09h

main        mov    !porta,#00h    ;a0 output
            mov    !portb,#0ffh    ;b0-b6 input
            mov    porta,#00h

loop_main   mov    register1,portb
            and    register1,#01111111b
            cjne   register1,#00000000b,next
            clrb   porta.0
            jmp    loop_main

next        mov    register2,#10000000b
            sub    register2,register1

            setb   porta.0

loop1       djnz   register1,loop1
            clrb   porta.0

loop2       djnz   register2,loop2
            setb   porta.0
            jmp    loop_main

end
```

โปรแกรมสร้างสัญญาณ Sampling Period

```
device pic16c56,hs_osc,wdt_off,protect_off
```

```
ontime    =    08h
offtime   =    09h
delay     =    0ah

main      mov    !porta,#00h    ;a0 output
          mov    !portb,#0ffh   ;b0-b7 input
          mov    porta,#00h

loop      mov    ontime,#0ffh
          mov    offtime,portb
          setb   porta.0

onloop    djnz   ontime,onloop
          clrb   porta.0

offloop1  mov    delay,#0ah

offloop2  djnz   delay,offloop2
          djnz   offtime,offloop1
          jmp    loop

end
```

โปรแกรมแปลงสัญญาณ Encoder เฟส A และ B เป็นค่าตำแหน่ง

```
device pic16c57,hs_osc,wdt_off,protect_off

main      mov    !porta,#03h
          mov    !portb,#00h
          mov    !portc,#00h

          mov    portb,#00h
          mov    portc,#00h

          snb    porta.1
          jmp    phase1x
          jnb    porta.0,phase10
          jmp    phase11

phase1x   jnb    porta.0,phase00
          jmp    phase01

phase00   jb     porta.0,detect01_up
          jb     porta.1,detect10_down
          jmp    phase00

phase01   jnb    porta.0,detect00_down
          jb     porta.1,detect11_up
          jmp    phase01

phase11   jnb    porta.0,detect10_up
          jnb    porta.1,detect01_down
          jmp    phase11

phase10   jb     porta.0,detect11_down
          jnb    porta.1,detect00_up
```

```

        jmp     phase10

detect00_up  cje     portb,#9fh,testb00_up
next00_up    incsz   portb
             jmp     phase00
             inc     portc
             jmp     phase00

testb00_up   cse     portc,#0fh
             jmp     next00_up
             mov     portb,#00h
             mov     portc,#00h
             jmp     phase00

detect01_up  cje     portb,#9fh,testb01_up
next01_up    incsz   portb
             jmp     phase01
             inc     portc
             jmp     phase01

testb01_up   cse     portc,#0fh
             jmp     next01_up
             mov     portb,#00h
             mov     portc,#00h
             jmp     phase01

detect11_up  cje     portb,#9fh,testb11_up
next11_up    incsz   portb
             jmp     phase11
             inc     portc
             jmp     phase11

testb11_up   cse     portc,#0fh
             jmp     next11_up
             mov     portb,#00h

```

```

        mov     portc,#00h
        jmp     phase11

detect10_up  cje     portb,#9fh,testb10_up
next10_up   incsz   portb
            jmp     phase10
            inc     portc
            jmp     phase10

testb10_up  cse     portc,#0fh
            jmp     next10_up
            mov     portb,#00h
            mov     portc,#00h
            jmp     phase10

detect00_down cje     portb,#00h,testc00_down
            dec     portb
            jmp     phase00

testc00_down cje     portc,#00h,next00_down
            dec     portb
            dec     portc
            jmp     phase00

next00_down mov     portb,#9fh
            mov     portc,#0fh
            jmp     phase00

detect01_down cje     portb,#00h,testc01_down
            dec     portb
            jmp     phase01

testc01_down cje     portc,#00h,next01_down
            dec     portb
            dec     portc
            jmp     phase01

```

```

next01_down  mov    portb,#9fh
              mov    portc,#0fh
              jmp    phase01

detect11_down cje    portb,#00h,testc11_down
              dec    portb
              jmp    phase11

testc11_down  cje    portc,#00h,next11_down
              dec    portb
              dec    portc
              jmp    phase11

next11_down   mov    portb,#9fh
              mov    portc,#0fh
              jmp    phase11

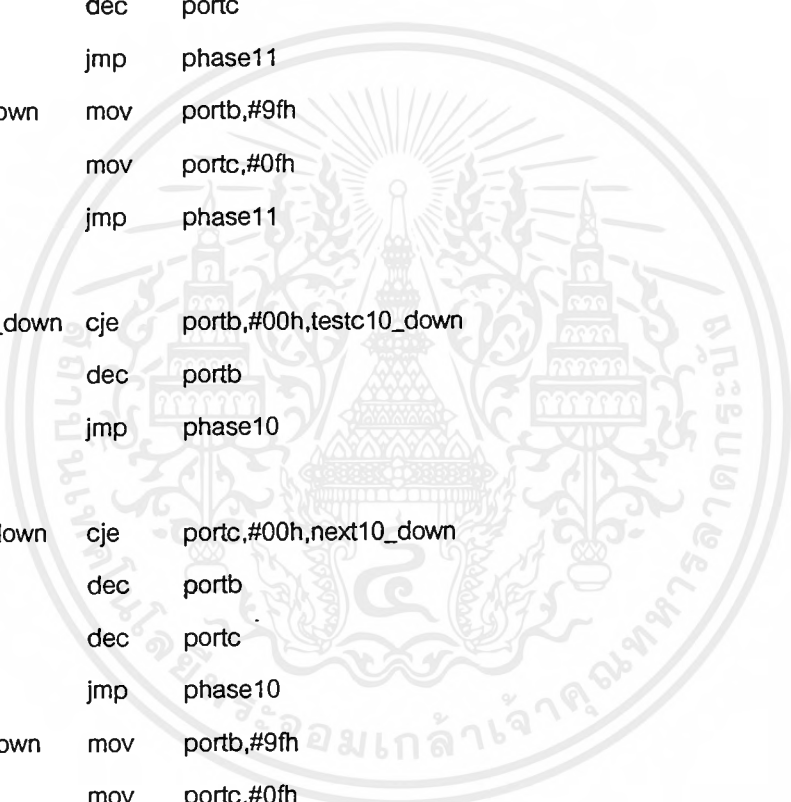
detect10_down cje    portb,#00h,testc10_down
              dec    portb
              jmp    phase10

testc10_down  cje    portc,#00h,next10_down
              dec    portb
              dec    portc
              jmp    phase10

next10_down   mov    portb,#9fh
              mov    portc,#0fh
              jmp    phase10

              end

```



Instruction Set ของ PIC16C5X

***** Parallax PIC16Cxx Assembler PASM v4.5 '5x Test File' *****

(formatted with 8-space tabs)

; NOTE: Assembler is case-insensitive, except within strings

; DEVICE directive - must precede all other directives/instructions

; with the exception of INCLUDE

device pic16c58,lp_osc,wdt_off,protect_off

; Equates

expression	=	0	
fr	=	0h	;0-7Fh
fr1	=	0h	;0-7Fh
fr2	=	0h	;0-7Fh
d	=	0	;0 or 1
b	=	0	;0-07h
bit	=	fr.b	;0-3FFh
bit1	=	bit	;0-3FFh
bit2	=	bit+1	;0-3FFh
port	=	5h	;5, 6, or 7
literal	=	0h	;0-0FFh
address	=	0h	;0-3FFh

```

:
:
; Directives
:
    id    1234h    ;Set device id to 1234h
        •
symbol_2 =    expression    ;Equate a symbol to a value

symbol_1 equ    expression    ;Same as =

    org    expression    ;Set code origin

    ds    expression    ;Define space in bytes

    reset    address    ;Set reset JMP

;include 'stuff.asm'    ;Include another source file

;end                    ;End assembly (optional)
:
:
; Data Types
:

    retw    100        ;decimal 100
    retw    27h        ;hex 27
    retw    0B6h        ;hex B6
    retw    10101110b    ;binary 10101110
    retw    'A'        ;ASCII value 'A' (41h)
    retw    $          ;current origin

    djnz    fr,$        ;use $ for delay loops, etc.
    jb     bit,$

```

; Operators and expressions (resolved strictly from left to right)

; . = BIT ADDRESS

; & = AND

; | = OR

; ^ = XOR

; + = ADD

; - = SUBTRACT

; * = MULTIPLY

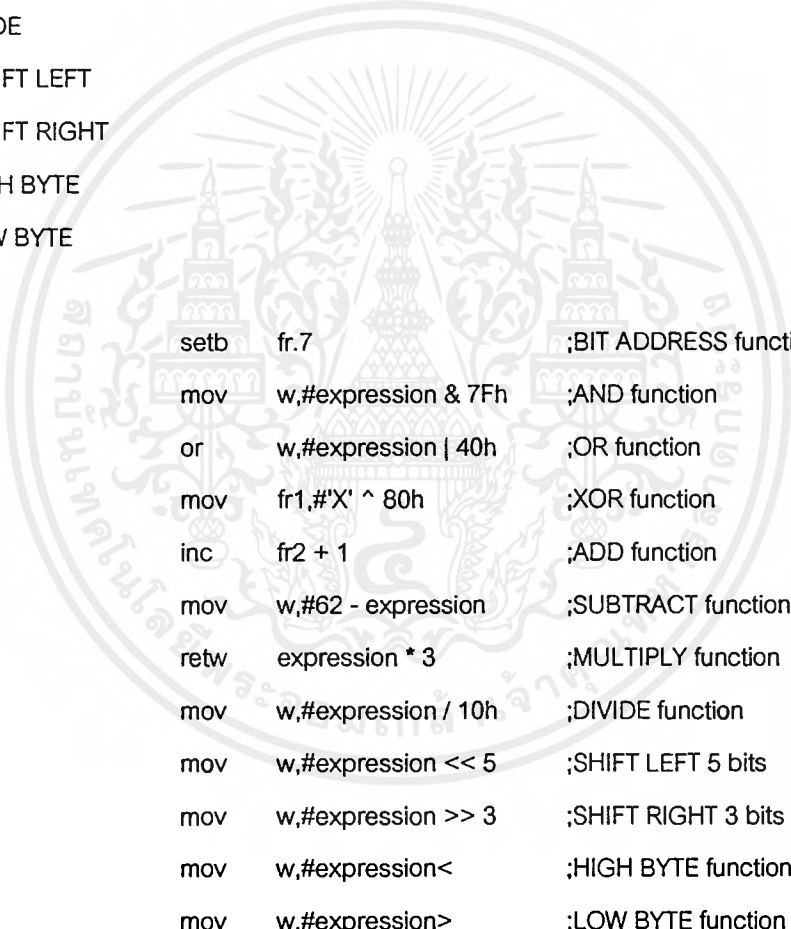
; / = DIVIDE

; <<= SHIFT LEFT

; >>= SHIFT RIGHT

; < = HIGH BYTE

; > = LOW BYTE



```
setb    fr.7           ;BIT ADDRESS function
mov     w,#expression & 7Fh ;AND function
or      w,#expression | 40h ;OR function
mov     fr1,#'X' ^ 80h    ;XOR function
inc     fr2 + 1          ;ADD function
mov     w,#62 - expression ;SUBTRACT function
retw    expression * 3   ;MULTIPLY function
mov     w,#expression / 10h ;DIVIDE function
mov     w,#expression << 5 ;SHIFT LEFT 5 bits
mov     w,#expression >> 3 ;SHIFT RIGHT 3 bits
mov     w,#expression<   ;HIGH BYTE function
mov     w,#expression>   ;LOW BYTE function
```

; Symbols can be up to 32 characters and may contain alpha, numeric, '_', and

; ':' characters. They must not begin with a numeric character.

```

;
; Local symbols
;
Routine_1      mov    w,#100      ;A local symbol can be called
               mov    fr,W        ;...within its regular-symbol
:Loop          call   Some_Routine ;...boundary by its own name.
               djnz  fr,:Loop
               ret

Routine_2      mov    w,#200      ;Local symbols can be used over
               mov    fr,W        ;...and over again.
:Loop          call   Some_Routine
               djnz  fr,:Loop
               ret

Routine_3      mov    w,#250      ;A local symbol can be called
               mov    fr,W        ;...outside its regular-symbol
               jmp   Routine_2:Loop ;...boundary as shown.

Some_Routine   ret
;
;
; Using DS to allocate sequential variables
;
               org    0Ch        ;Set variables' origin.
byte_variable_1 ds    1          ;List symbolic variables with length.
byte_variable_2 ds    1          ;This way, equate 'stacking' is avoided
byte_variable_3 ds    1          ;...in naming sequential variables.
word_variable_1 ds    2          ;Also, variables can be moved without
word_variable_2 ds    2          ;...typing new equate operands.
word_variable_3 ds    2          ;Of course, no code is generated.
string_variable ds    8

               org    40h        ;Now that variables are declared,
               ;...set up code origin.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; Using RETW to define complex strings
```

```
String          retw    'This string has an inverted end chr.'^80h
                retw    0FFh^'This string has an inversed first chr.'
                retw    'This string ends with 0.',0
                retw    1,'This string starts with 1.'
```

```
*****
;*
;* PARALLAX INSTRUCTION SET *
;*
*****
```

```
; Byte-oriented operations
```

```
mov    w,#literal          ;mov's
mov    w,fr
mov    fr,w
mov    fr,#literal
mov    fr1,fr2

add    w,fr                ;add's
add    fr,w
add    fr,#literal
add    fr1,fr2

mov    w,fr-w              ;sub's
sub    fr,w
sub    fr,#literal
sub    fr1,fr2
```

```
and    w,#literal      ;and's
and    w,fr
and    fr,w
and    fr,#literal
and    fr1,fr2
```

```
or     w,#literal      ;or's
or     w,fr
or     fr,w
or     fr,#literal
or     fr1,fr2
```

```
xor    w,#literal      ;xor's
xor    w,fr
xor    fr,w
xor    fr,#literal
xor    fr1,fr2
```

```
clr    w                ;clear's
clr    fr
```

```
mov    w,++fr          ;inc's
inc    fr
```

```
mov    w,-fr           ;dec's
dec    fr
```

```
mov    w,>>fr          ;right shifts
rr     fr
```

```
mov    w,<<fr          ;left shifts
rl     fr
```

```
mov    w,<>fr          ;nibble swaps
swap   fr
```

```

not    w                ;not's
mov    w,/fr
not    fr

test   w                ;test's
test   fr

```

```

; Bit-oriented operations

```

```

clrb   bit              ;bit set/clear's
clc
clz

setb   bit
stc
stz

addb   fr,bit           ;bit add/sub's
addb   fr,/bit
subb   fr,bit
subb   fr,/bit

movb   bit1,bit2       ;bit moves
movb   bit1,/bit2

```

```

; Inc/dec conditional branches

```

```

movsz  w,++fr          ;inc/dec with skip if zero
incsz  fr
ijnz   fr,address

movsz  w,--fr

```

```
decsz fr
djnz fr,address
```

```
; Compare-conditional branches
```

```
csa fr,#literal ;compare skips
csa fr1,fr2
csae fr,#literal
csae fr1,fr2
csb fr,#literal
csb fr1,fr2
csbe fr,#literal
csbe fr1,fr2
cse fr,#literal
cse fr1,fr2
csne fr,#literal
csne fr1,fr2
cja fr,#literal,address;compare jumps
cja fr1,fr2,address
cjae fr,#literal,address
cjae fr1,fr2,address
cjb fr,#literal,address
cjb fr1,fr2,address
cjbe fr,#literal,address
cjbe fr1,fr2,address
cje fr,#literal,address
cje fr1,fr2,address
cjne fr,#literal,address
cjne fr1,fr2,address
```

; Bit-conditional branches

sb bit ;bit skips

sc

sz

snb bit

snc

snz

jb bit,address ;bit jumps

jc address

jz address

jnb bit,address

jnc address

jnz address

; Unconditional branches

skip ;skip

jmp address ;jumps

jmp pc+w

jmp w

call address ;call/return

ret

retw 'String'\80h

lset address ;long addressing

ljmp address

lcall address

```
; I/O and control operations
```

```
mov    !port,w           ;tristate registers
mov    !port,#literal
mov    !port,fr

mov    !option,w         ;option register
mov    !option,#literal
mov    !option,fr
```

```
clr    wdt               ;miscellaneous
sleep

nop                    ;no operation
```

```
*****
;* MICROCHIP INSTRUCTION SET
;*
*****
```

```
addwf  fr,d             ;byte-oriented file
addwf  fr,w             ;register operations
addwf  fr
andwf  fr,d
andwf  fr,w
andwf  fr
clrf   fr
clrw

comf   fr,d
comf   fr,w
comf   fr
```

decf	fr,d
decf	fr,w
decf	fr
decfsz	fr,d
decfsz	fr,w
decfsz	fr
incf	fr,d
incf	fr,w
incf	fr
incfsz	fr,d
incfsz	fr,w
incfsz	fr
iorwf	fr,d
iorwf	fr,w
iorwf	fr
movf	fr,d
movf	fr,w
movf	fr
movwf	fr
nop	
rff	fr,d
rff	fr,w
rff	fr
rrf	fr,d
rrf	fr,w
rrf	fr
subwf	fr,d
subwf	fr,w
subwf	fr
swapf	fr,d
swapf	fr,w
swapf	fr
xorwf	fr,d
xorwf	fr,w
xorwf	fr



bcf	fr,b	;bit-oriented file
bcf	bit	;register operations
bsf	fr,b	
bsf	bit	
btfs	fr,b	
btfs	bit	
btfs	fr,b	
btfs	bit	

andlw	literal	;literal and control
call	address	;operations

clrw

goto	address
------	---------

iorlw	literal
-------	---------

movlw	literal
-------	---------

option

retlw	literal
-------	---------

sleep

tris	port
------	------

xorlw	literal
-------	---------

clrc		;special instructions
------	--	-----------------------

setc

clr

setd

clr

setz

skpc

skpnc

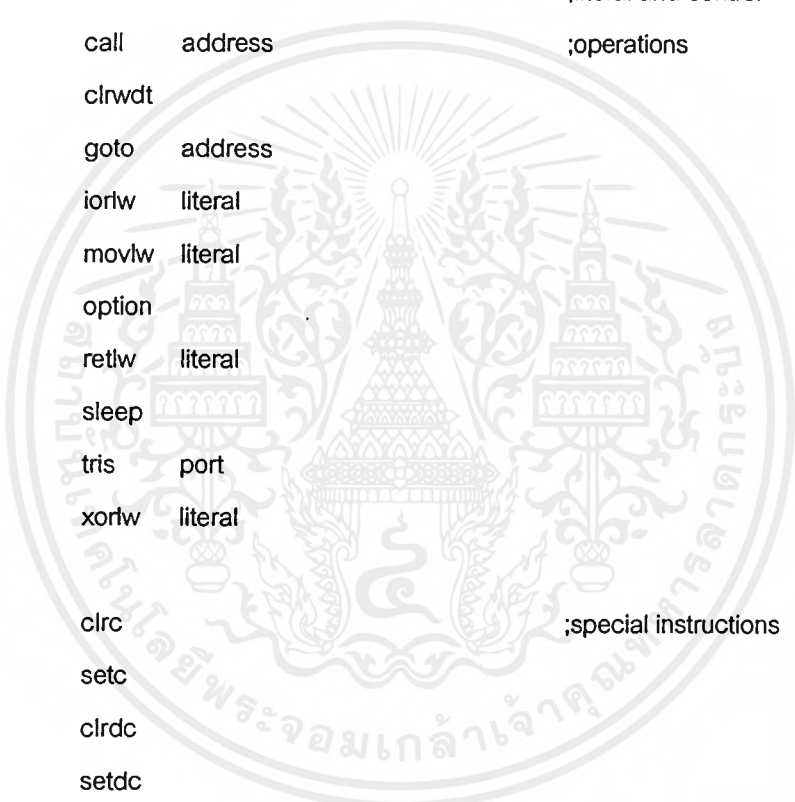
skpdc

skpndc

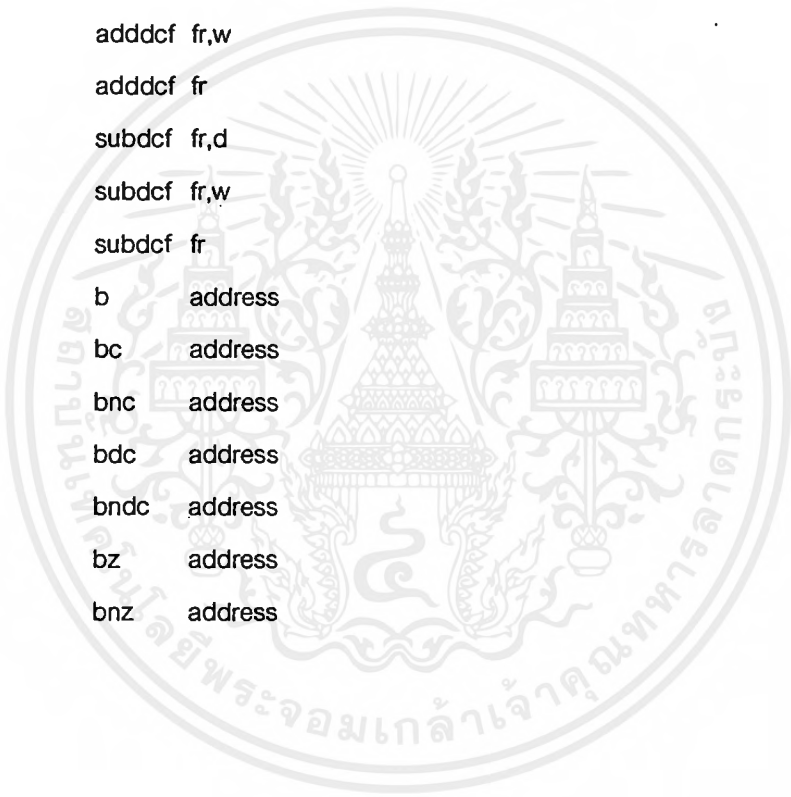
skpz

skpnz

tstf	fr
------	----



movfw fr
negf fr,d
negf fr,w
negf fr
addcf fr,d
addcf fr,w
addcf fr
subcf fr,d
subcf fr,w
subcf fr
adddcf fr,d
adddcf fr,w
adddcf fr
subdcf fr,d
subdcf fr,w
subdcf fr
b address
bc address
bnc address
bdc address
bndc address
bz address
bnz address



โปรแกรมแปลงสัญญาณ Encoder เฟส A และ B เป็นค่าตำแหน่ง

```
device pic16c57,hs_osc,wdt_off,protect_off

main      mov    !porta,#03h
          mov    !portb,#00h
          mov    !portc,#00h

          mov    portb,#00h
          mov    portc,#00h

          snb    porta.1
          jmp    phase1x
          jnb    porta.0,phase10
          jmp    phase11

phase1x   jnb    porta.0,phase00
          jmp    phase01

phase00   jb     porta.0,detect01_up
          jb     porta.1,detect10_down
          jmp    phase00

phase01   jnb    porta.0,detect00_down
          jb     porta.1,detect11_up
          jmp    phase01

phase11   jnb    porta.0,detect10_up
          jnb    porta.1,detect01_down
          jmp    phase11

phase10   jb     porta.0,detect11_down
          jnb    porta.1,detect00_up
```

ภาคผนวก ข

คำจำเพาะของมอเตอร์

DC Motor 100 V. M3300 Series

Motor size M-3358-C

พิกัดแรงดันไฟฟ้า	V	100	-	-
พิกัดกระแสไฟฟ้า	A	6	-	-
พิกัดกระแสไฟฟ้าสูงสุด	A	37	-	-
ความเร็วรอบ	rpm	5000	-	-
พิกัดแรงบิด	LB-IN	8.8	NM	0.99
พิกัดแรงบิดสูงสุด	LB-IN	59.4	NM	6.71
ค่าคงที่แรงบิด	LB-IN/A	1.61	NM/A	0.181
แรงดันไฟฟ้าย้อนกลับ	V/Krpm	19	-	-
ความต้านทานอาร์เมเจอร์	OHMS	1.2	-	-
ความเหนี่ยวนำอาร์เมเจอร์	mH	3.6	-	-
โมเมนต์ความเฉื่อย	LB-IN-S ²	0.0024	KG-CM ²	2.75
น้ำหนัก	LBS	10	KG	4.5

High-Current Complementary Silicon Transistors

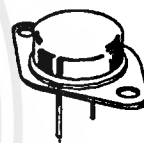
... for use as output devices in complementary general purpose amplifier applications.

- High DC Current Gain — $h_{FE} = 1000$ (Min) @ $I_C = 20$ Adc
- Monolithic Construction with Built-in Base Emitter Shunt Resistor
- Junction Temperature to $+200^\circ\text{C}$

PNP
MJ11013
MJ11015
NPN
MJ11012
MJ11014
MJ11016*

*Motorola Preferred Device

30 AMPERE
DARLINGTON
POWER TRANSISTORS
COMPLEMENTARY
SILICON
60-120 VOLTS
200 WATTS



CASE 1-07
TO-204AA
(TO-3)

MAXIMUM RATINGS

Rating	Symbol	MJ11012	MJ11013 MJ11014	MJ11015 MJ11016	Unit
Collector-Emitter Voltage	V_{CEO}	60	90	120	Vdc
Collector-Base Voltage	V_{CB}	60	90	120	Vdc
Emitter-Base Voltage	V_{EB}		5		Vdc
Collector Current	I_C		30		Adc
Base Current	I_B		1		Adc
Total Device Dissipation @ $T_C = 25^\circ\text{C}$ Derate above 25°C @ $T_C = 100^\circ\text{C}$	P_D		200 1.15		Watts W/ $^\circ\text{C}$
Operating Storage Junction Temperature Range	T_J, T_{stg}		-55 to +200		$^\circ\text{C}$

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Case	$R_{\theta JC}$	0.87	$^\circ\text{C}/\text{W}$
Maximum Lead Temperature for Soldering Purposes for ≤ 10 Seconds.	T_L	275	$^\circ\text{C}$

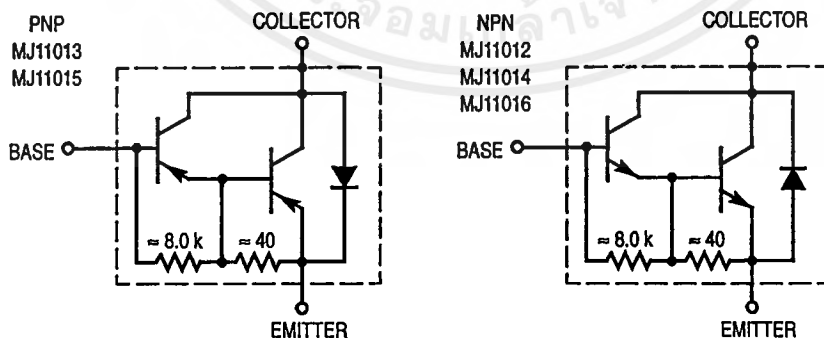


Figure 1. Darlington Circuit Schematic

Preferred devices are Motorola recommended choices for future use and best overall value.

REV 1

MJ11013 MJ11015 MJ11012 MJ11014 MJ11016

ELECTRICAL CHARACTERISTICS (T_C = 25°C unless otherwise noted.)

Characteristics	Symbol	Min	Max	Unit
-----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Breakdown Voltage(1) (I _C = 100 mA, I _B = 0)	MJ11012 MJ11013, MJ11014 MJ11015, MJ11016	V _{(BR)CEO}	60 90 120	— — —	V _{dc}
Collector-Emitter Leakage Current (V _{CE} = 60 Vdc, R _{BE} = 1k ohm) (V _{CE} = 90 Vdc, R _{BE} = 1k ohm) (V _{CE} = 120 Vdc, R _{BE} = 1k ohm) (V _{CE} = 60 Vdc, R _{BE} = 1k ohm, T _C = 150°C) (V _{CE} = 90 Vdc, R _{BE} = 1k ohm, T _C = 150°C) (V _{CE} = 120 Vdc, R _{BE} = 1k ohm, T _C = 150°C)	MJ11012 MJ11013, MJ11014 MJ11015, MJ11016 MJ11012 MJ11013, MJ11014 MJ11015, MJ11016	I _{CER}	— — — — — —	1 1 1 5 5 5	mA _{dc}
Emitter Cutoff Current (V _{BE} = 5 Vdc, I _C = 0)		I _{EBO}	—	5	mA _{dc}
Collector-Emitter Leakage Current (V _{CE} = 50 Vdc, I _B = 0)		I _{CEO}	—	1	mA _{dc}

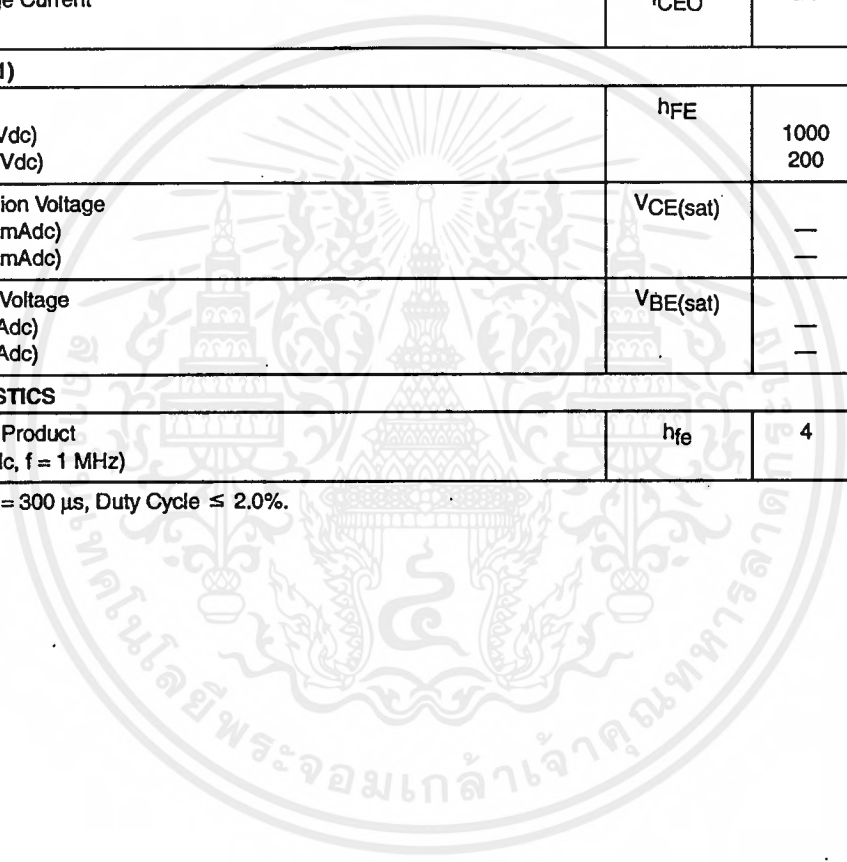
ON CHARACTERISTICS(1)

DC Current Gain (I _C = 20 A, V _{CE} = 5 Vdc) (I _C = 30 A, V _{CE} = 5 Vdc)		h _{FE}	1000 200	— —	—
Collector-Emitter Saturation Voltage (I _C = 20 A, I _B = 200 mA _{dc}) (I _C = 30 A, I _B = 300 mA _{dc})		V _{CE(sat)}	— —	3 4	V _{dc}
Base-Emitter Saturation Voltage (I _C = 20 A, I _B = 200 mA _{dc}) (I _C = 30 A, I _B = 300 mA _{dc})		V _{BE(sat)}	— —	3.5 5	V _{dc}

DYNAMIC CHARACTERISTICS

Current-Gain Bandwidth Product (I _C = 10 A, V _{CE} = 3 Vdc, f = 1 MHz)		h _{fe}	4	—	MHz
---	--	-----------------	---	---	-----

(1) Pulse Test: Pulse Width = 300 μs, Duty Cycle ≤ 2.0%.



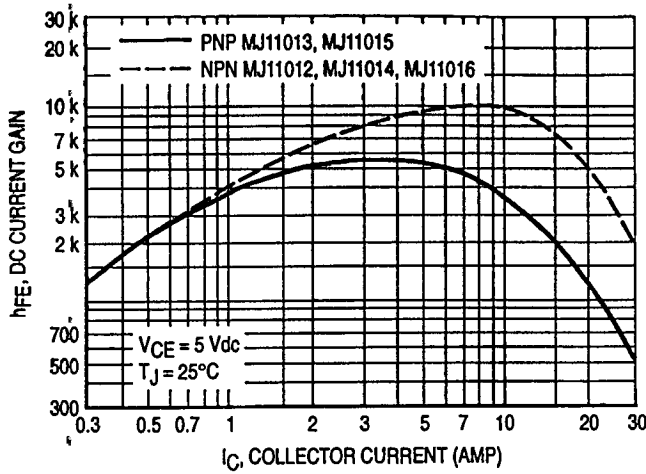


Figure 2. DC Current Gain (1)

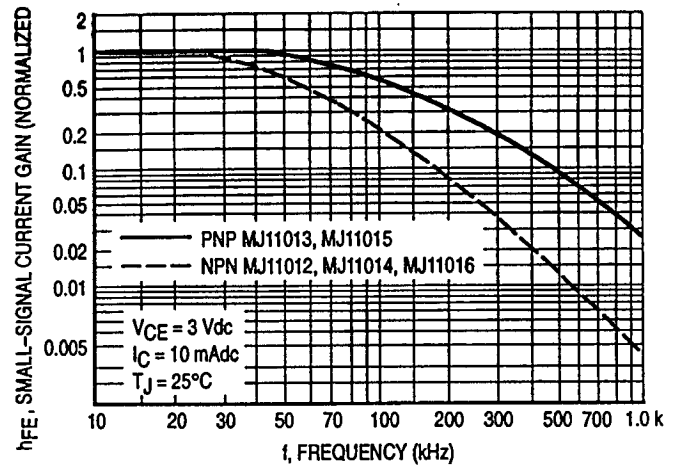


Figure 3. Small-Signal Current Gain

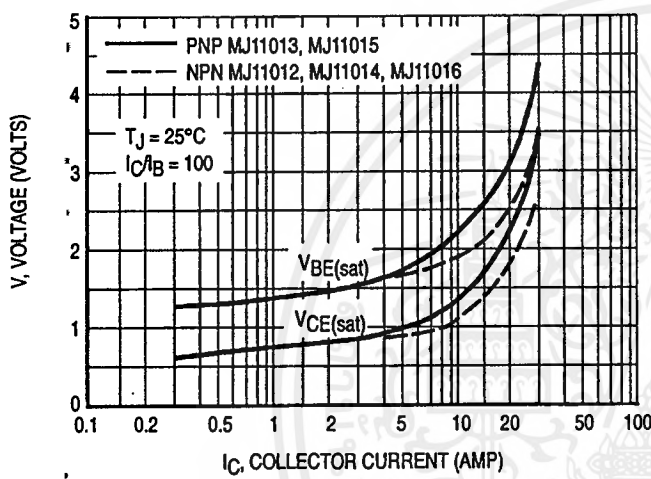


Figure 4. "On" Voltages (1)

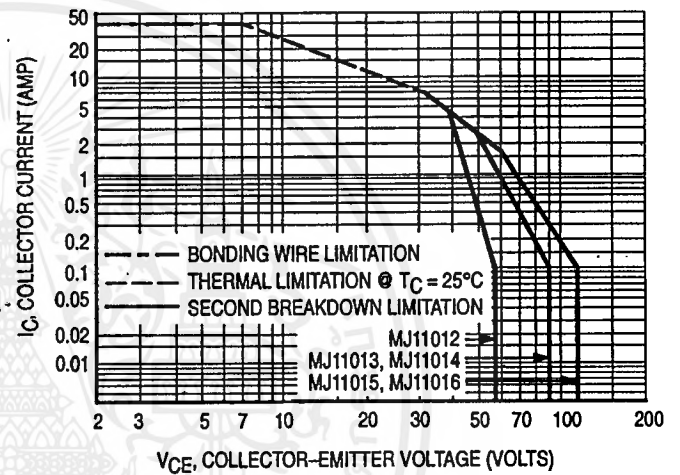


Figure 5. Active Region DC Safe Operating Area

There are two limitations on the power handling ability of a transistor average junction temperature and secondary breakdown. Safe operating area curves indicate $I_C - V_{CE}$ limits of the transistor that must be observed for reliable operations e.g., the transistor must not be subjected to greater

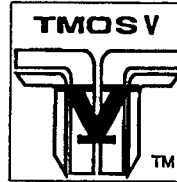
dissipation than the curves indicate.

At high case temperatures, thermal limitations will reduce the power that can be handled to values less than the limitations imposed by secondary breakdown.

Designer's™ Data Sheet
TMOS V™

Power Field Effect Transistor
N-Channel Enhancement-Mode Silicon Gate

TMOS V is a new technology designed to achieve an on-resistance area product about one-half that of standard MOSFETs. This new technology more than doubles the present cell density of our 50 and 60 volt TMOS devices. Just as with our TMOS E-FET designs, TMOS V is designed to withstand high energy in the avalanche and commutation modes. Designed for low voltage, high speed switching applications in power supplies, converters and power motor controls, these devices are particularly well suited for bridge circuits where diode speed and commutating safe operating areas are critical and offer additional safety margin against unexpected voltage transients.



MTP3055VL

Motorola Preferred Device

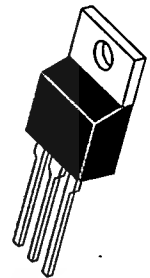
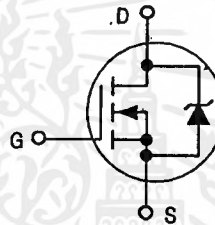
TMOS POWER FET
12 AMPERES
60 VOLTS
R_{DS(on)} = 0.18 OHM

New Features of TMOS V

- On-resistance Area Product about One-half that of Standard MOSFETs with New Low Voltage, Low R_{DS(on)} Technology
- Faster Switching than E-FET Predecessors

Features Common to TMOS V and TMOS E-FETS

- Avalanche Energy Specified
- I_{DSS} and V_{DS(on)} Specified at Elevated Temperature
- Static Parameters are the Same for both TMOS V and TMOS E-FET



CASE 221A-06, Style 5
TO-220AB

MAXIMUM RATINGS (T_C = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Drain-Source Voltage	V _{DSS}	60	Vdc
Drain-Gate Voltage (R _{GS} = 1.0 MΩ)	V _{DGR}	60	Vdc
Gate-Source Voltage — Continuous	V _{GS}	±15	Vdc
— Single Pulse (t _p ≤ 50 μs)	V _{GSM}	± 20	Vpk
Drain Current — Continuous @ 25°C	I _D	12	Adc
— Continuous @ 100°C	I _D	8.0	
— Single Pulse (t _p ≤ 10 μs)	I _{DM}	42	Apk
Total Power Dissipation @ 25°C	P _D	48	Watts
Derate above 25°C		0.32	W/°C
Operating and Storage Temperature Range	T _J , T _{stg}	-55 to 175	°C
Single Pulse Drain-to-Source Avalanche Energy — Starting T _J = 25°C (V _{DD} = 25 Vdc, V _{GS} = 5.0 Vdc, I _L = 12 Apk, L = 1.0 mH, R _G = 25 Ω)	E _{AS}	72	mJ
Thermal Resistance — Junction to Case	R _{θJC}	3.13	°C/W
— Junction to Ambient	R _{θJA}	62.5	
Maximum Lead Temperature for Soldering Purposes, 1/8" from case for 10 seconds	T _L	260	°C

Designer's Data for "Worst Case" Conditions — The Designer's Data Sheet permits the design of most circuits entirely from the information presented. SOA Limit curves — representing boundaries on device characteristics — are given to facilitate "worst case" design.

E-FET, Designer's and TMOS V are trademarks of Motorola, Inc. TMOS is a registered trademark of Motorola, Inc.

Preferred devices are Motorola recommended choices for future use and best overall value.

REV 2

MTP3055VL

ELECTRICAL CHARACTERISTICS (T_J = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit				
OFF CHARACTERISTICS									
Drain-Source Breakdown Voltage (V _{GS} = 0 Vdc, I _D = 250 μAdc) Temperature Coefficient (Positive)	V _{(BR)DSS}	60	— 62	—	Vdc mV/°C				
Zero Gate Voltage Drain Current (V _{DS} = 60 Vdc, V _{GS} = 0 Vdc) (V _{DS} = 60 Vdc, V _{GS} = 0 Vdc, T _J = 150°C)	I _{DSS}	—	—	10 100	μAdc				
Gate-Body Leakage Current (V _{GS} = ±15 Vdc, V _{DS} = 0)	I _{GSS}	—	—	100	nAdc				
ON CHARACTERISTICS (1)									
Gate Threshold Voltage (V _{DS} = V _{GS} , I _D = 250 μAdc) Temperature Coefficient (Negative)	V _{GS(th)}	1.0	1.6 3.0	2.0	Vdc mV/°C				
Static Drain-Source On-Resistance (V _{GS} = 5.0 Vdc, I _D = 6.0 Adc)	R _{DS(on)}	—	0.12	0.18	Ohm				
Drain-Source On-Voltage (V _{GS} = 5.0 Vdc) (I _D = 12 Adc) (I _D = 6.0 Adc, T _J = 150°C)	V _{DS(on)}	—	1.6	2.6 2.5	Vdc				
Forward Transconductance (V _{DS} = 8.0 Vdc, I _D = 6.0 Adc)	g _{FS}	5.0	8.8	—	mhos				
DYNAMIC CHARACTERISTICS									
Input Capacitance	C _{iss}	—	410	570	pF				
Output Capacitance						C _{oss}	—	114	160
Reverse Transfer Capacitance						C _{rss}	—	21	40
SWITCHING CHARACTERISTICS (2)									
Turn-On Delay Time	t _{d(on)}	—	9.0	20	ns				
Rise Time						t _r	—	85	190
Turn-Off Delay Time						t _{d(off)}	—	14	30
Fall Time						t _f	—	43	90
Gate Charge (See Figure 8)	Q _T	—	8.1	10	nC				
						Q ₁	—	1.8	—
						Q ₂	—	4.2	—
						Q ₃	—	3.8	—
SOURCE-DRAIN DIODE CHARACTERISTICS									
Forward On-Voltage (1)	V _{SD}	—	0.97 0.86	1.3	Vdc				
Reverse Recovery Time (See Figure 14)	t _{rr}	—	55.7	—	ns				
						t _a	—	37	—
						t _b	—	18.7	—
Reverse Recovery Stored Charge	Q _{RR}	—	0.116	—	μC				
INTERNAL PACKAGE INDUCTANCE									
Internal Drain Inductance (Measured from contact screw on tab to center of die) (Measured from the drain lead 0.25" from package to center of die)	L _D	—	3.5 4.5	—	nH				
Internal Source Inductance (Measured from the source lead 0.25" from package to source bond pad)	L _S	—	7.5	—	nH				

(1) Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2%.

(2) Switching characteristics are independent of operating junction temperature.

TYPICAL ELECTRICAL CHARACTERISTICS

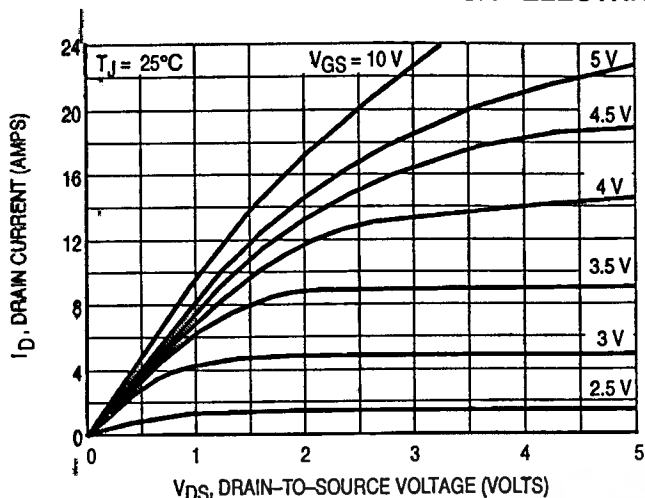


Figure 1. On-Region Characteristics

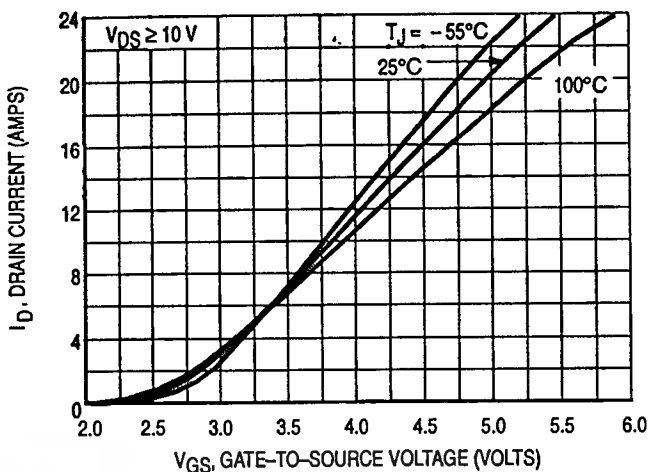


Figure 2. Transfer Characteristics

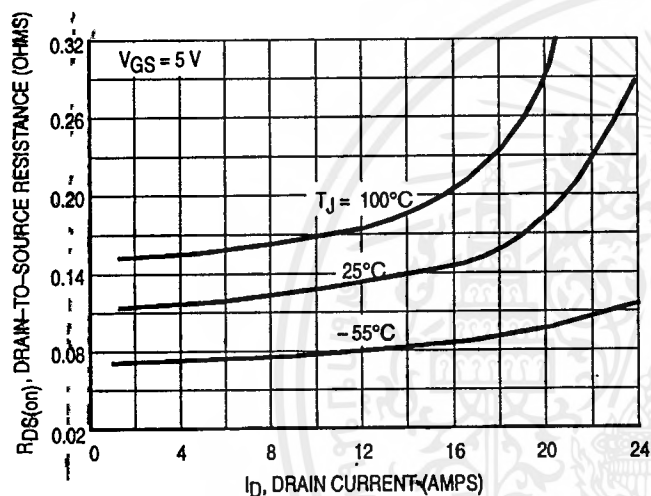


Figure 3. On-Resistance versus Drain Current and Temperature

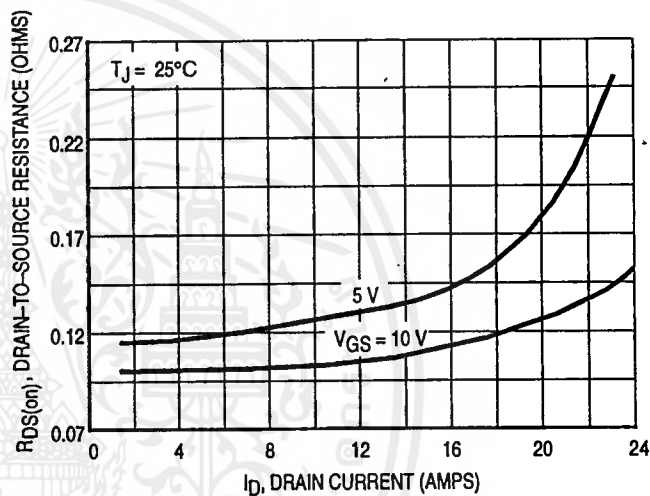


Figure 4. On-Resistance versus Drain Current and Gate Voltage

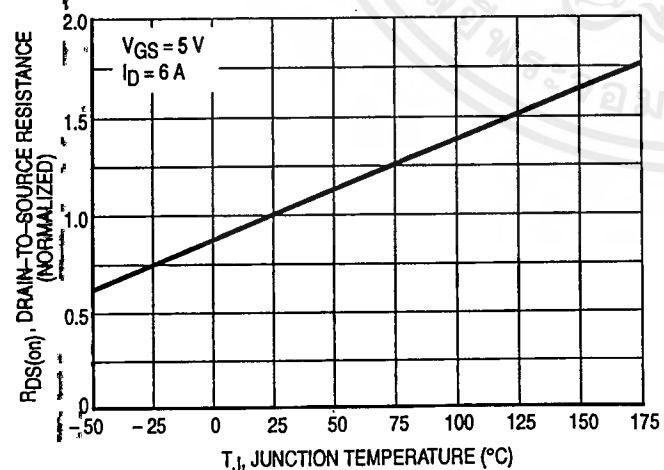


Figure 5. On-Resistance Variation with Temperature

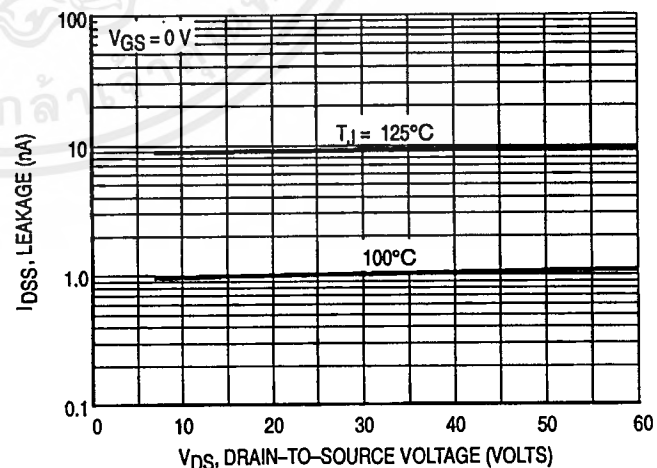


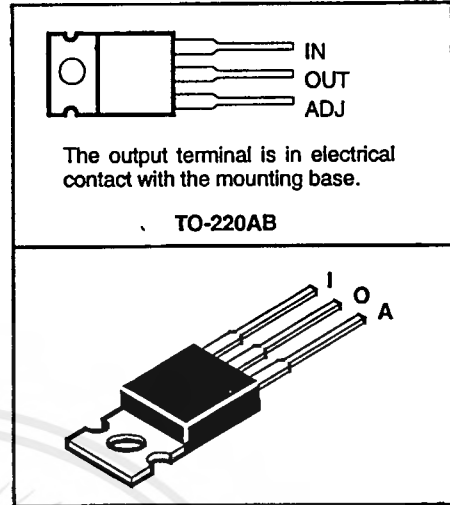
Figure 6. Drain-To-Source Leakage Current versus Voltage

TL783C, TL783Y HIGH-VOLTAGE ADJUSTABLE REGULATOR

SLVS036C – SEPTEMBER 1981 – REVISED APRIL 1997

- Output Adjustable From 1.25 V to 125 V When Used With an External Resistor Divider
- 700-mA Output Current
- Full Short-Circuit, Safe-Operating-Area, and Thermal Shutdown Protection
- 0.001%/V Typical Input Voltage Regulation
- 0.15% Typical Output Voltage Regulation
- 76-dB Typical Ripple Rejection
- Standard TO-220AB Package

KC PACKAGE
(TOP VIEW)



description

The TL783C is an adjustable three-terminal high-voltage regulator with an output range of 1.25 V to 125 V and a DMOS output transistor capable of sourcing more than 700 mA. It is designed for use in high-voltage applications where standard bipolar regulators cannot be used. Excellent performance specifications, superior to those of most bipolar regulators, are achieved through circuit design and advanced layout techniques.

As a state-of-the-art regulator, the TL783C combines standard bipolar circuitry with high-voltage double-diffused MOS transistors on one chip to yield a device capable of withstanding voltages far higher than standard bipolar integrated circuits. Because of its lack of secondary breakdown and thermal runaway characteristics usually associated with bipolar outputs, the TL783C maintains full overload protection while operating at up to 125 V from input to output. Other features of the device include current limiting, safe-operating-area (SOA) protection, and thermal shutdown. Even if ADJ is inadvertently disconnected, the protection circuitry remains functional.

Only two external resistors are required to program the output voltage. An input bypass capacitor is necessary only when the regulator is situated far from the input filter. An output capacitor, although not required, improves transient response and protection from instantaneous output short circuits. Excellent ripple rejection can be achieved without a bypass capacitor at the adjustment terminal.

AVAILABLE OPTIONS

T _J	PACKAGED DEVICE	CHIP FORM (Y)
	HEAT-SINK MOUNTED (3-PIN) (KC)	
0°C to 125°C	TL783CKC	TL783Y

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1997, Texas Instruments Incorporated

 **TEXAS
INSTRUMENTS**

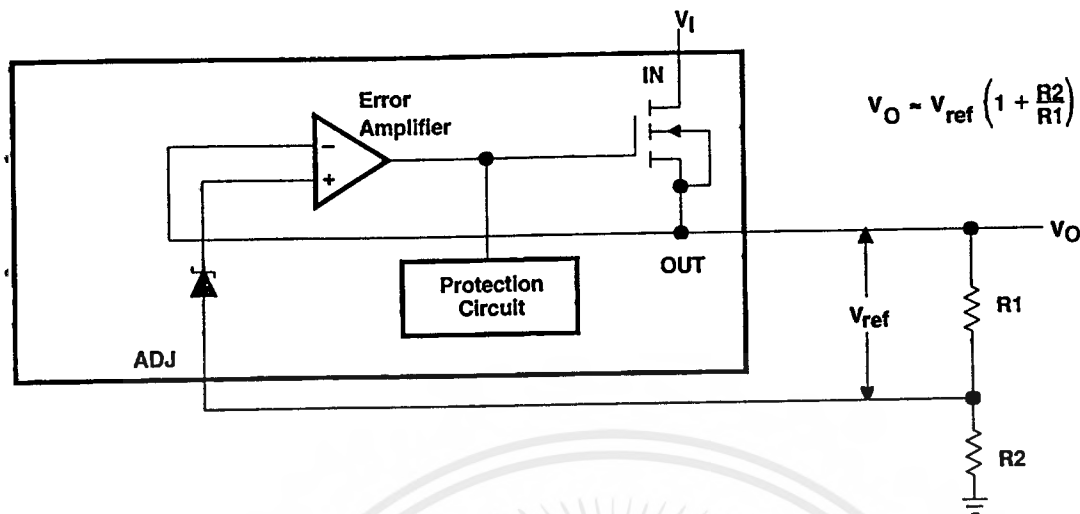
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีพิมพ์ลงนิตยสารหรือสิ่งพิมพ์ใดๆ ของเอกสารทุกครั้งที่มีการนำไปใช้

TL783C, TL783Y HIGH-VOLTAGE ADJUSTABLE REGULATOR

SLVS036C – SEPTEMBER 1981 – REVISED APRIL 1997

functional block diagram



absolute maximum ratings over operating temperature range (unless otherwise noted)[†]

Input-to-output differential voltage, $V_I - V_O$	125 V
Continuous total power dissipation at (or below) $T_A = 25^\circ\text{C}$ (see Note 1)	2 W
Continuous total power dissipation at (or below) $T_C = 70^\circ\text{C}$ (see Note 1)	20 W
Operating free-air, T_A , case, T_C , or virtual junction, T_J , temperature range	0°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

[†] Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: For operation above $T_A = 25^\circ\text{C}$ or $T_C = 70^\circ\text{C}$, refer to Figures 1 and 2, respectively. To avoid exceeding the design maximum virtual junction temperature, these ratings should not be exceeded. Due to variations in individual device electrical characteristics and thermal resistance, the built-in thermal overload protection may be activated at power levels slightly above or below the rated dissipation

**FREE-AIR TEMPERATURE
DISSIPATION DERATING CURVE**

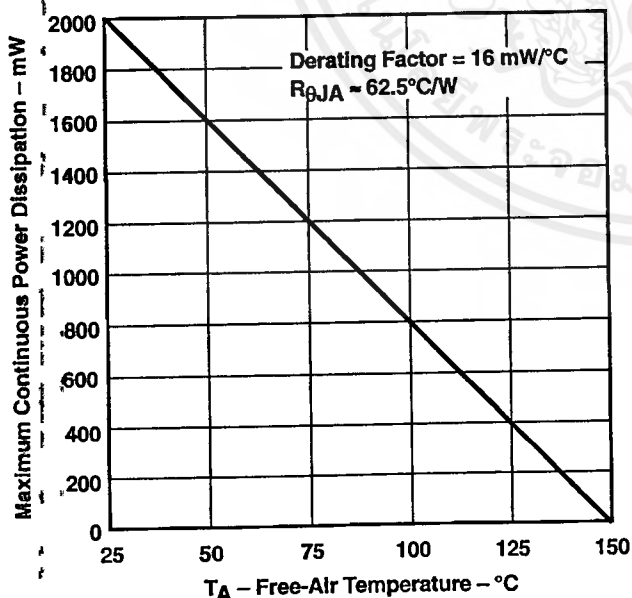


Figure 1

**CASE TEMPERATURE
DISSIPATION DERATING CURVE**

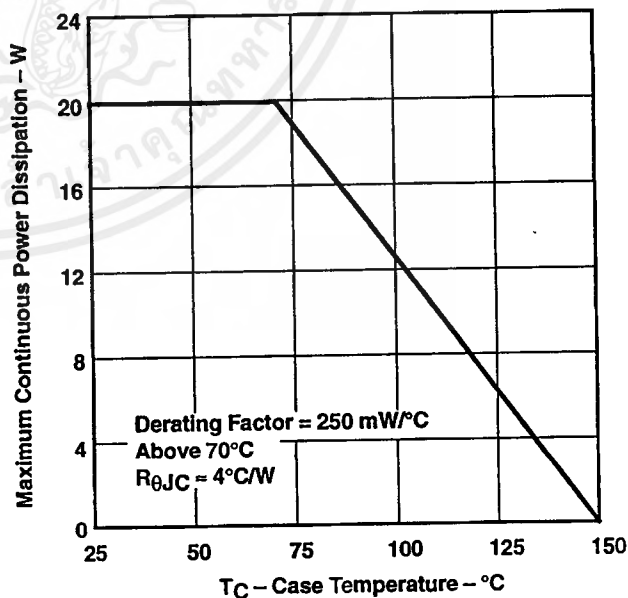


Figure 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายใน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลของเอกสารทุกครั้งที่มีการนำไปใช้

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

TL783C, TL783Y HIGH-VOLTAGE ADJUSTABLE REGULATOR

SLV6938C - SEPTEMBER 1981 - REVISED APRIL 1997

recommended operating conditions

	MIN	MAX	UNIT
Input-to-output voltage differential, $V_I - V_O$		125	V
Output current, I_O	15	700	mA
Operating virtual junction temperature, T_J	0	125	°C

electrical characteristics at $V_I - V_O = 25$ V, $I_O = 0.5$ A, $T_J = 0^\circ\text{C}$ to 125°C (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	TL783C			UNIT
		MIN	TYP	MAX	
Input voltage regulation‡	$V_I - V_O = 20$ V to 125 V, $P \leq$ rated dissipation	$T_J = 25^\circ\text{C}$	0.001	0.01	% / V
		$T_J = 0^\circ\text{C}$ to 125°C	0.004	0.02	
Ripple rejection	$\Delta V_I(\text{pp}) = 10$ V, $V_O = 10$ V, $f = 120$ Hz	66	76		dB
Output voltage regulation	$I_O = 15$ mA to 700 mA, $T_J = 25^\circ\text{C}$	$V_O \leq 5$ V	7.5	25	mV
		$V_O \geq 5$ V	0.15%	0.5%	
	$I_O = 15$ mA to 700 mA, $P \leq$ rated dissipation	$V_O \leq 5$ V	20	70	mV
		$V_O \geq 5$ V	0.3%	1.5%	
Output voltage change with temperature		0.4%			
Output voltage long-term drift	1000 hours at $T_J = 125^\circ\text{C}$, $V_I - V_O = 125$ V, See Note 2	0.2%			
Output noise voltage	$f = 10$ Hz to 10 kHz, $T_J = 25^\circ\text{C}$	0.003%			
Minimum output current to maintain regulation	$V_I - V_O = 125$ V			15	mA
Peak output current	$V_I - V_O = 25$ V, $t = 1$ ms		1100		mA
	$V_I - V_O = 15$ V, $t = 30$ ms		715		
	$V_I - V_O = 25$ V, $t = 30$ ms	700	900		
	$V_I - V_O = 125$ V, $t = 30$ ms	100	250		
Input current at ADJ		83	110		μA
Change in input current at ADJ	$V_I - V_O = 15$ V to 125 V, $I_O = 15$ mA to 700 mA, $P \leq$ rated dissipation		0.5	5	μA
Reference voltage (OUT to ADJ)	$V_I - V_O = 10$ V to 125 V, $I_O = 15$ mA to 700 mA, $P \leq$ rated dissipation, See Note 3	1.2	1.27	1.3	V

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately.

‡ Input voltage regulation is expressed here as the percentage change in output voltage per 1-V change at the input.

- NOTES:
- Since long-term drift cannot be measured on the individual devices prior to shipment, this specification is not intended to be a guarantee or warranty. It is an engineering estimate of the average drift to be expected from lot to lot.
 - Due to the dropout voltage and output current-limiting characteristics of this device, output current is limited to less than 700 mA at input-to-output voltage differentials of less than 25 V.



TEXAS
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีพิมพ์ลงนิตยสารหรือเผยแพร่ต่อสาธารณชนโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGN CONSIDERATIONS

protection circuitry (continued)

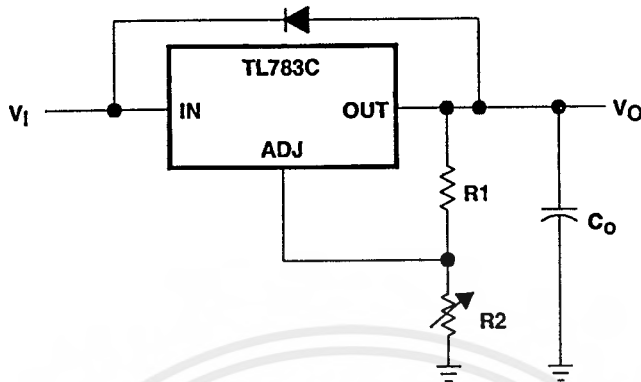


Figure 17. Regulator With Protective Diode

load regulation

The current set resistor (R1) should be located close to the regulator output terminal rather than near the load. This eliminates long line drops from being amplified through the action of R1 and R2 to degrade load regulation. To provide remote ground sensing, R2 should be near the load ground.

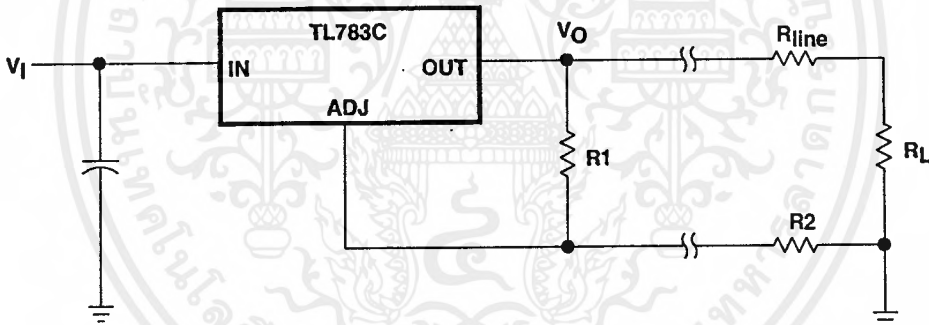


Figure 18. Regulator With Current-Set Resistor



TEXAS INSTRUMENTS

RFG50N06, RFP50N06, RF1S50N06, RF1S50N06SM

50A, 60V, 0.022 Ohm,
N-Channel Power MOSFETs

October 1997

Features

- 50A, 60V
- $r_{DS(ON)} = 0.022\Omega$
- Temperature Compensating PSPICE Model
- Peak Current vs Pulse Width Curve
- UIS Rating Curve
- 175°C Operating Temperature
- Related Literature
 - TB334, "Guidelines for Soldering Surface Mount Components to PC Boards"

Description

These N-Channel power MOSFETs are manufactured using the MegaFET process. This process, which uses feature sizes approaching those of LSI integrated circuits gives optimum utilization of silicon, resulting in outstanding performance. They were designed for use in applications such as switching regulators, switching converters, motor drivers, and relay drivers. These transistors can be operated directly from integrated circuits.

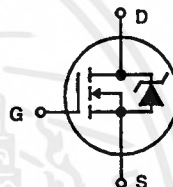
Formerly developmental type TA49018.

Ordering Information

PART NUMBER	PACKAGE	BRAND
RFG50N06	TO-247	RFG50N06
RFP50N06	TO-220AB	RFP50N06
RF1S50N06	TO-262AA	F1S50N06
RF1S50N06SM	TO-263AB	F1S50N06

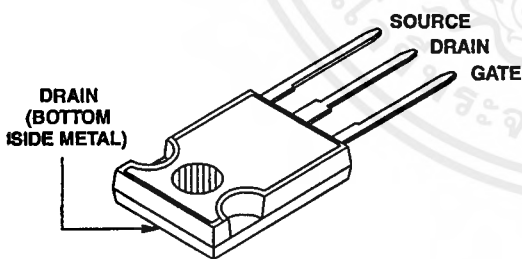
NOTE: When ordering, use the entire part number. Add the suffix, 9A, to obtain the TO-263AB variant in tape and reel, i.e. RF1S50N06SM9A.

Symbol

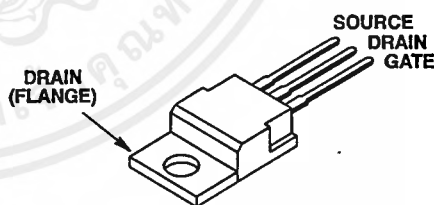


Packaging

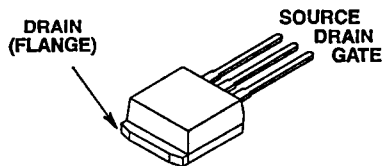
JEDEC STYLE TO-247



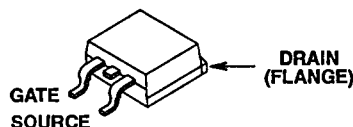
JEDEC TO-220AB



JEDEC TO-262AA



JEDEC TO-263AB



RFG50N06, RFP50N06, RF1S50N06, RF1S50N06SM

Absolute Maximum Ratings $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

	RFG50N06, RFP50N06 RF1S50N06, RF1S50N06SM	UNITS
Drain to Source Voltage (Note 1)	60	V
Drain to Gate Voltage ($R_{GS} = 20k\Omega$) (Note 1)	60	V
Gate to Source Voltage	± 20	V
Continuous Drain Current (Figure 2)	50	A
Pulsed Drain Current	(Figure 5)	
Pulsed Avalanche Rating	(Figure 6, 14, 15)	
Power Dissipation	131	W
Linear Derating Factor	0.877	$W/^\circ\text{C}$
Operating and Storage Temperature	-55 to 175	$^\circ\text{C}$
Maximum Temperature for Soldering		
Leads at 0.063in (1.6mm) from Case for 10s	300	$^\circ\text{C}$
Package Body for 10s, see Techbrief 334	260	$^\circ\text{C}$

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

1. $T_J = 25^\circ\text{C}$ to 150°C .

Electrical Specifications $T_C = 25^\circ\text{C}$, Unless Otherwise Specified

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS	
Drain to Source Breakdown Voltage	BV_{DSS}	$I_D = 250\mu\text{A}$, $V_{GS} = 0\text{V}$ (Figure 11)	60	-	-	V	
Gate to Source Threshold Voltage	$V_{GS(TH)}$	$V_{GS} = V_{DS}$, $I_D = 250\mu\text{A}$ (Figure 10)	2	-	4	V	
Zero Gate Voltage Drain Current	I_{DSS}	$V_{DS} = 60\text{V}$, $V_{GS} = 0\text{V}$	$T_C = 25^\circ\text{C}$	-	-	1	μA
			$T_C = 150^\circ\text{C}$	-	-	50	μA
Gate to Source Leakage Current	I_{GSS}	$V_{GS} = \pm 20\text{V}$	-	-	± 100	nA	
Drain to Source On Resistance	$r_{DS(ON)}$	$I_D = 50\text{A}$, $V_{GS} = 10\text{V}$ (Figures 9)	-	-	0.022	Ω	
Turn-On Time	t_{ON}	$V_{DD} = 30\text{V}$, $I_D = 50\text{A}$ $R_L = 0.6\Omega$, $V_{GS} = 10\text{V}$ $R_{GS} = 3.6\Omega$ (Figures 16, 17)	-	-	95	ns	
Turn-On Delay Time	$t_{d(ON)}$		-	12	-	ns	
Rise Time	t_r		-	55	-	ns	
Turn-Off Delay Time	$t_{d(OFF)}$		-	37	-	ns	
Fall Time	t_f		-	13	-	ns	
Turn-Off Time	t_{OFF}		-	-	-	75	ns
Total Gate Charge	$Q_g(TOT)$	$V_{GS} = 0$ to 20V	-	125	150	nC	
Gate Charge at 10V	$Q_g(10)$	$V_{GS} = 0$ to 10V					
Threshold Gate Charge	$Q_g(TH)$	$V_{GS} = 0$ to 2V					
Input Capacitance	C_{ISS}	$V_{DS} = 25\text{V}$, $V_{GS} = 0\text{V}$ $f = 1\text{MHz}$ (Figure 12)	-	2020	-	pF	
Output Capacitance	C_{OSS}		-	600	-	pF	
Reverse Transfer Capacitance	C_{RSS}		-	200	-	pF	
Thermal Resistance Junction to Case	$R_{\theta JC}$	(Figure 3)	-	-	1.14	$^\circ\text{C/W}$	
Thermal Resistance Junction to Ambient	$R_{\theta JA}$	TO-247	-	-	30	$^\circ\text{C/W}$	
		TO-220, TO-262, TO-263	-	-	62	$^\circ\text{C/W}$	

Source to Drain Diode Specifications

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNITS
Source to Drain Diode Voltage	V_{SD}	$I_{SD} = 50\text{A}$	-	-	1.5	V
Reverse Recovery Time	t_{rr}	$I_{SD} = 50\text{A}$, $dI_{SD}/dt = 100\text{A}/\mu\text{s}$	-	-	125	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา 3-51 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Curves Unless Otherwise Specified

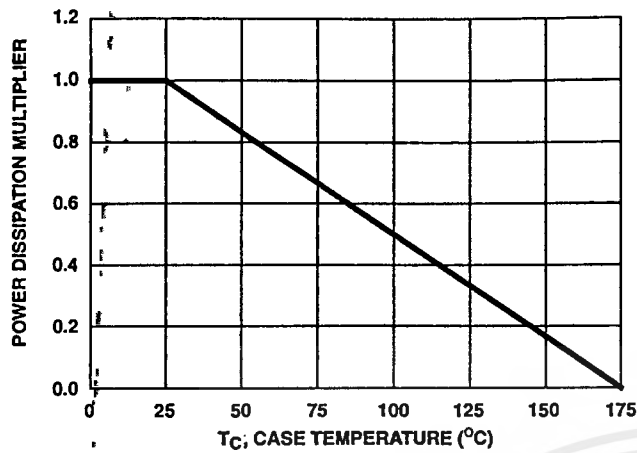


FIGURE 1. NORMALIZED POWER DISSIPATION vs CASE TEMPERATURE

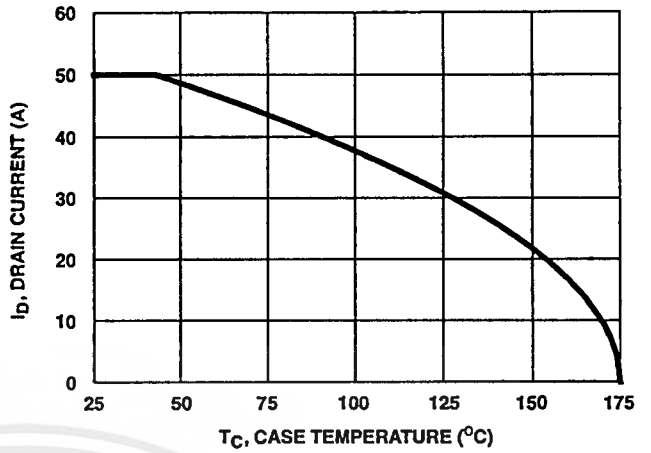


FIGURE 2. MAXIMUM CONTINUOUS DRAIN CURRENT vs CASE TEMPERATURE

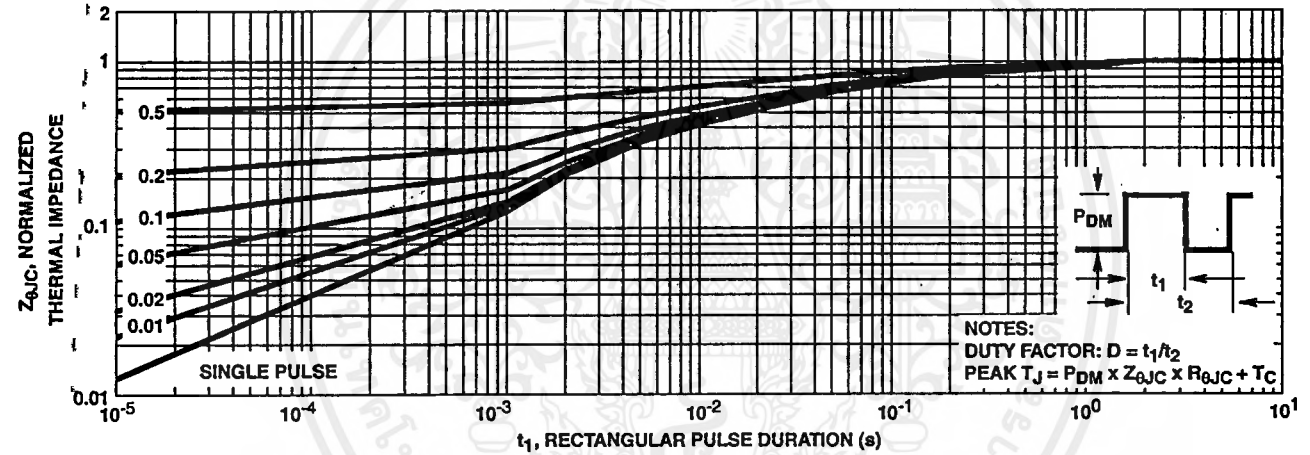


FIGURE 3. NORMALIZED MAXIMUM TRANSIENT THERMAL IMPEDANCE

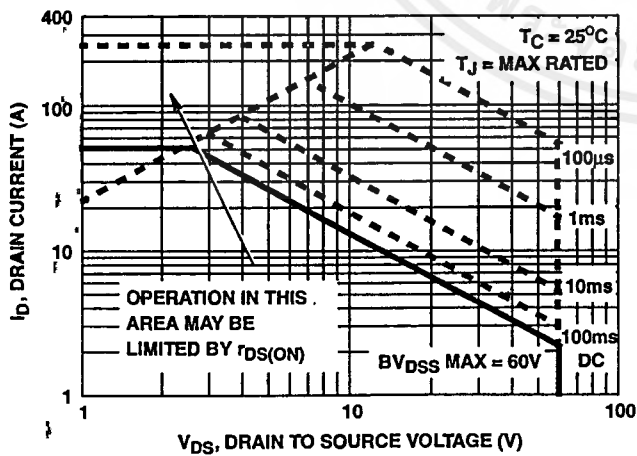


FIGURE 4. FORWARD BIAS SAFE OPERATING AREA

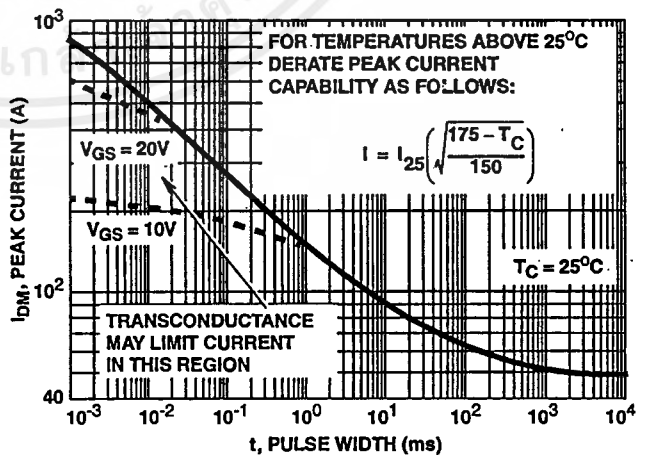
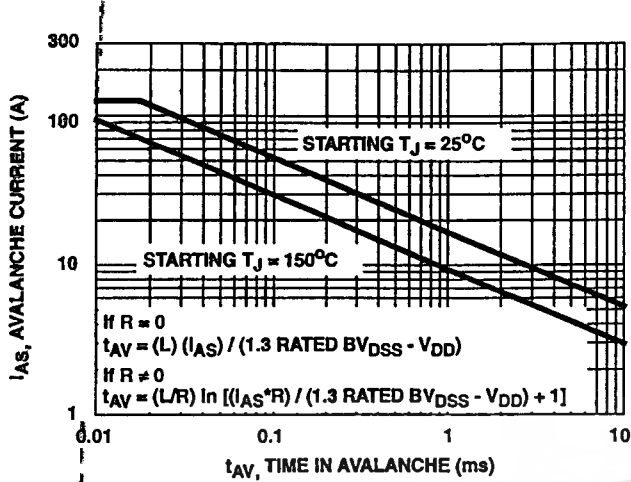


FIGURE 5. PEAK CURRENT CAPABILITY

Typical Performance Curves Unless Otherwise Specified (Continued)



NOTE: Refer to Harris Application Notes 9321 and 9322.

FIGURE 6. UNCLAMPED INDUCTIVE SWITCHING CAPABILITY

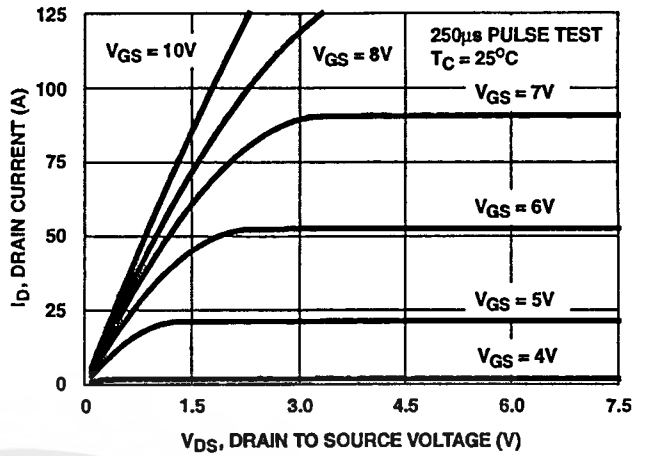


FIGURE 7. SATURATION CHARACTERISTICS

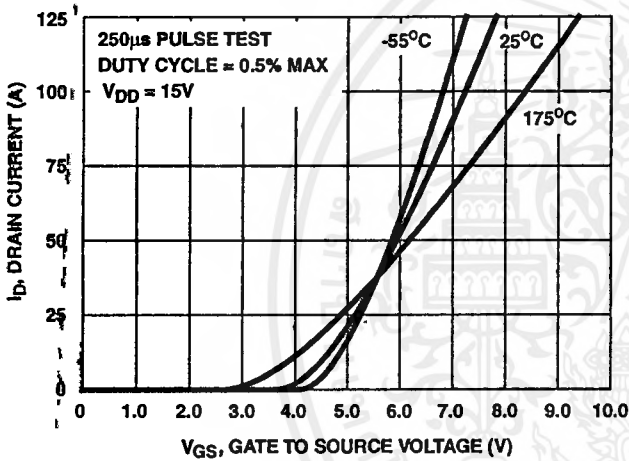


FIGURE 8. TRANSFER CHARACTERISTICS

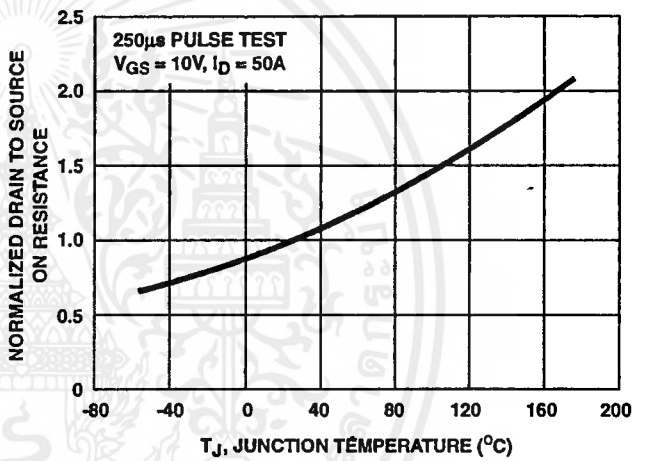


FIGURE 9. NORMALIZED DRAIN TO SOURCE ON RESISTANCE vs JUNCTION TEMPERATURE

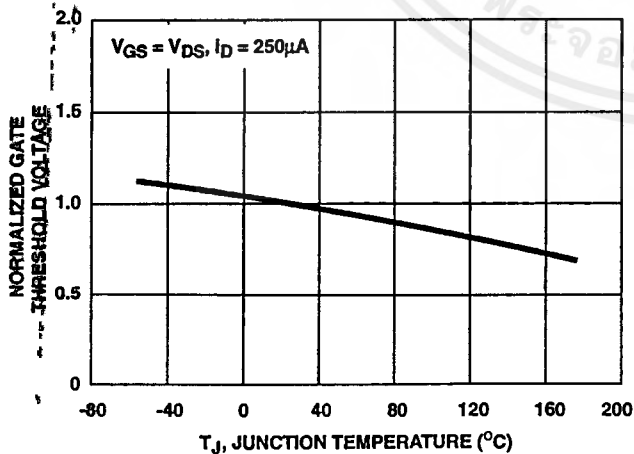


FIGURE 10. NORMALIZED GATE THRESHOLD VOLTAGE vs JUNCTION TEMPERATURE

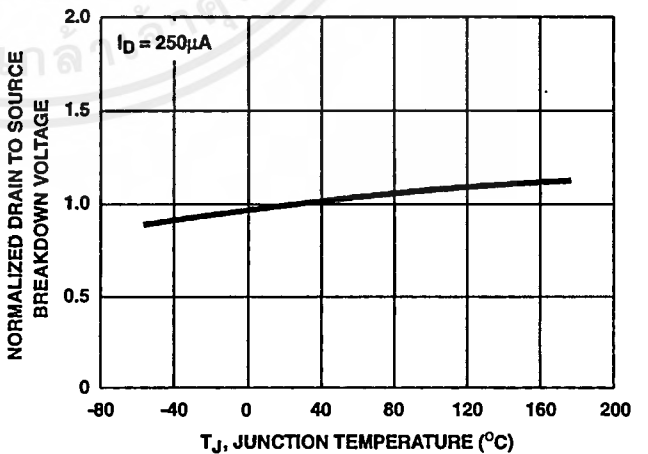


FIGURE 11. NORMALIZED DRAIN TO SOURCE BREAKDOWN VOLTAGE vs JUNCTION TEMPERATURE



MICROCHIP

PIC16C5X

EPROM/ROM-Based 8-Bit CMOS Microcontroller Series

Devices Included in this Data Sheet:

- PIC16C52
- PIC16C54s
- PIC16CR54s
- PIC16C55s
- PIC16C56s
- PIC16CR56s
- PIC16C57s
- PIC16CR57s
- PIC16C58s
- PIC16CR58s

Note: The letter "s" used following the part numbers throughout this document indicate plural, meaning there is more than one part variety for the indicated device.

High-Performance RISC CPU:

- Only 33 single word instructions to learn
- All instructions are single cycle (200 ns) except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle

Device	Pins	I/O	EPROM/ROM	RAM
PIC16C52	18	12	384	25
PIC16C54	18	12	512	25
PIC16C54A	18	12	512	25
PIC16C54B	18	12	512	25
PIC16C54C	18	12	512	25
PIC16CR54A	18	12	512	25
PIC16CR54B	18	12	512	25
PIC16CR54C	18	12	512	25
PIC16C55	28	20	512	24
PIC16C55A	28	20	512	24
PIC16C56	18	12	1K	25
PIC16C56A	18	12	1K	25
PIC16CR56A	18	12	1K	25
PIC16C57	28	20	2K	72
PIC16C57C	28	20	2K	72
PIC16CR57B	28	20	2K	72
PIC16CR57C	28	20	2K	72
PIC16C58A	18	12	2K	73
PIC16C58B	18	12	2K	73
PIC16CR58A	18	12	2K	73
PIC16CR58B	18	12	2K	73

- 12-bit wide instructions
- 8-bit wide data path
- Seven or eight special function hardware registers
- Two-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

Peripheral Features:

- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- Power-On Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options:
 - RC: Low-cost RC oscillator
 - XT: Standard crystal/resonator
 - HS: High-speed crystal/resonator
 - LP: Power saving, low-frequency crystal

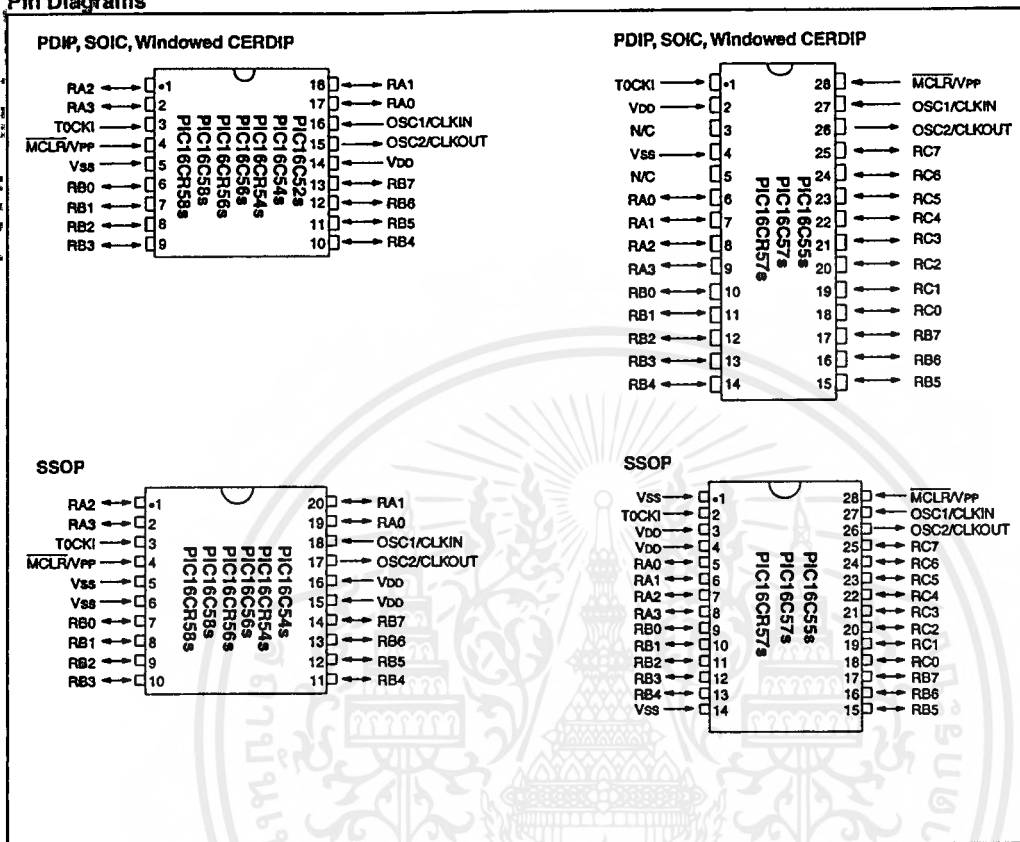
CMOS Technology:

- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide-operating voltage and temperature range:
 - EPROM Commercial/Industrial 2.0V to 6.25V
 - ROM Commercial/Industrial 2.0V to 6.25V
 - EPROM Extended 2.5V to 6.0V
 - ROM Extended 2.5V to 6.0V
- Low-power consumption
 - < 2 mA typical @ 5V, 4 MHz
 - 15 μ A typical @ 3V, 32 kHz
 - < 0.6 μ A typical standby current (with WDT disabled) @ 3V, 0°C to 70°C

Note: In this document, figure and table titles refer to all varieties of the part number indicated, (i.e., The title "Figure 14-1: Load Conditions - PIC16C54A", also refers to PIC16LC54A and PIC16LV54A parts).

PIC16C5X

Pin Diagrams



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16C5X

Device Differences

Device	Voltage Range	Oscillator Selection (Program)	Oscillator	Process Technology (Microns)	ROM Equivalent	MCLR Filter
PIC16C52	3.0-6.25	User	See Note 1	0.9	—	No
PIC16C54	2.5-6.25	Factory	See Note 1	1.2	PIC16CR54A	No
PIC16C54A	2.0-6.25	User	See Note 1	0.9	—	No
PIC16C54B	2.5-5.5	User	See Note 1	0.7	PIC16CR54B	Yes
PIC16C54C	2.5-5.5	User	See Note 1	0.7	PIC16CR54C	Yes
PIC16C55	2.5-6.25	Factory	See Note 1	1.7	—	No
PIC16C55A	2.5-5.5	User	See Note 1	0.7	—	Yes
PIC16C56	2.5-6.25	Factory	See Note 1	1.7	—	No
PIC16C56A	2.5-5.5	User	See Note 1	0.7	PIC16CR56A	Yes
PIC16C57	2.5-6.25	Factory	See Note 1	1.2	—	No
PIC16C57C	2.5-5.5	User	See Note 1	0.7	PIC16CR57C	Yes
PIC16C58A	2.0-6.25	User	See Note 1	0.9	PIC16CR58A	No ⁽²⁾
PIC16C58B	2.5-5.5	User	See Note 1	0.7	PIC16CR58B	Yes
PIC16CR54A	2.5-6.25	Factory	See Note 1	1.2	N/A	Yes
PIC16CR54B	2.5-5.5	Factory	See Note 1	0.7	N/A	Yes
PIC16CR54C	2.5-5.5	Factory	See Note 1	0.7	N/A	Yes
PIC16CR56A	2.5-5.5	Factory	See Note 1	0.7	N/A	Yes
PIC16CR57B	2.5-6.25	Factory	See Note 1	0.9	N/A	Yes
PIC16CR57C	2.5-5.5	Factory	See Note 1	0.7	N/A	Yes
PIC16CR58A	2.5-6.25	Factory	See Note 1	0.9	N/A	Yes
PIC16CR58B	2.5-5.5	Factory	See Note 1	0.7	N/A	Yes

Note 1: If you change from this device to another device, please verify oscillator characteristics in your application.

Note 2: In PIC16LV58A, MCLR Filter = Yes