



การควบคุมการเคลื่อนที่ของเครน
CRANE MOVEMENT CONTROL

โดย

1. นาย ปวิณ โรจนวงศ์
2. นาย อาณัติ บูรานนท์
3. นาย อุกฤษฏ์ ผิวละออ

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. จงกล งามวิวิทย์

อาจารย์ สุมิตร พนาอุดมทรัพย์

วัน เดือน ปี..... 11. ๑๑. 2541
เลขทะเบียน..... 038879
เลขเรียกหนังสือ..... ๓๐๑๕๓ ๒๓๑๕๓

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้

038879

ปริญญาโทปีการศึกษา 2540

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมการเคลื่อนที่ของเครน
CRANE MOVEMENT CONTROL

ผู้จัดทำ นาย ปวิณ โรจนวงศ์ 37014250
นาย อาณัติ บุรานนท์ 37014584
นาย อุกฤษฏ์ ศิวละอ 37014595

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

..... อาจารย์ที่ปรึกษา
(อาจารย์ สุมิตร พนาอุดมทรัพย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเคลื่อนที่ของเครน

CRANE MOVEMENT CONTROL

โดย

นาย ปวิณ โรจนวงศ์ 37014250
 นาย อาณัติ นุรานนท์ 37014584
 นาย อุกฤษฏ์ ฬิวระออบ 37014595

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. จงกล งามวิวิทย์

อาจารย์ สุมิตร พนาอุคมทรัพย์

บทคัดย่อ

โครงการควบคุมการเคลื่อนที่ของเครนในปีนี้มีจุดมุ่งหมายในการพัฒนาโครงการที่มีอยู่เดิมให้ดียิ่งขึ้น โดยปรับปรุงในส่วนของการควบคุมระบบและพัฒนาโปรแกรมให้มีความสะดวกในการใช้งาน และเหมาะในการพัฒนาโครงการนี้ต่อไป โดยใช้วิธี LQR เพื่อออกแบบตัวควบคุมให้ควบคุมรถเครนเคลื่อนที่ไปยังจุดที่ต้องการ ในขณะที่เดียวกันก็สามารถควบคุมให้ภาระที่แขวนอยู่แกว่งน้อยที่สุด

ABSTRACT

The project of crane movement control for this year has improved the hardware and software parts for more convenience in controlling the crane system. The LQR method is utilized for designing the controller in order to control the desired position of the crane with small swinging of the load.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะได้รับความเมตตาจาก

รองศาสตราจารย์ ดร. จงกล งามวิวิทย์ อาจารย์ สุมิตร พนาอุคมทรัพย์ อาจารย์ที่ปรึกษา
รองศาสตราจารย์ ดร. โยธิน เปรมปรานีรัชต์ ดร. นนทวัฒน์ จุลเดชะ ดร. วันชัย ธีรรูจา ดร. สุธี
ผู้เจริญชนะชัย รองศาสตราจารย์ วิพันธ์ ปรีชาพานิช รองศาสตราจารย์ สุเชียร เกียรติสุนทร
อาจารย์ พรสุข เทศเจริญ อาจารย์ เกียรติศักดิ์ คมวัชระ อาจารย์ เกียรติวรรณ ทรงศักดิ์ อาจารย์
กงศักดิ์ อนันตธีรวิรัตน์ อาจารย์ เทพจิตร เขยโกคา อาจารย์ สว่าง เลิศธิรสุนทร และขอขอบคุณที่
อมรเทพ ที่ให้ความกรุณาदानอุปการณ์ตลอดมา ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ
โอกาสนี้ด้วย และยังขอขอบคุณคณาจารย์ต่างๆ ที่ได้ประสิทธิประสาทวิชาความรู้แก่ผู้จัดทำ

ขอขอบคุณ บริษัท เนชั่นเนล เซมิคอนดักเตอร์ จำกัด ที่ได้ให้ความอนุเคราะห์อุปการณ์และ
ข้อมูลที่เป็นประโยชน์

ขอกราบขอบพระคุณบิดามารดาของผู้จัดทำที่ได้อุปการะและยังเป็นผู้ให้กำลังใจแก่ผู้จัดทำ
ตลอดมา และต้องขอขอบคุณเพื่อนๆ ที่คอยให้ความช่วยเหลือและเป็นกำลังใจแก่ผู้จัดทำ

ผู้จัดทำ

นาย ปวิณ โรจนวงศ์

นาย อาณัติ บุรานนท์

นาย อุกฤษฏ์ ผิวระอ

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	II
สารบัญ	III
สารบัญภาพ	V
บทที่ 1 บทนำ	1
1.1 ประวัติการควบคุมเครน	1
1.2 จุดประสงค์ของการศึกษาและการควบคุมการเคลื่อนที่ของเครน	2
1.3 เนื้อหาที่จะกล่าวในปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 หลักการในการควบคุม และบล็อกไดอะแกรมของระบบ	3
2.2 โครงสร้างและส่วนประกอบของแบบจำลองเครน	4
2.3 ชุดควบคุมและเซ็นเซอร์	6
บทที่ 3 วงจรโดยรวมของระบบ	11
3.1 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์	11
3.2 วงจรชุดควบคุมความเร็ว	14
บทที่ 4 การหาแบบจำลองทางคณิตศาสตร์ของระบบ	20
4.1 แบบจำลองทางคณิตศาสตร์ของภาระ	20
4.2 การหาแบบจำลองทางคณิตศาสตร์ของตัวรถเครน	22
4.3 การหาแบบจำลองทางคณิตศาสตร์รวมทั้งระบบในรูปแบบการสแตกในระบบเวลาต่อเนื่อง	24
4.4 การแปลงสมการสแตกในระบบเวลาต่อเนื่องให้อยู่ในรูปแบบการสแตกในระบบเวลาดิสครีต	25
บทที่ 5 การควบคุมมุมการแกว่งและตำแหน่งของเครน	27
5.1 การควบคุมออฟติมัล	27
5.2 การนำระบบการควบคุมออฟติมัลแบบเรกดูเลเตอร์เชิงเส้นมาใช้ในการควบคุมการเคลื่อนที่ของเครน	31
บทที่ 6 ผลการทดลอง	34
6.1 การทดลองที่ความยาวเชือก 41.25 เซนติเมตร	34
6.2 การทดลองที่ความยาวเชือก 68.85 เซนติเมตร	49

เอกสารนี้ 6.3 การทดลองที่ความยาวเชือก 89.75 เซนติเมตร เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 64 การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 สรุปและวิจารณ์	79
หนังสืออ้างอิง	81
ภาคผนวก	82
ภาคผนวก ก กล้องชุดควบคุมเครน	83
ภาคผนวก ข โปรแกรม MATLAB ที่ใช้ในการคำนวณ	89
ภาคผนวก ค โปรแกรมควบคุมการเคลื่อนที่ของเครน	94
ภาคผนวก ง Source Code	98
ภาคผนวก จ คู่มือ Line Driver และ Receiver	129



สารบัญภาพ

	หน้า
รูปที่ 1.1 แสดงการใช้เทรนยกของในโรงงานอุตสาหกรรม	1
รูปที่ 2.1 แสดงบล็อกไดอะแกรมโดยรวมของระบบการควบคุมการเคลื่อนที่ของเทรน	3
รูปที่ 2.2 ระบบการควบคุมการเคลื่อนที่ของเทรน	4
รูปที่ 2.3 แสดงโครงสร้างและส่วนประกอบของแบบจำลองเทรน	4
รูปที่ 2.4 ก. แสดงการส่งกำลังแบบที่มีการทดรอบของไทม์มิ่งพูลเลย์	5
รูปที่ 2.4 ข. แสดงการส่งกำลังแบบที่ไม่มีการทดรอบของไทม์มิ่งพูลเลย์	5
รูปที่ 2.4 ค. ชุดส่งกำลังของระบบเทรน	6
รูปที่ 2.5 รูปแสดงบล็อกไดอะแกรมของชุดควบคุมการเคลื่อนที่ของเทรน	6
รูปที่ 2.6 วงจรตรวจวัดค่ามุม	8
รูปที่ 2.7 เซ็นเซอร์ต่างๆบนระบบเทรน	9
รูปที่ 2.8 แสดงตัวรถเทรน	10
รูปที่ 2.9 แสดงรถเทรนและภาระ	10
รูปที่ 3.1 แสดงตัวอย่างการเลือกตำแหน่งของคิปลิวิตซ์	11
รูปที่ 3.2 แสดงวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์	12
รูปที่ 3.3 แสดงการทำงานโดยรวมของวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์	13
รูปที่ 3.4 แสดงแผ่นวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์ และการ์ด PCL-714	14
รูปที่ 3.5 วงจรกำเนิดสัญญาณการขับมอเตอร์โดยใช้ LM555	15
รูปที่ 3.6 วงจรแปลงสัญญาณการขับมอเตอร์จาก LM629	16
รูปที่ 3.7 วงจรเลือกโหมดการทำงาน	17
รูปที่ 3.8 วงจรขับมอเตอร์ในแนวแกนนอน	18
รูปที่ 3.9 วงจรขับมอเตอร์ในแนวแกนตั้ง	19
รูปที่ 3.10 วงจรป้องกันการชน	19
รูปที่ 4.1 แสดงแบบจำลองของตัวเทรน	20
รูปที่ 4.2 รูปแสดงแบบจำลองของภาระ	21
รูปที่ 4.3 กราฟทดสอบ Step Response	22
รูปที่ 4.4 กราฟ Step Response	23
รูปที่ 5.1 แสดงบล็อกไดอะแกรมของระบบเปิด	28
รูปที่ 5.2 แสดงบล็อกไดอะแกรมของระบบเมื่อป้อนกลับด้วย $u_k = -Kx_k$	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ขออนุญาต

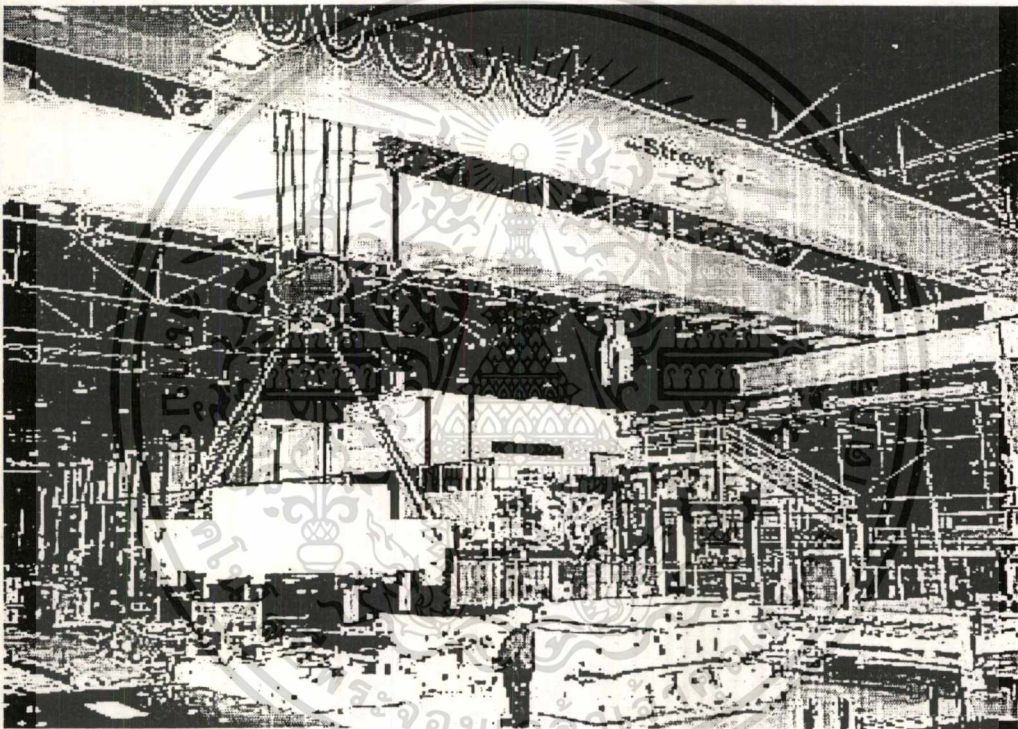
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.3	แสดงผลตอบสนองของมุมที่ได้จากการ Simulate	32
รูปที่ 5.4	แสดงผลตอบสนองของมุมที่ได้จากการทดลอง	32
รูปที่ 5.5	แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate	33
รูปที่ 5.6	แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง	33
รูปที่ 6.1 ถึง 6.20	แสดงผลการทดลองที่ความยาวเชือก 41.25 เซนติเมตร	34
รูปที่ 6.21 ถึง 6.40	แสดงผลการทดลองที่ความยาวเชือก 68.85 เซนติเมตร	49
รูปที่ 6.41 ถึง 6.60	แสดงผลการทดลองที่ความยาวเชือก 89.75 เซนติเมตร	64
รูปที่ ก.1	แสดงตำแหน่งของวงจรภายในกล่องควบคุม	84
รูปที่ ก.2	แสดงวงจรหลักภายในกล่องควบคุม	85
รูปที่ ก.3	แสดงตำแหน่งของอุปกรณ์บนวงจรหลัก	85
รูปที่ ก.4	แสดงตำแหน่งของวงจรตรวจวัดค่ามุม	86
รูปที่ ก.5	แสดงตำแหน่งของอุปกรณ์บนวงจรขับเคลื่อนมอเตอร์	87
รูปที่ ก.6	วงจรแสดงผลด้วย LED	88
รูปที่ ค.1	Flow Chart ของโปรแกรมการควบคุมการเคลื่อนที่ของเครน	95
รูปที่ ค.2	ลักษณะเมนูของ โปรแกรม CMC3.EXE	95
รูปที่ ค.3	เมนูการเลือกเวอร์ชันในการเริ่ม โปรแกรม	96
รูปที่ ค.4	เมนูการตั้งค่าพารามิเตอร์	97
รูปที่ ค.5	เมนูการทดสอบ Step Response	97

บทที่ 1

บทนำ

วิวัฒนาการในการควบคุมเครื่องจักรกลต่างๆ ได้พัฒนามาถึงขั้นที่ไม่ต้องใช้มนุษย์คอยควบคุมการทำงานตลอดเวลา แต่ให้เครื่องจักรทำงานโดยอัตโนมัติได้ ในปัจจุบันมีการนำเครนมาใช้ยกของที่มีน้ำหนักมาก อย่างแพร่หลาย ทั้งในงานก่อสร้าง ในโรงงานอุตสาหกรรม หรือการขนตู้คอนเทนเนอร์ในท่าเรือ ซึ่งได้มีการศึกษาและพัฒนาระบบการนำเครนมาใช้งาน เช่น การควบคุมการแกว่งของภาระ การควบคุมการทำงานให้ใช้เวลาน้อยที่สุด เป็นต้น เพื่อช่วยเพิ่มประสิทธิภาพการทำงานให้ดีขึ้น



รูปที่ 1.1 แสดงการใช้เครนยกของในโรงงานอุตสาหกรรม

1.1 ประวัติการศึกษาการควบคุมเครน

ในปี ค.ศ. 1979 Mita และ Kannai ได้พัฒนาวิธีการควบคุมการแกว่งของภาระโดยวิธีการควบคุมออปติมัล (Optimal Control) เพื่อให้ใช้เวลาในการควบคุมน้อยที่สุด

ในปี ค.ศ. 1987 J. W. Auerning ปรับปรุงวิธีแก้ปัญหาคอมพิวเตอร์ให้ใช้เวลาในการควบคุม (Minimum time control) โดยการเปลี่ยนค่าความยาวลวดที่แขวนภาระขณะทำการควบคุม

ในปี ค.ศ. 1993 J. Shirai ได้ทำการพัฒนาวิธีการคำนวณแบบใหม่ เพื่อใช้ในการควบคุมการแกว่งของภาระที่แขวนอยู่บนเครน ที่ใช้ในการขนตู้คอนเทนเนอร์ภายในท่าเรือ

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 จุดประสงค์ของการศึกษาและการควบคุมการเคลื่อนที่ของเครน

จุดประสงค์หลักในการควบคุมการเคลื่อนที่ของเครนมีเหตุผลดังต่อไปนี้

- 1.2.1 ช่วยเพิ่มประสิทธิภาพการทำงานโดยเฉลี่ยแม้ว่าผู้ปฏิบัติงานจะไม่มี ความกดดัน เนื่องจากการใช้งานที่ง่ายและสะดวก
- 1.2.2 ช่วยเพิ่มประสิทธิภาพการทำงานในสภาวะแวดล้อมที่ไม่เหมาะสม เช่น การที่ผู้ปฏิบัติงานต้องทำงานในบริเวณที่มีอุณหภูมิสูง หรือในบริเวณที่มีทัศนวิสัยไม่ดี เป็นต้น
- 1.2.3 ช่วยลดค่าใช้จ่ายแรงงาน
- 1.2.4 ช่วยทำให้งานเสร็จเร็วยิ่งขึ้น
- 1.2.5 สามารถนำไปประยุกต์ใช้งานกับกระบวนการผลิต หรือกระบวนการขนส่ง ในโรงงานอุตสาหกรรมได้อย่างมีประสิทธิภาพ
- 1.2.6 ช่วยลดความเสี่ยงของการทำงาน และได้งานที่มีคุณภาพ

1.3 เนื้อหาที่จะกล่าวถึงในปริญญาานิพนธ์ฉบับนี้

ในปริญญาานิพนธ์ฉบับนี้จะประกอบไปด้วยเนื้อหา ดังนี้

- 1.3.1 โครงสร้างเชิงกล และวงจรถวลีเกททรอนิกส์ที่ใช้ในการวัดค่าสัญญาณต่างๆ ที่ใช้ในแบบจำลองเครน
- 1.3.2 แผ่นวงจรที่ใช้ติดต่อกับคอมพิวเตอร์และวงจรควบคุมมอเตอร์
- 1.3.3 การประมาณค่าแบบจำลองทางคณิตศาสตร์ของระบบเครน จากผลตอบสนองที่ได้จากการทดลอง
- 1.3.4 การควบคุมการเคลื่อนที่ของเครน โดยการป้อนกลับสเตท (State Feedback) ซึ่งออกแบบค่าเกนป้อนกลับ (Feedback Gain) โดยวิธี Linear Quadratic Regulator (LQR)
- 1.3.5 การทดลองและผลการทดลอง
- 1.3.6 การใช้งานชุดควบคุม ในการควบคุมแบบจำลองเครน

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการในการควบคุมและบล็อกไดอะแกรมของระบบ

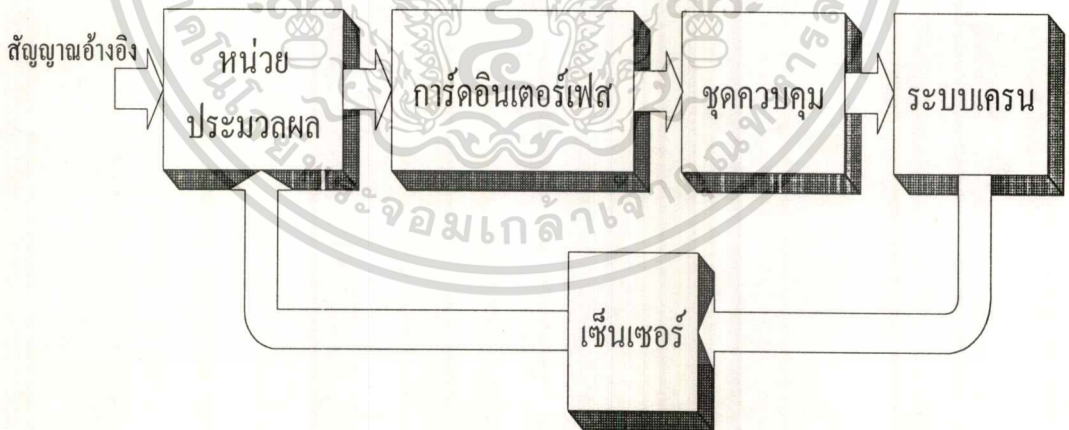
2.1.1 หลักการในการควบคุม

หลักการในการควบคุมระบบการเคลื่อนที่ของครนให้ไปยังเป้าหมาย พร้อมทั้งสามารถควบคุม มุมที่เกิดจากการแกว่งของภาระของครน คือเราต้องควบคุมการหมุนของมอเตอร์ให้ได้ เพื่อควบคุมการ เคลื่อนที่ของครนให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการได้ และในขณะเดียวกันก็ทำให้เกิดการแกว่งของ ภาระของครนมีค่าน้อยที่สุด

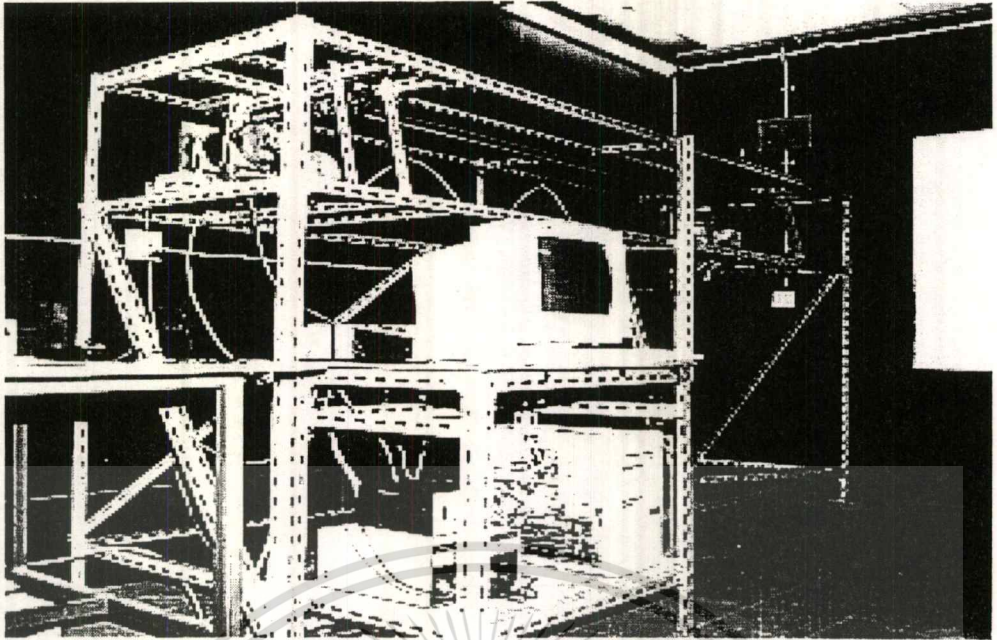
การควบคุมการหมุนของมอเตอร์ ได้จากการทำงานร่วมกันของ คอมพิวเตอร์ ชดควบคุม และ เซ็นเซอร์ (Sensor) โดยเซ็นเซอร์จะทำการวัดสัญญาณป้อนกลับ (Feedback Signal) ค่าตำแหน่งของรด ครน และค่ามุมการแกว่ง มาคำนวณหาค่าสเตตต่างๆ และสัญญาณควบคุม (Control Signal) หรือ สัญญาณความเร็วอ้างอิง (Velocity Reference Command) ส่งไปยังชดควบคุม เพื่อควบคุมให้ครนเคลื่อน ที่ไปยังตำแหน่งที่ต้องการ และมีมุมการแกว่งของภาระเป็นศูนย์ โดยไม่มีค่าผิดพลาด หรือมีค่าผิดพลาด น้อยที่สุด

2.1.2 บล็อกไดอะแกรมของระบบ

บล็อกไดอะแกรมโดยรวมของระบบสามารถเขียนได้ดังนี้



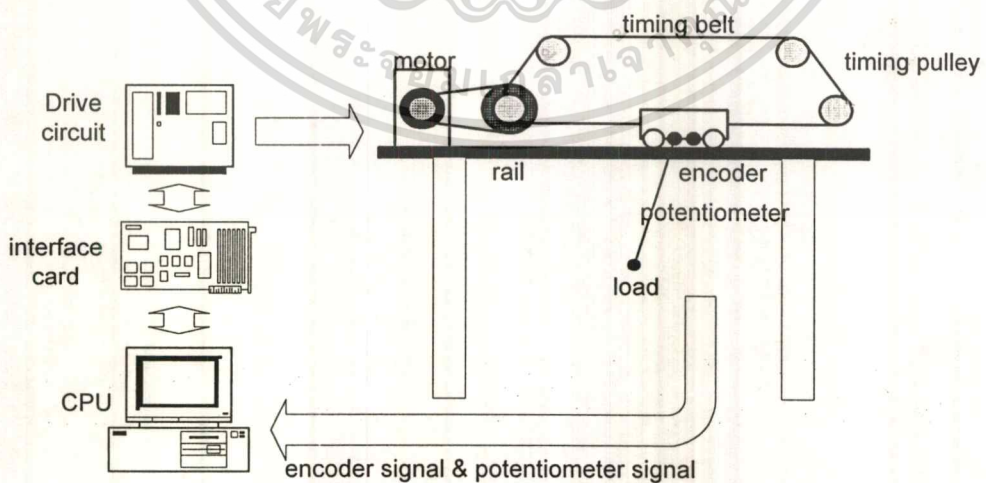
รูปที่ 2.1 แสดงบล็อกไดอะแกรมโดยรวมของระบบการควบคุมการเคลื่อนที่ของครน



รูปที่ 2.2 ระบบการควบคุมการเคลื่อนที่ของเครื่อง

2.2 โครงสร้างและส่วนประกอบของแบบจำลองเครื่อง

แบบจำลองเครื่องที่ออกแบบไว้ใช้รางที่ทำด้วยสแตนเลสมีระยะทางวิ่งประมาณ 2 เมตร และสามารถยกภาระขึ้นได้สูงประมาณ 80 เซนติเมตร การวิ่งในแนวระนาบใช้ดีซีมอเตอร์เป็นตัวส่งกำลังไปขับเคลื่อนรถผ่านระบบส่งกำลัง เพื่อให้ได้แรงบิดสูงขึ้น 3 เท่า และให้มีค่าเบ็คเลข (back latch) และค่าสลিপ (slip) น้อยที่สุด เพื่อให้ได้ความแม่นยำ และรวดเร็วที่สุดในการควบคุมการเคลื่อนที่ของตัวเครื่อง



รูปที่ 2.3 แสดงโครงสร้างและส่วนประกอบของแบบจำลองเครื่อง

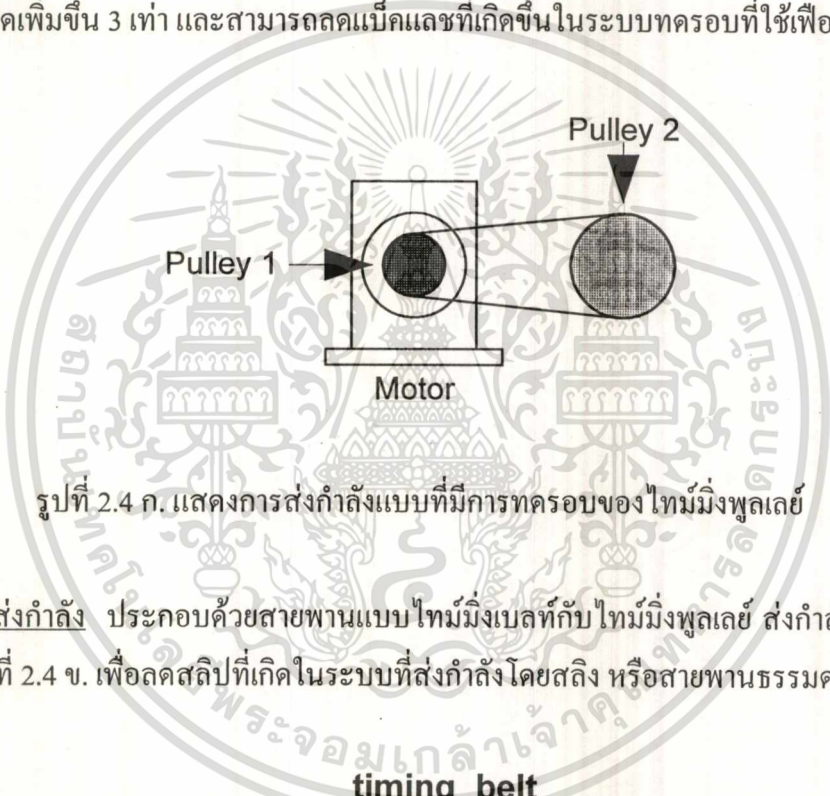
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมีคุณสมบัติของอุปกรณ์ต่างๆ ของแบบจำลองเครนต่างประกอบไปด้วย

2.2.1 ตัวส่งกำลัง เลือกใช้ดีซีมอเตอร์แบบแม่เหล็กถาวรเป็นตัวส่งกำลัง ไปยังตัวเครนซึ่งให้อัตราส่วนระหว่างกระแสแอมเพอร์และแรงบิดมีค่าคงที่ ให้ความสัมพันธ์ระหว่างกระแสแอมเพอร์ แรงบิดและความเร็ว เป็นเชิงเส้น และไม่มีกำลังสูญเสียในขดลวดฟิลด์ มีประสิทธิภาพสูงกว่า ให้ค่ากระแสแอมเพอร์สูงกว่าและขนาดเล็กกว่า เมื่อเทียบกับดีซีมอเตอร์แบบขดลวดฟิลด์ ที่มีขนาดแรงม้าเท่ากัน

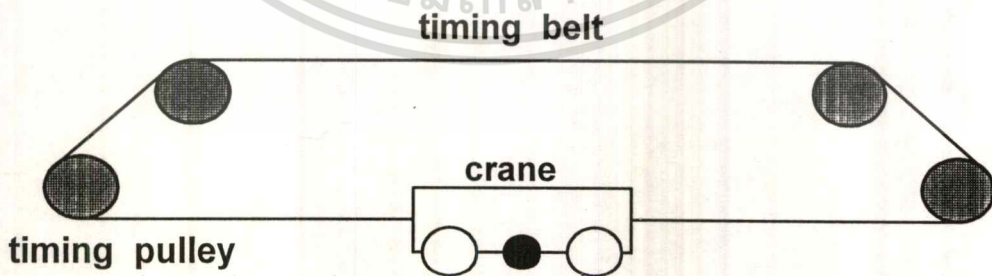
2.2.2 ระบบส่งกำลัง ประกอบด้วย

- ชุดทศรอบ ประกอบด้วยสายพานแบบไทม์มิ่งเบลท์ (timing belt) และไทม์มิ่งพูลเลย์ (timing pulley) 2 ขนาด ที่มีอัตราส่วนของรัศมีของไทม์มิ่งพูลเลย์ที่ยึดอยู่กับมอเตอร์เป็น 1/3 ของไทม์มิ่งพูลเลย์ที่ยึดกับแกนหมุน ของชุดส่งกำลังที่ส่งกำลัง ไปยังระบบ ดังรูปที่ 2.4 ก. เพื่อทศรอบมอเตอร์ให้ความเร็วลดลง แต่ให้แรงบิดเพิ่มขึ้น 3 เท่า และสามารถลดแบ็คแลชที่เกิดขึ้นในระบบทศรอบที่ใช้เฟืองเกียร์ได้



รูปที่ 2.4 ก. แสดงการส่งกำลังแบบที่มีการทศรอบของไทม์มิ่งพูลเลย์

- ชุดส่งกำลัง ประกอบด้วยสายพานแบบไทม์มิ่งเบลท์กับไทม์มิ่งพูลเลย์ ส่งกำลังไปยังตัวเครนโดยตรง ดังรูปที่ 2.4 ข. เพื่อลดสลลิปที่เกิดในระบบที่ส่งกำลังโดยสลิง หรือสายพานธรรมดา



รูปที่ 2.4 ข. แสดงการส่งกำลังแบบที่ไม่มีการทศรอบของไทม์มิ่งพูลเลย์



รูปที่ 2.4 ก. ชุดส่งกำลังของระบบเครน

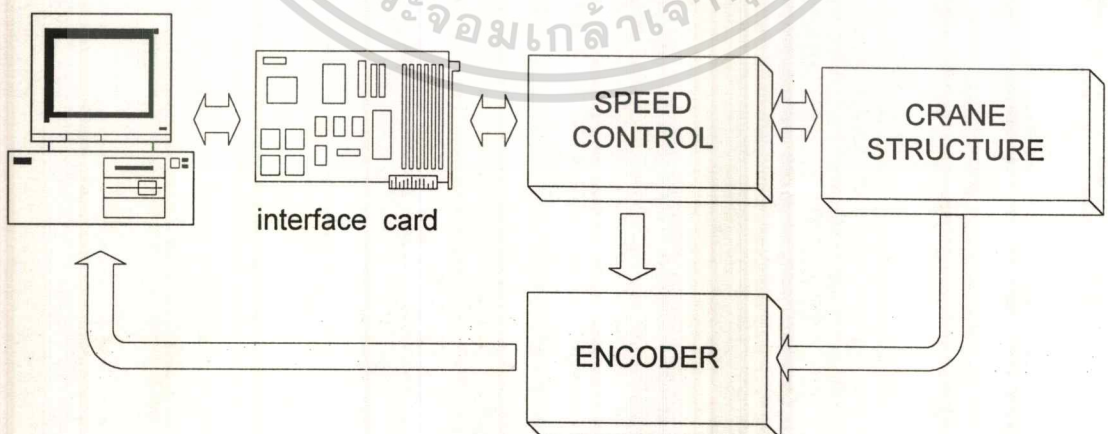
2.3 ชุดควบคุมและเซ็นเซอร์

2.3.1 ชุดควบคุมการเคลื่อนที่ของเครน ประกอบด้วย

2.3.1.1 คอมพิวเตอร์ส่วนบุคคล (Personal Computer ,PC)

2.3.1.2 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์ (Interface Card)

2.3.1.3 ชุดควบคุมความเร็ว (Speed Control)



รูปที่ 2.5 รูปแสดงบล็อกไดอะแกรมของชุดควบคุมการเคลื่อนที่ของเครน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 คอมพิวเตอร์ส่วนบุคคล

คอมพิวเตอร์ที่เลือกใช้เป็นแบบที่เข้ากันได้กับของบริษัท IBM เนื่องจากเป็นคอมพิวเตอร์ส่วนบุคคลที่หาได้ง่าย และนิยมใช้กันอย่างแพร่หลายในปัจจุบัน ใช้ควบคุมระบบโดย software ที่ถูกเขียนขึ้น โดยใช้ทฤษฎีระบบควบคุม

2.3.1.2 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์

เป็นวงจรที่ทำหน้าที่ช่วยให้คอมพิวเตอร์สามารถติดต่อส่งผ่านสัญญาณข้อมูล และสัญญาณควบคุมซึ่งเป็น PWM (Pulse Width Modulate) Magnitude และ PWM Sign ขนาด 5V TTL กับวงจรขับมอเตอร์ หรืออุปกรณ์ภายนอกได้

2.3.1.3 ชุดควบคุมความเร็ว

ใช้สัญญาณควบคุมที่ได้จากวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมความเร็วมาป้อนให้กับวงจรขับมอเตอร์โดยที่

ค่า PWM MAG เป็น 0 มอเตอร์จะหยุดทำงาน

ค่า PWM MAG เป็น 1 มอเตอร์จะทำงานตามสัญญาณ PWM SIGN

ค่า PWM SIGN เป็น 0 มอเตอร์จะหมุนทวนเข็มนาฬิกา

ค่า PWM SIGN เป็น 1 มอเตอร์จะหมุนตามเข็มนาฬิกา

และจ่ายแรงดันไฟตรงให้กับวงจรขับมอเตอร์เพื่อให้รถเคลื่อนที่ มากกว่า 12 โวลต์ แต่ไม่เกิน 45 โวลต์ และในขณะที่รถเคลื่อนที่นั้น เอนโคเดอร์ก็นำหน้าที่เป็นเซ็นเซอร์ส่งผ่าน เฟส A,B ไปยังชิพควบคุมความเร็ว LM629¹ ตัวที่ 1 เพื่อเป็นเกาท์เตอร์คำนวณระยะทางที่รถเคลื่อนที่ไปได้ และให้รถหยุดได้ตามต้องการ

2.3.2 วงจรเซ็นเซอร์

เซ็นเซอร์เป็นส่วนประกอบที่สำคัญในระบบคอนโทรลที่มีการป้อนกลับเพราะสัญญาณที่ป้อนกลับผ่านมาทางเซ็นเซอร์นั้น สามารถนำมาวิเคราะห์หาคุณสมบัติของระบบหรือสามารถบอกสถานะของระบบขณะนั้นได้ว่าเป็นอย่างไร ซึ่งจะได้นำมาหาสัญญาณควบคุมที่จะส่งไปควบคุมระบบ ดังนั้นถ้าสัญญาณป้อนกลับที่ได้มีความถูกต้องมากและมีสัญญาณรบกวนน้อย ก็จะได้ค่าสัญญาณควบคุมที่สามารถควบคุมระบบให้เกิดประสิทธิภาพสูงสุดได้ ซึ่งประกอบไปด้วยเซ็นเซอร์ 2 ชนิด คือ โปเทนทิโอมิเตอร์ (Potentiometer) และเอนโคเดอร์ (Encoder)

2.3.2.1 โปเทนทิโอมิเตอร์ (Potentiometer)

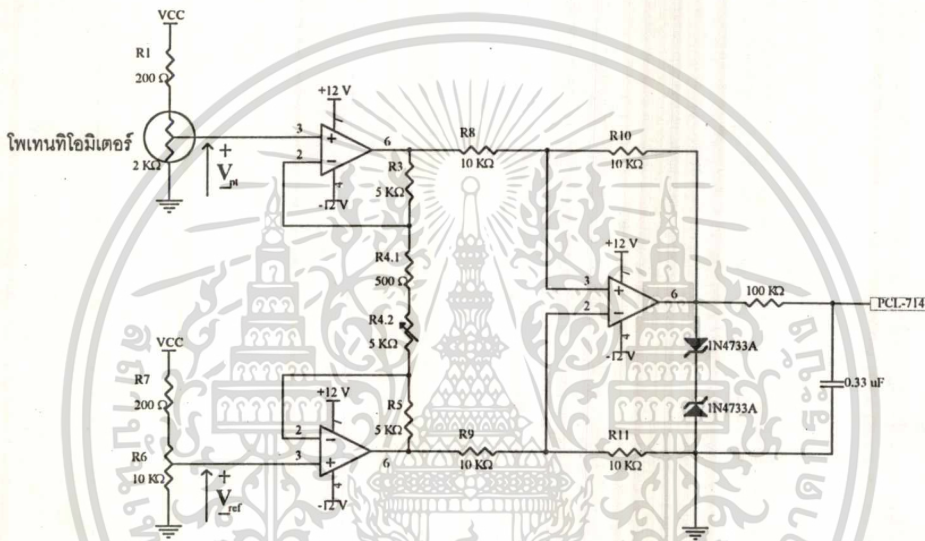
ใช้วัดค่ามุมของการแกว่งของภาระ โดยโปเทนทิโอมิเตอร์เป็นตัวต้านทานปรับค่าได้ชนิดหนึ่งที่มีแกนกลางหมุนได้ ทำให้ค่าความต้านทานเปลี่ยนแปลง มี 2 แบบที่นิยมใช้กัน คือ แบบ Wire wound มีค่าความละเอียดต่ำ แต่ทนกระแสได้ดี และอีกแบบที่เราเลือกใช้ คือ แบบฟิล์มพลาสติก (ขนาดสูงสุด 2

¹ อ้างอิง: ภาคผนวก ก. หนังสืออ้างอิงลำดับที่ 1 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิโลโอห์ม) ซึ่งมีค่าความละเอียดสูง ให้แรงดันเอาต์พุตที่เรียบ มีสัญญาณรบกวนต่ำ และค่าความต้านทานที่เกิดขึ้นจะมีความสัมพันธ์กับระยะทางเชิงมุมที่เกิดขึ้นเป็นเชิงเส้น

วงจรตรวจวัดค่ามุมโดยใช้โพเทนทิโอมิเตอร์

โดยให้โพเทนทิโอมิเตอร์ ทำหน้าที่เปลี่ยนค่ามุมที่ได้มาเป็นสัญญาณ โวลต์เตจ ดังในรูปวงจร โดยการใช้การจ่ายค่าแรงดันคงที่ให้กับโพเทนทิโอมิเตอร์ ค่าแรงดันตกรวมที่ได้จะถูกนำมาเปรียบเทียบกับค่าแรงดันที่ได้จากค่าความต้านทานปรับค่าได้ R6 (ขนาดสูงสุด 10 กิโลโอห์ม) ซึ่งใช้เป็นแรงดันอ้างอิง และค่าแรงดันที่ได้จะถูกขยายสัญญาณ โดย Instrument Amplifier ซึ่งมีอัตราขยายเป็นแบบ differential ที่สามารถปรับค่าได้จากความต้านทาน R4.2 (ขนาดสูงสุด 5 กิโลโอห์ม) ดังนั้นค่า V_{out} ก็จะมี ความสัมพันธ์กับค่ามุมของการแกว่งที่เกิดขึ้น เป็นเชิงเส้นเช่นกัน



รูปที่ 2.6 วงจรตรวจวัดค่ามุม

จากรูปจะ ได้ความสัมพันธ์ระหว่างค่าแรงดันเอาต์พุตและค่าแรงดันอินพุตดังนี้

$$V_{out} = \left(\frac{R_{10}}{R_a R_8} \right) (R_a + R_3 + R_5) (V_{pt} - V_{ref})$$

เมื่อ V_{pt} คือ ค่าแรงดันที่ได้จากการหมุนของโพเทนทิโอมิเตอร์

V_{ref} คือ ค่าแรงดันอ้างอิงที่ได้จากการปรับค่าความต้านทาน R6

V_{out} คือ ค่าแรงดันเอาต์พุต มีค่าอยู่ในช่วง -5 V ถึง $+5$ V

$$R_a = R_{4.1} + R_{4.2}$$

$$R_8 = R_9 = R_{10} = R_{11} = 10 \text{ กิโลโอห์ม}$$

$$R_3 = R_5 = 5 \text{ กิโลโอห์ม}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้ค่ามูบ้อนกลับที่แปลงเป็นสัญญาณโวลต์เตจ แล้วสัญญาณที่ได้จะผ่าน Low-Pass Filter เพื่อกรองสัญญาณรบกวนที่ความถี่สูงออก และแปลงสัญญาณจากอนาลอกเป็นดิจิตอลโดยผ่านการ์ด PCL-714² เพื่อให้สัญญาณที่ได้สามารถนำไปคำนวณในคอมพิวเตอร์ได้

*** ข้อควรระวัง สำหรับการใช้การ์ด PCL-714

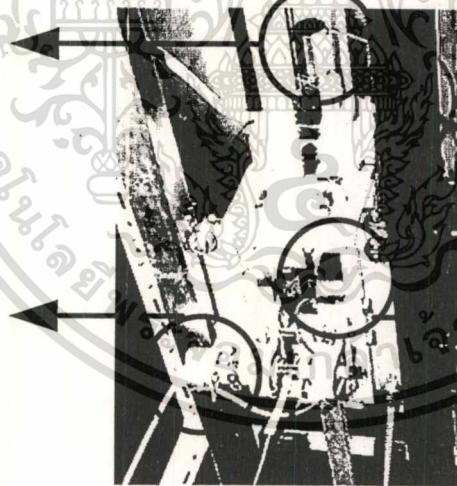
เนื่องจาก PCL-714 ตัวนี้ใช้ 4052 CMOS chip ในการ multiplex สำหรับ analog input channel ซึ่งห้ามไม่ให้โวลต์เตจสูงกว่า 12.5 V หรือต่ำกว่า -5.5 V ถ้า analog input channel ได้รับนอกเหนือจากช่วงนี้แล้ว จะทำให้วงจร multiplexer เสียหายได้

2.3.2.2 เอนโคเดอ (Encoder)

ใช้เพื่อวัดระยะทางของรถเข็น ติดตั้งไว้ที่ด้านข้างของรถเข็น เมื่อรถเข็นเคลื่อนที่ไปก็จะทำให้ลูกยางที่ยึดอยู่กับแกนของเอนโคเดอเคลื่อนไปด้วย สัญญาณที่ได้จากการหมุนของเอนโคเดอจะถูกป้อนเข้าไปที่ ไอซี LM629 ตัวที่ 1 และเราสามารถอ่านค่าระยะทางของเข็นที่เคลื่อนที่ไปได้ โดยใช้คอมพิวเตอร์อ่านข้อมูลจากรีจิสเตอร์ของ ไอซี LM629

เอนโคเดอสำหรับ
มอเตอร์ในแนวแกนตั้ง

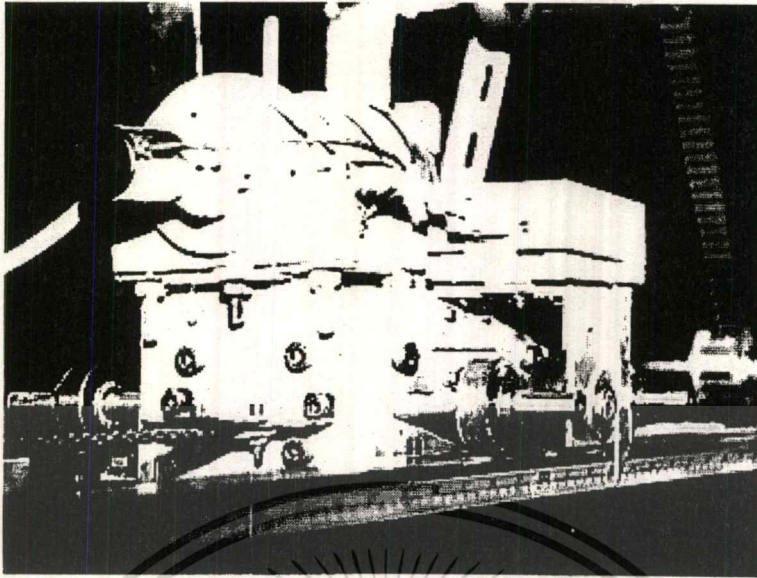
โพเทนทิโอมิเตอร์



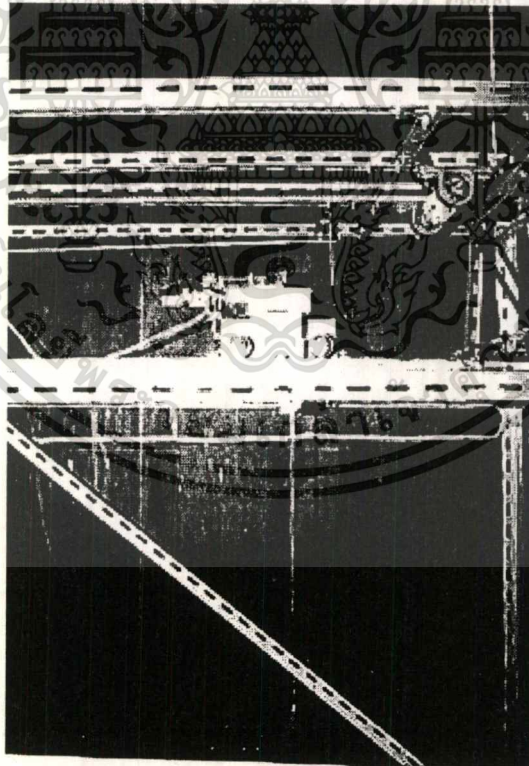
เอนโคเดอสำหรับมอเตอร์
ในแนวแกนนอน

รูปที่ 2.7 เซ็นเซอร์ต่างๆบนระบบเข็น

² อ้างอิง: ภาคผนวก ข. หนังสืออ้างอิงลำดับที่ 1 งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงตัวรถเครน



รูปที่ 2.9 แสดงรถเครนและภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

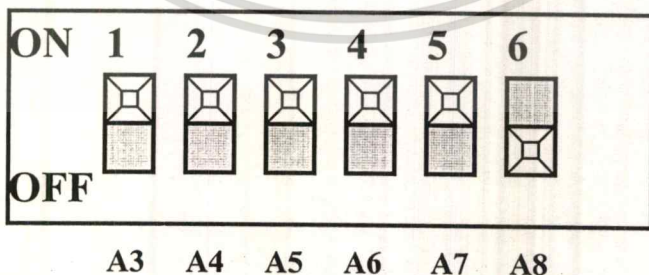
วงจรโดยรวมของระบบ

3.1 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์ (Interface Card)

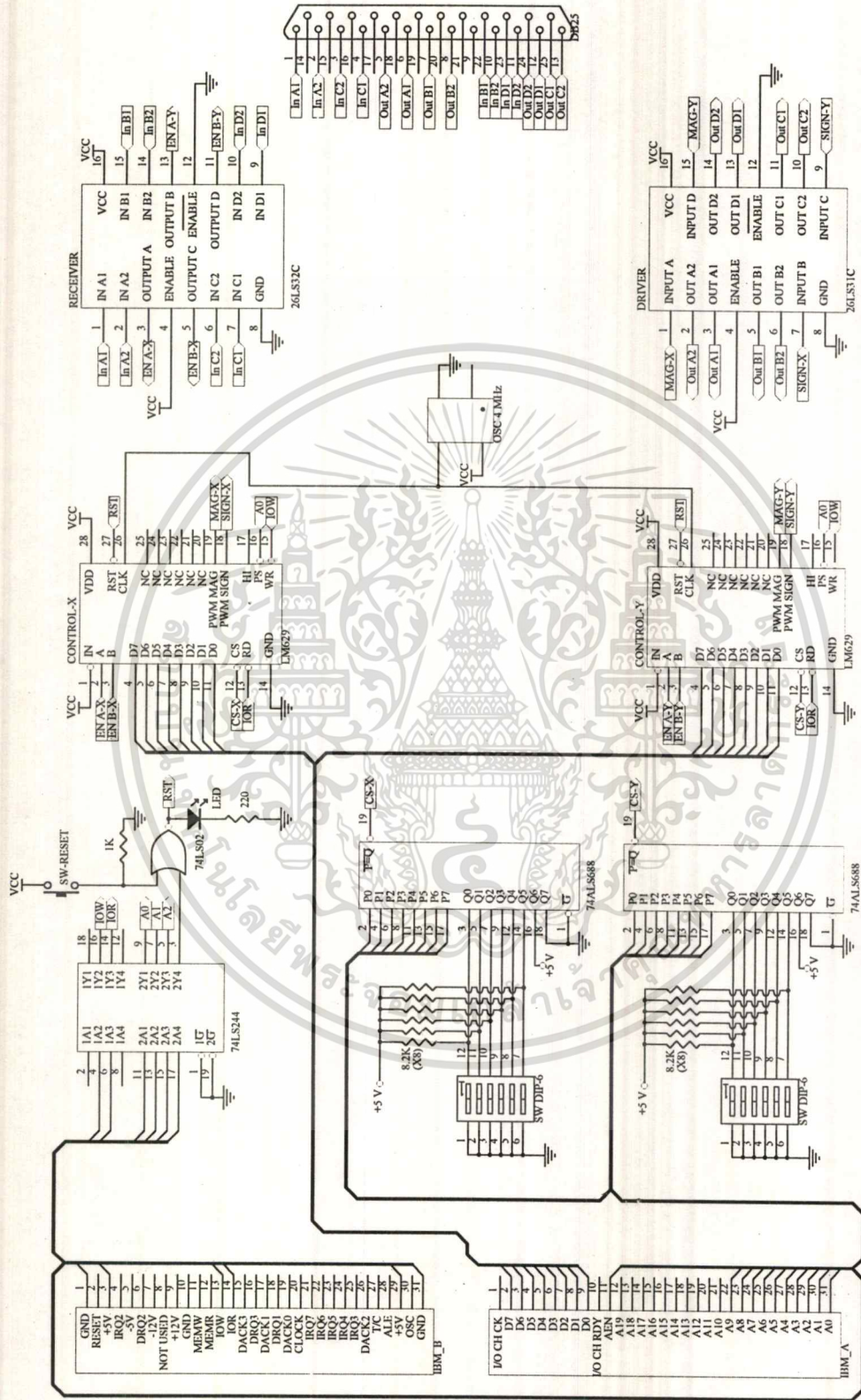
เป็นการ์วงจรที่ออกแบบขึ้นเพื่อใช้ในระบบควบคุมการเคลื่อนที่ของเครน และได้รวมชิพ ควบคุมความเร็วไอซี LM629 สองตัวเข้าไว้บนการ์ดนี้ด้วย โดยวงจรเชื่อมต่อนี้จะทำหน้าที่เป็นเหมือน พอร์ต (Port) และทำหน้าที่เป็นบัฟเฟอร์ของสัญญาณข้อมูล (Data Buffer) และบัฟเฟอร์ของสัญญาณควบคุมต่างๆ (Signal Buffer) ซึ่งช่วยให้คอมพิวเตอร์สามารถติดต่อส่งผ่านข้อมูลกับวงจรควบคุมมอเตอร์ได้ โดยในโครงการครั้งนี้ ได้ปรับปรุงในส่วนของตัวการ์ดและวงจรจากโครงการเดิม ซึ่งในส่วนของตัวการ์ดได้ทำเป็นแผ่นปริ้นท์การ์ดวงจร เพื่อความเป็นระเบียบ คงทน และมีประสิทธิภาพ ในส่วนของวงจร ได้เพิ่ม ไอซี 26LS31 และ 26LS32 ซึ่งเป็นตัวขับและตัวรับในสายส่ง ตามลำดับ เพื่อความชัดเจนและแน่นอนของสัญญาณ ไอซี LM629 ตัวที่ 1 (LM629-X) ใช้ในการรับส่งสัญญาณในการควบคุมตำแหน่งการเคลื่อนที่ของรถเครน และเตรียมไอซี LM629 ตัวที่ 2 (LM629-Y) ไว้เพื่อพัฒนาในการยกภาระขึ้นลงได้ต่อไป ใช้ไอซี 74LS244 ทำหน้าที่เป็นบัฟเฟอร์ของสัญญาณควบคุม ส่งสัญญาณไปควบคุมการทำงานของพอร์ตต่างๆในวงจรเชื่อมต่อได้ถูกต้อง และผ่านแอดเดรสที่ต้องการ ดังแสดงในรูปที่ 3.2

หลักในการกำหนดแอดเดรส มีดังนี้

เนื่องจากเราใช้ไอซี 74LS688 ในการดีโคดแอดเดรส (Decode Address) ดังนั้นเราจะสามารถเลือกใช้แอดเดรสเพื่อใช้ติดต่อกับวงจรควบคุมมอเตอร์ได้จากตำแหน่งการ ปิด-เปิด ของคิปสวิทช์ ตัวอย่าง เช่น ถ้าเราต้องการใช้แอดเดรส 300H (11 0000 0XXX B) ในการติดต่อกับวงจรควบคุมมอเตอร์ หรือ A9=1, A8=1, A7=0, A6=0, A5=0, A4=0, A3=0 ส่วน A2, A1 และ A0 จะมีค่าเป็นอะไรก็ได้ ดังนั้นเราจะสามารถกำหนดตำแหน่งของคิปสวิทช์ได้ ดังแสดงในรูปที่ 3.1 ซึ่ง A9 ได้กำหนดให้เป็น 1 แล้ว

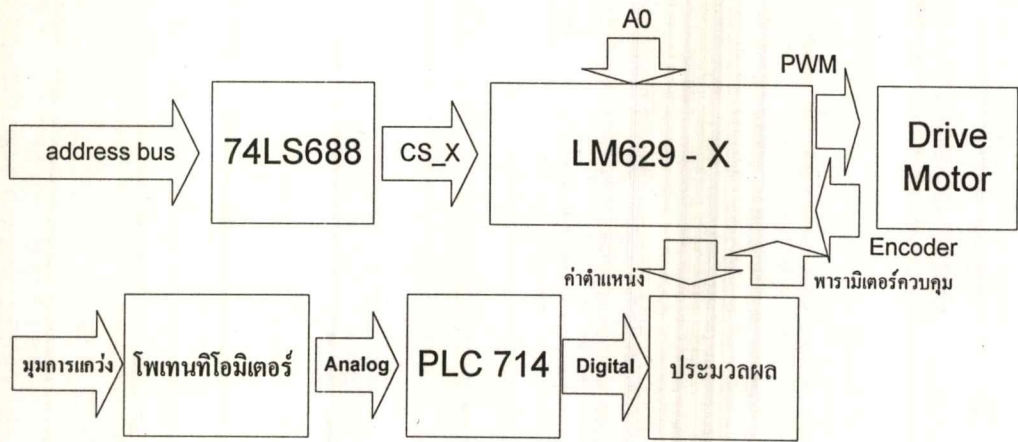


รูปที่ 3.1 แสดงตัวอย่างการเลือกตำแหน่งของคิปสวิทช์



รูปที่ 3.2 แสดงวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรมอดูมความเร็ว LM629

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงการทำงานโดยรวมของวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์

หลักการการทำงานของวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์

ไอซี 74LS688 จะดีโคดแอดเดรสจากแอดเดรสบัส (Address Bus) A0-A9 เมื่อตรงกับแอดเดรสที่กำหนดไว้จากการ เปิด-ปิด ของคิปสวิทช์แล้ว จะส่งสัญญาณ CS-X ไปที่ขา CS (Chip Select) ของชิพ LM629-X (LM629 ที่ควบคุมการเคลื่อนที่ในแนวแกน X) และเมื่อได้รับสัญญาณจากบิต A0 ที่ขา PS (Port Select) แล้ว ไอซี LM629-X จะทำงานโดยส่งสัญญาณเป็น PWM (Pulse Width Modulate) Magnitude ควบคุมความเร็วของการเคลื่อนที่ และ PWM Sign ควบคุมทิศทาง การเคลื่อนที่ ไปที่วงจรขับมอเตอร์ และรับค่าตำแหน่งของการเคลื่อนที่ที่ส่งกลับมาทางเอ็นโคดเดอร์ เฟส A และ B แล้วไอซี LM629-X จะส่งค่าตำแหน่งนี้ผ่านไปให้คอมพิวเตอร์ทำการประมวลผลร่วมกับค่ามุมการแกว่งที่ได้จากโพเทนทิโอมิเตอร์ ซึ่งเป็นสัญญาณอนาลอก ผ่านให้การ์ด PCL-714 แปลงเป็นสัญญาณดิจิทัล เมื่อประมวลผลแล้ว คอมพิวเตอร์ก็จะส่งค่าพารามิเตอร์ควบคุมกลับไปให้ชิพ LM629-X เพื่อให้ส่ง PWM ไปให้วงจรขับมอเตอร์ควบคุมการเคลื่อนที่จนได้ตำแหน่งที่ต้องการ

การปรับปรุงพัฒนาวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์

- เนื่องจากได้รวมเอาชิพควบคุมความเร็ว LM629 ทั้ง 2 ตัว เข้าไว้บนการ์ดนี้ด้วย จึงใช้การกำหนดแอดเดรสใช้ไอซี 74LS688 สองตัว กำหนดแอดเดรสแยกให้ชิพ LM629 แต่ละตัว ซึ่งช่วยเพิ่มความยืดหยุ่นในการกำหนดแอดเดรสจากการ เปิด-ปิด ของคิปสวิทช์ ภายในช่วงแอดเดรสที่คอมพิวเตอร์ยอมให้ใช้ได้ (200H-3FFH)

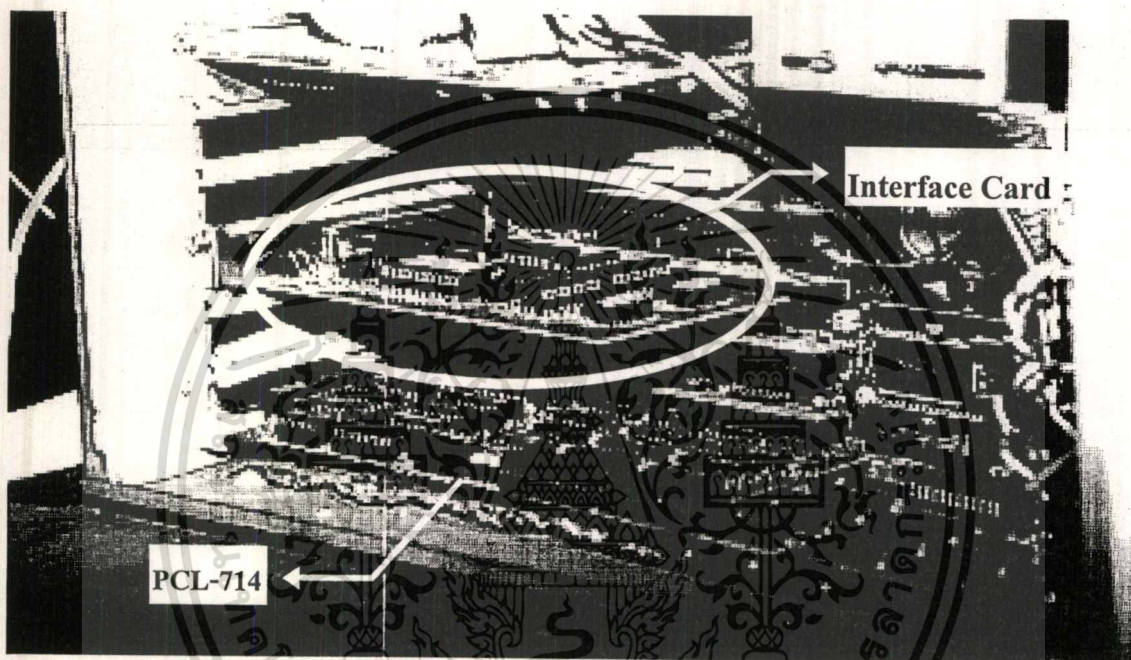
- เพิ่มวงจรขับและรับสัญญาณในสายส่ง (Line Driver) คือ ไอซี 26LS31 เป็นตัวขับสัญญาณ (Driver) และไอซี 26LS32 เป็นตัวรับสัญญาณ (Receiver) เพื่อช่วยให้สัญญาณที่ได้รับหรือส่งออกไปนั้น มีความชัดเจน และแน่นอน

- แก้ปัญหาเรื่องความร้อนและสัญญาณรบกวน โดยได้ตรวจพบสาเหตุประการหนึ่งคือ พบว่ามีลูปตัวนำ ทั้งระบบกราวด์และระบบไฟ ซึ่งเป็นสาเหตุให้เกิดการเหนี่ยวนำกระแสไฟฟ้าขึ้น ทำให้ไปเอกสารถนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รบกวนการทำงานของไอซีตัวอื่นๆ ให้ทำงานผิดพลาด จึงแก้ไขไม่ให้เกิดเป็นลูปตัวนำ และเมื่อได้ตรวจสอบข้อผิดพลาดต่างๆ และทำการแก้ไขทั้งหมดแล้ว ก็ไม่เกิดปัญหาเรื่องความร้อนหรือสัญญาณรบกวนใดๆอีก

- เพิ่มสวิตช์รีเซ็ตไอซี LM629 เพื่อป้องกันระบบอันเนื่องจากการรวนของไอซี LM629 หรือจากการคงค่าเดิมของพารามิเตอร์ต่างๆ ที่อาจเกิดขึ้นได้

- จัดทำเป็นแผ่นวงจรกร์ดवार (แผ่นปริ้นท์) เพื่อความเป็นระเบียบ แน่นอน กงทน และมีประสิทธิภาพ



รูปที่ 3.4 แสดงแผ่นวงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์ และการ์ด PCL-714

3.2 วงจรชุดควบคุมความเร็ว (Speed Control)

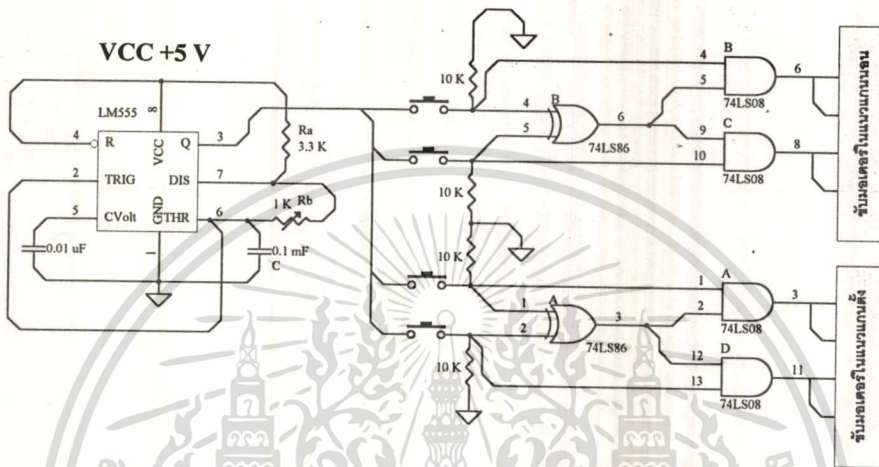
แบ่งออกเป็น 3 ส่วน ดังนี้

1. วงจรควบคุมด้วยมือ
2. วงจรขับเคลื่อนมอเตอร์
3. วงจรป้องกันการชน

ทั้งนี้เพื่อให้การพัฒนาในอนาคต ซึ่งจะพัฒนาให้สามารถควบคุมในแนวแกนตั้ง หรือสามารถควบคุมการยกภาระขึ้นลงได้ จึงได้เตรียมวงจรที่จำเป็นต่อการควบคุมไว้แล้ว

3.2.1 วงจรควบคุมด้วยมือ

ใช้ไอซี LM555 เป็นตัวกำเนิดสัญญาณในการขับเคลื่อนมอเตอร์ โดยปรับค่าของความถี่และค่าของคิวตี้ไซเคิล จากการปรับที่ตัวต้านทานปรับค่าได้ ซึ่งจะได้ค่าของคาบการทำงานและค่าของความถี่สัมพันธ์กับค่าของตัวต้านทานและตัวเก็บประจุ ตามรูปที่ 3.5



รูปที่ 3.5 วงจรกำเนิดสัญญาณการขับเคลื่อนมอเตอร์โดยใช้ LM555

ดังสมการ¹ต่อไปนี้
ค่าความถี่

$$F = \frac{1.44}{(R_A + 2R_B)C}$$

ค่าคิวตี้ไซเคิล

$$D = \frac{R_B}{R_A + 2R_B}$$

ให้ $R_A = 33$ กิโลโอห์ม และ $C = 0.1$ ไมโครฟารัด

ดังนั้นจึงได้ค่าของ ความถี่และคิวตี้ไซเคิลแปรตาม R_B ปรับค่าได้ 1 กิโลโอห์มเป็น

$$F = \frac{14.4}{33 K + R_B}$$

¹ อ้างอิง : LM555/LM555C Applications Information, National Application Specific Analog Products Databook,

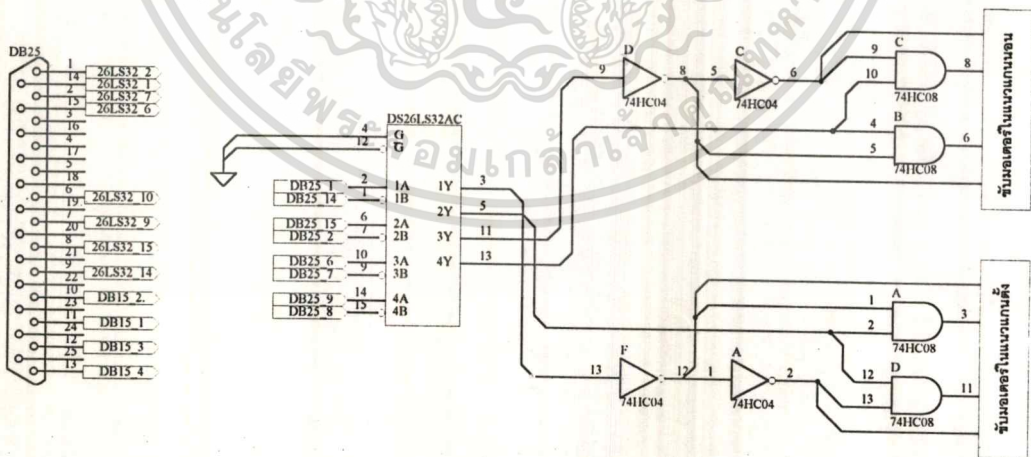
$$D = \frac{R_B}{33 K + 2R_B}$$

หลังจากนั้นใช้สวิทช์กดติดปล่อยดับเป็นตัวเลือกทิศทางในการขับเคลื่อนมอเตอร์ เข้าสู่วงจรป้องกันการขับเคลื่อนพร้อมกัน 2 ทิศทางในแนวแกนเดียวโดยใช้ ไอซีเบอร์ 74LS86 ซึ่งเป็นเอ็กซอร์คูซีฟอเกต (Exclusive Or) และ ไอซีเบอร์ 74LS08 ซึ่งเป็นแอนเกต (AND) ซึ่งมีหลักการทำงานคือ

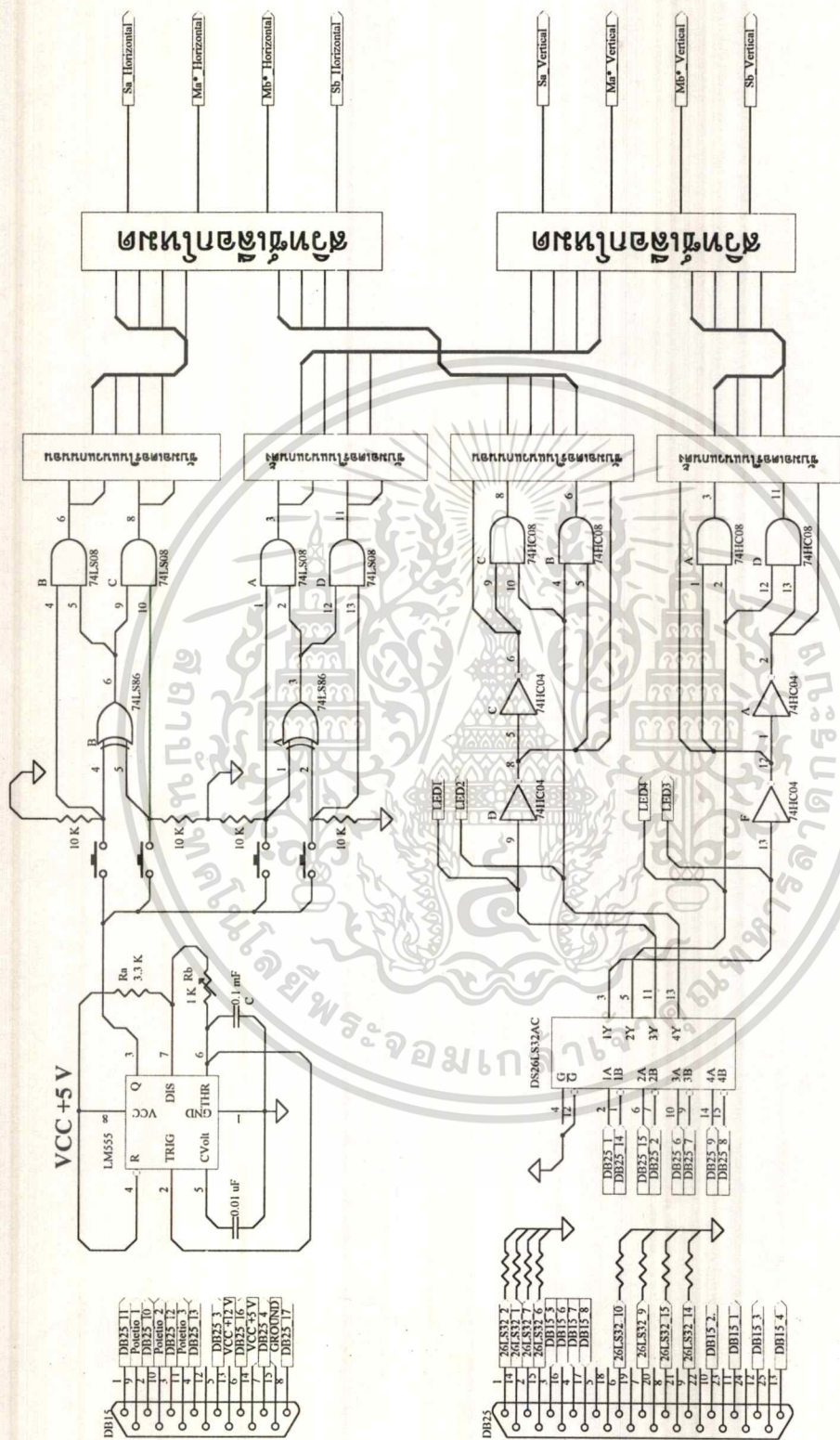
เมื่อกดสวิทช์ตัวเดียว สัญญาณที่ผ่าน 74LS86 จะมีค่าเหมือนกับค่าสัญญาณที่เข้า ทำให้ค่าสัญญาณที่ออกจาก 74LS08 มีค่าเหมือนกับวงจรถูกเปิดสัญญาณ แต่ถ้าหากว่ากดสวิทช์ 2 ตัวพร้อมกันค่าสัญญาณที่ออกจาก 74LS86 มีค่าเป็น 0 ทำให้สัญญาณที่ออกจาก 74LS08 มีค่าเป็น 0 เช่นเดียวกัน

วงจรแปลงสัญญาณการขับเคลื่อนมอเตอร์จาก LM629

ค่าสัญญาณที่ได้จาก LM629 จะประกอบไปด้วยค่าสัญญาณที่เป็นขนาดและทิศทาง จะผ่านวงจรขับสัญญาณ (26LS31) เพื่อลดการรบกวนจากการส่งสัญญาณ โดยทำการแปลงสัญญาณกลับด้วยวงจรรับสัญญาณ (26LS32) ซึ่งค่าของสัญญาณที่ผ่านการแปลงแล้วจะถูกแบ่งให้ชัดเจน โดยใช้ไอซีเบอร์ 74HC04 ซึ่งเป็นนอทเกต (NOT) และ ไอซีเบอร์ 74HC08 ซึ่งเป็นแอนเกต (AND) (เนื่องจากสัญญาณที่ LM629 สร้างออกมาเพื่อขับเคลื่อนมอเตอร์มีความถี่สูง ทำให้ไอซีตระกูล 74 ที่เป็นแบบ LS ทำงานไม่ทัน จึงเลือกใช้ไอซีที่เป็นแบบ HC)



รูปที่ 3.6 วงจรแปลงสัญญาณการขับเคลื่อนมอเตอร์จาก LM629



Sa, Sb คือสัญญาณทิศทางของการนับของมอเตอร์

Ma*, Mb* คือสัญญาณขนาดของการนับของมอเตอร์ที่ยังไม่ได้ผ่าน limit switch

รูปที่ 3.7 วงจรเด็กโหมคการทำงาน

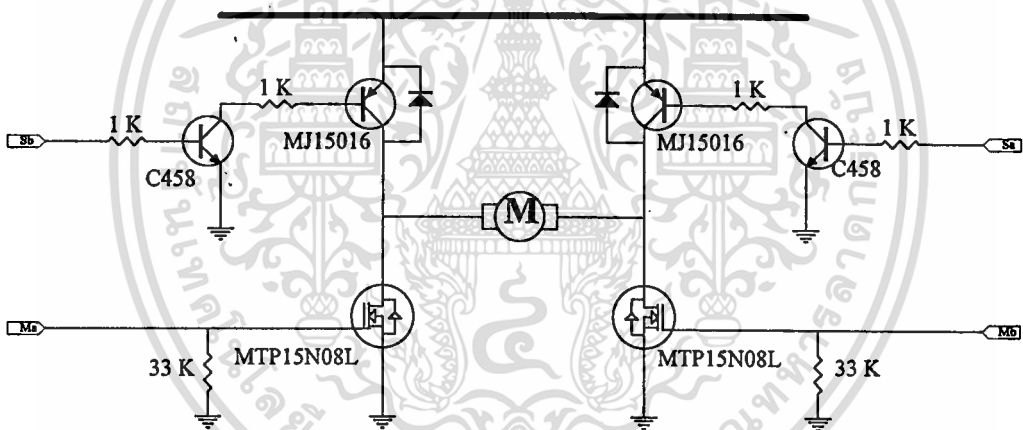
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าสัญญาณที่เป็นทิศทางจะเป็นตัวกำหนดให้ค่าสัญญาณที่เป็นขนาดออกจาก 74LS08 ในขาใด ขกตัวอย่างเช่น

เมื่อค่าสัญญาณทิศทางมีค่าเป็น 0 จะส่งผลทำให้สัญญาณที่ออกจาก 74HC04 ในขาที่ 8 และ 6 เป็น 0 และ 1 ตามลำดับ หากมีการส่งสัญญาณการขั้วมอเตอร์ค่าสัญญาณที่เป็นขนาดจะมีค่าเป็นสัญญาณพัลส์ ค่าสัญญาณที่ออกจาก 74HC08 ในขาที่ 8 และขาที่ 6 จะมีค่าเป็น 0 และพัลส์ตามลำดับ แต่ถ้าค่าสัญญาณทิศทางมีค่าเป็น 1 ค่าสัญญาณที่ออกจาก 74HC08 ในขาที่ 8 และขาที่ 6 จะมีค่าเป็น พัลส์ และ 0 ตามลำดับ

3.2.2 วงจรขับเคลื่อนมอเตอร์

เป็นวงจรขับเคลื่อนมอเตอร์ 2 ทิศทาง โดยนำมอสเฟตและทรานซิสเตอร์มาทำงานร่วมกันดังรูปที่ 3.8 และ 3.9

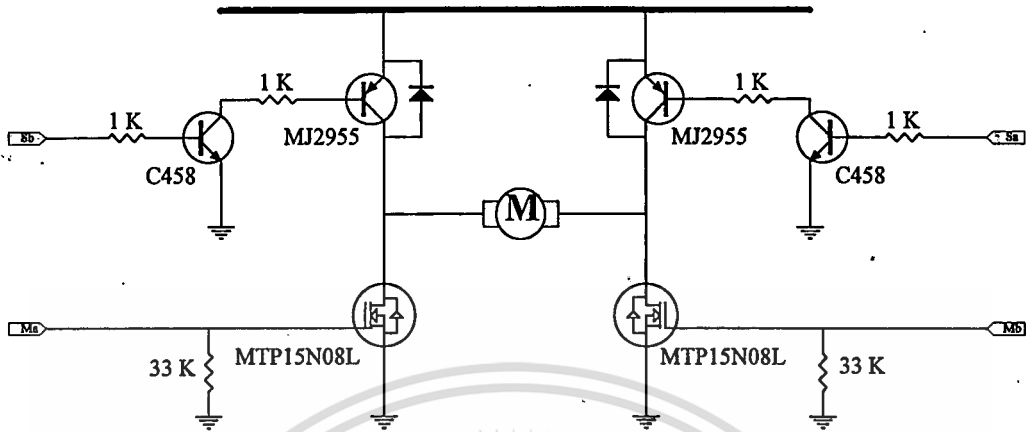


รูปที่ 3.8 วงจรขับเคลื่อนมอเตอร์ในแนวแกนนอน

สำหรับมอเตอร์ในแนวแกนนอนใช้ไฟเลี้ยง 42 โวลต์ เนื่องจากเป็นค่าแรงดันที่ทำให้มอเตอร์ทำงานได้ประสิทธิภาพดีที่สุด จากการทดสอบ Step Response และใช้ทรานซิสเตอร์เบอร์ MJ15016 รองรับแรงดันได้ 120 โวลต์ และรองรับกระแสได้ 15 แอมป์

ส่วนค่าของมอสเฟตเบอร์ MTP15N08L ซึ่งรองรับแรงดันได้ 80 โวลต์ และรองรับกระแสได้ 15 แอมป์ โดยจากการทดลองที่ผ่านมาสามารถทำการขับเคลื่อนได้โดยปกติ ทั้งในแนวแกนนอนและแนวแกนตั้ง

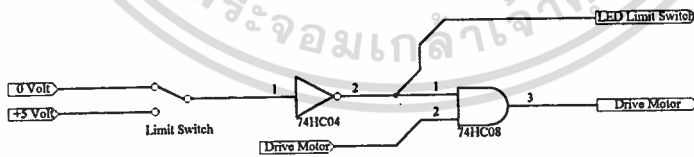
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรขับมอเตอร์ในแนวแกนตั้ง

ในแนวแกนตั้งใช้ทรานซิสเตอร์เบอร์ MJE2955 รองรับแรงดันได้ 60 โวลต์ และรองรับกระแสได้ 10 แอมป์ และใช้มอสเฟตเบอร์เดียวกัน

3.2.3 วงจรป้องกันการชน



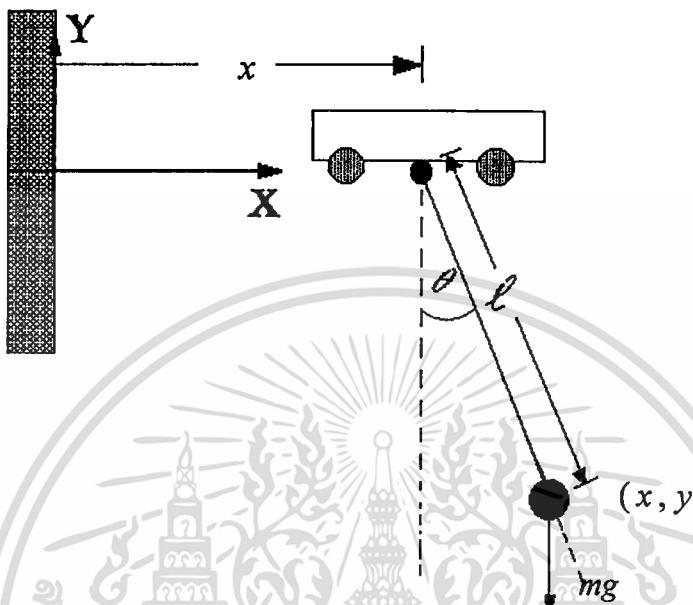
รูปที่ 3.10 วงจรป้องกันการชน

ใช้ 74HC08 เป็นตัวกำหนดว่าจะยอมให้สัญญาณผ่านไปยังวงจรขับมอเตอร์หรือไม่ โดยในสถานะปกติให้สัญญาณที่ออกจากลิมิตสวิทช์เป็น 0 เมื่อผ่าน 74HC04 จะกลายเป็น 1 ทำให้ 74HC08 ยอมให้ค่าสัญญาณผ่านไปขับมอเตอร์ แต่ถ้าหากว่าเกิดการชน สัญญาณที่ออกจากลิมิต สวิทช์จะเป็น 1 เมื่อผ่าน 74HC04 จะกลายเป็น 0 ทำให้ค่าที่ออกจาก 74HC08 เป็น 0 ด้วย

บทที่ 4

การหาแบบจำลองทางคณิตศาสตร์ของระบบ

4.1 แบบจำลองทางคณิตศาสตร์ของภาวะ



รูปที่ 4.1 รูปแสดงแบบจำลองของคัตวเครน

x_1 : ระยะทางของเครนที่เคลื่อนที่ไปได้ตามแกนนอน เมื่อเทียบกับจุดอ้างอิง (m)

y : ระยะทางเครนที่เคลื่อนที่ไปได้ตามแกนตั้ง เมื่อเทียบกับจุดอ้างอิง (m)

m : มวลของภาวะ (kg)

x : ระยะทางของภาวะที่เคลื่อนที่ไปตามแกนนอน เมื่อเทียบกับจุดอ้างอิง (m)

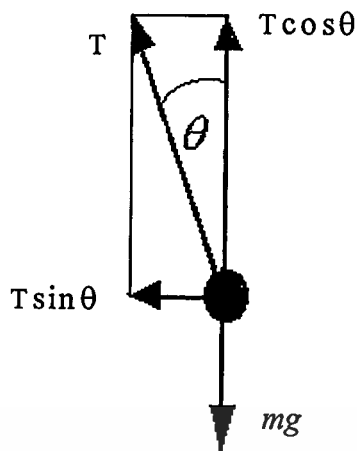
l : ความยาวของเชือกที่แขวนภาวะ (m)

g : ความเร่งโน้มถ่วงของโลก (m/s^2)

จะได้ความสัมพันธ์

$$x = x_1 + l \sin\theta \quad \text{----- (4.1)}$$

$$y = -l \cos\theta \quad \text{----- (4.2)}$$



รูปที่ 4.2 รูปแสดงแบบจำลองของภาวะ

จากรูปที่ 4.2 เมื่อ T : แรงตึงเชือก (N)
จะได้ความสัมพันธ์

$$m\ddot{x} = T \sin \theta \quad \text{----- (4.3)}$$

$$m\ddot{y} = T \cos \theta - mg \quad \text{----- (4.4)}$$

และจากสมการ (4.3) และ (4.4) ได้

$$\ddot{x} \cos \theta + \ddot{y} \sin \theta = -g \sin \theta \quad \text{----- (4.5)}$$

สมมติว่า ℓ มีความยาวคงที่และหาอนุพันธ์อันดับสองของสมการ (4.1) และ (4.2) ได้

$$\ddot{x} = \ddot{x}_1 - \dot{\theta}^2 \sin \theta + \ell \ddot{\theta} \cos \theta \quad \text{----- (4.6)}$$

$$\ddot{y} = \ell \dot{\theta}^2 \cos \theta + \ell \ddot{\theta} \sin \theta \quad \text{----- (4.7)}$$

แทนค่าจากสมการ (4.6) และ (4.7) ลงในสมการ (4.5) ได้

$$\begin{aligned} \ddot{x}_1 \cos \theta + \ell \ddot{\theta} &= -g \sin \theta \\ \ddot{\theta} &= -\left(\frac{g}{\ell}\right) \sin \theta - \left(\frac{\ddot{x}_1}{\ell}\right) \cos \theta \quad \text{----- (4.8)} \end{aligned}$$

จากสมการ (4.8) สมมติให้มีการแกว่งของมุม θ มีค่าน้อยๆ ทำให้เราสามารถประมาณค่า $\sin \theta$ เป็น θ และประมาณ $\cos \theta$ เป็น 1 ทำให้ได้สมการที่มีความสัมพันธ์เป็นเชิงเส้น ดังนี้

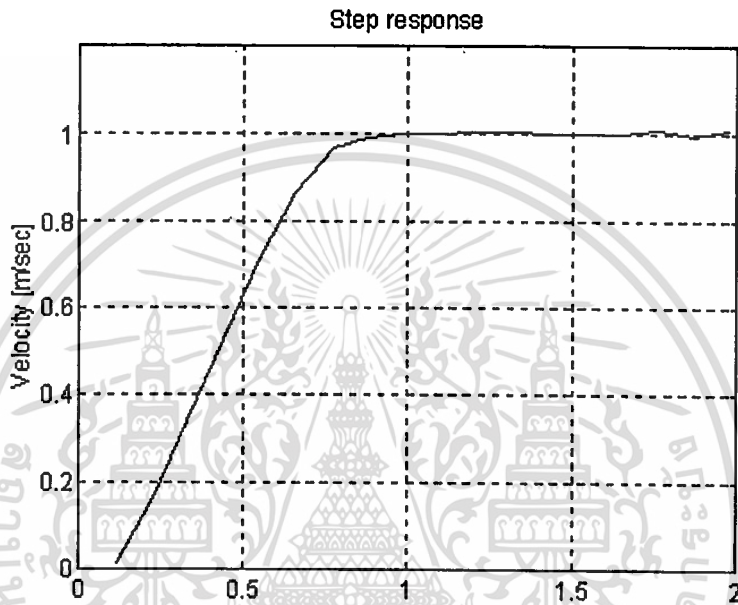
$$\ddot{\theta} = -\left(\frac{g}{\ell}\right) \theta - \left(\frac{\ddot{x}_1}{\ell}\right) \quad \text{----- (4.9)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การหาแบบจำลองทางคณิตศาสตร์ของตัวรถเครน

ระบบรถเครนประกอบไปด้วย วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และวงจรควบคุมมอเตอร์ วงจรขับเคลื่อนมอเตอร์ ชุดเฟือง สายพาน และตัวรถเครนที่เคลื่อนที่บนราง จะทำการหาแบบจำลองทางคณิตศาสตร์ของระบบรถเครนด้วยการพิจารณาผลตอบสนอง step input 1 เมตรต่อวินาที โดยสามารถหาได้จากโปรแกรม CMC3.EXE (ดูรายละเอียด ในภาคผนวก ก)

จากการทดสอบ step response จะได้กราฟรูปที่ 4.3



รูปที่ 4.3 กราฟทดสอบ step response

จากรูปที่ 4.3 สามารถพิจารณาระบบเป็นระบบอันดับสอง

จากการพิจารณากราฟรูปที่ 4.3 จะได้ settling time (T_s) = 0.75 วินาที

จากสมการ (4.10)¹

$$T_s = \frac{4}{\zeta \omega_n} \quad \text{-----(4.10)}$$

และจากราฟรูปที่ 4.3 สามารถประมาณค่า damping ratio (ζ) ได้ $\zeta = 1.0$

จากสมการ (4.10) จะได้

$$\omega_n = 5.33 \quad \text{rad/sec}$$

¹ อ้างอิง : KATSUHIKO OGATA , Modern Control Engineering, 2nd Edition, Chapter 4 Transient-Response

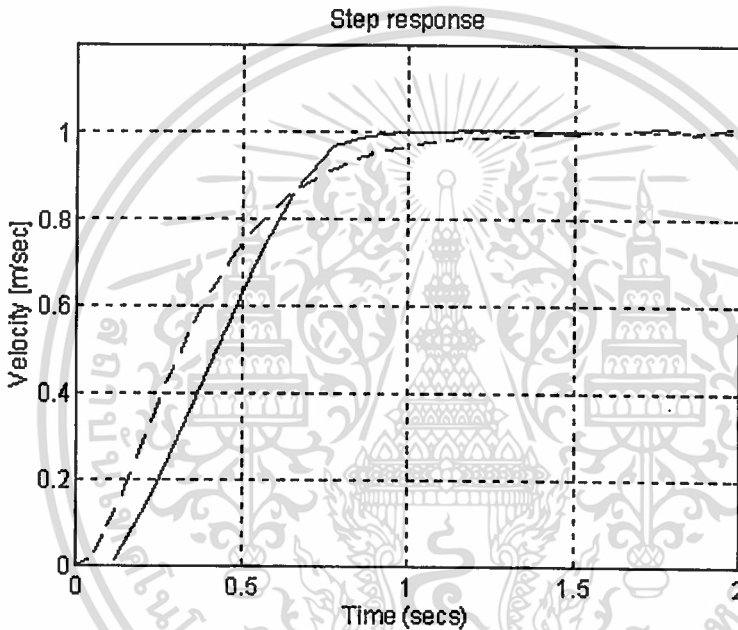
จากรูปแบบฟังก์ชันถ่ายโอนของระบบอันดับสองคือ

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \text{-----(4.11)}$$

แทนค่า ω_n และ ζ ที่ได้ลงในสมการ (4.11) จะได้

$$\frac{C(s)}{R(s)} = \frac{28.4444}{s^2 + 10.6667s + 28.4444} \quad \text{-----(4.12)}$$

แล้วใช้โปรแกรม MATLAB ทำการทดสอบ Step Response ในสมการที่ (4.12) (ดู ภาคผนวก ข.1) จะได้ผลดังรูปที่ 4.4



รูปที่ 4.4 กราฟ Step Response

- กราฟจากการ Simulate สมการที่ (4.12) ด้วย MATLAB
- กราฟจากการทดลอง

พิจารณารูปที่ 4.4 พบว่าค่า $\zeta = 1.0$ ที่กำหนดให้ ทำให้กราฟจากการ Simulate ใกล้เคียงกับกราฟที่ได้จากระบบจริง

พิจารณาสมการที่ (4.12) เปรียบเทียบกับระบบเครนจะได้

$C(s)$: ความเร็วของตัวรถเครน

$R(s)$: คำสั่งความเร็วที่ส่งมาจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$(s^2 + 2\zeta\omega_n s + \omega_n^2)C(s) = \omega_n^2 R(s) \quad \text{-----}(4.13)$$

ทำแปลงอินเวอร์สลาปลาซ สมการ(4.11)จะได้

$$\ddot{c}(t) + 2\zeta\omega_n \dot{c}(t) + \omega_n^2 c(t) = \omega_n^2 r(t) \quad \text{-----}(4.14)$$

จากหัวข้อที่ 4.1 จะพบว่า

$$c(t) = \dot{x}(t) \quad \text{-----}(4.15)$$

และให้

$$r(t) = u(t) \quad \text{-----}(4.16)$$

จาก(4.14) (4.15) และ (4.16) จะได้

$$\ddot{x}(t) = -2\zeta\omega_n \dot{x}(t) - \omega_n^2 x(t) + \omega_n^2 u(t) \quad \text{-----}(4.17)$$

4.3 การหาแบบจำลองทางคณิตศาสตร์รวมทั้งระบบ ในรูปสมการสเตทในระบบเวลาต่อเนื่อง

จากสมการที่(4.9) และ (4.17) สามารถเขียนสมการสถานะของระบบในรูป

$$\dot{x} = Ax + Bu \quad \text{ได้ดังนี้}$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{g}{l} & 0 & 0 & 0 & -\frac{1}{l} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix} u \quad \text{-----}(4.18)$$

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

$$\text{แทนค่า } \omega_n = 5.33 \quad \text{rad/sec}$$

$$\zeta = 1.0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$l = 0.8975 \text{ m}$$

$$g = 9.81 \text{ m/sec}^2 \quad \text{ในสมการ (4.7) จะได้}$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -10.9304 & 0 & 0 & 0 & -1.1142 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -28.4444 & -10.6667 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 28.4444 \end{bmatrix} u$$

ดังนั้น

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -10.9304 & 0 & 0 & 0 & -1.1142 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -28.4444 & -10.6667 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 28.4444 \end{bmatrix}$$

4.4 การแปลงสมการสเตทในระบบเวลาต่อเนื่องให้อยู่ในรูปสมการสเตทในระบบเวลาดิสครีต

จากรูปแบบสมการสเตทในระบบเวลาต่อเนื่อง คือ

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

สามารถทำการแปลงให้อยู่ในระบบสมการสเตทในระบบเวลาดิสครีตได้เป็น

$$x(k+1) = Gx(k) + Hu(k)$$

โดยใช้ความสัมพันธ์²

$$G(T) = e^{AT} \quad \text{-----(4.19)}$$

$$H(T) = \left(\int_0^T e^{A(T-t)} dt \right) B \quad \text{-----(4.20)}$$

เมื่อ T คือ คาบเวลาในการซิกคตัวอย่างของระบบ กำหนดให้ $T = 0.1$ วินาที

สามารถแก้สมการ (4.19) และ (4.20) โดยใช้โปรแกรม MATLAB (ดู ภาคผนวก ข.2)

² อ้างอิง : รศ. วิพันธ์ ปรีชาพานิช, การวิเคราะห์ระบบควบคุมเวลาดิสครีต, บทที่ 3 สมการสเตทดิสครีต, หน้า 194
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ยืมได้หนังสือฉบับนี้แล้ว กรุณา
ไม่ว่ากรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้

$$G = \begin{bmatrix} 0.9454 & 0.0982 & 0 & 0.0041 & -0.0039 \\ -1.0732 & 0.9458 & 0 & 0.1108 & -0.0639 \\ 0 & 0 & 1 & 0.0963 & 0.0035 \\ 0 & 0 & 0 & 0.8995 & 0.05787 \\ 0 & 0 & 0 & -1.6687 & 0.2738 \end{bmatrix}$$

$$H = \begin{bmatrix} -0.0041 \\ -0.1108 \\ 0.0037 \\ 0.1005 \\ 1.6687 \end{bmatrix} \text{-----(4.21)}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การควบคุมมุมการแกว่งและตำแหน่งของเครน

ในการควบคุมการเคลื่อนที่ของเครนเราต้องการควบคุมตัวแปรหรือเอาต์พุตของระบบถึง 2 ค่าคือ

1. ต้องการควบคุมตำแหน่งของเครน
2. ต้องการควบคุมการแกว่งของภาระให้หยุดนิ่ง

เราจึงเลือกวิธีการควบคุมโดยการป้อนกลับสแตทแบบสแตทเรกกูเรเตอร์ (State Regulator) ที่หาค่ากฎการควบคุมจากการนำสแตทที่ป้อนกลับมาคูณด้วยค่าเกนป้อนกลับ K (Feedback Gain, K) ซึ่งทำการออกแบบค่าเกนป้อนกลับ โดยวิธีการควบคุมออปติมัลแบบเรกกูเลเตอร์เชิงเส้นที่สถานะอยู่ตัว (Steady State Linear Quadratic Regulator หรือ LQR) ที่มีดัชนีการทำงานแบบควอดราติกเป็น (Quadratic Performance Index) เพื่อให้ได้ผลตอบสนองดีที่สุดภายใต้ค่าดัชนีการทำงานที่ตั้งไว้ แต่เนื่องจากการควบคุมแบบสแตทนี้ เราจำเป็นต้องเข้าถึงหรือรู้ค่าของสแตททุกๆตัวที่เราสนใจ ซึ่งประกอบไปด้วย

1. มุมการแกว่งของภาระ (θ)
2. ความเร็วเชิงมุมของภาระ ($\dot{\theta}$)
3. ตำแหน่งของเครน (x)
4. ความเร็วของเครน (\dot{x})
5. ความเร่งของเครน (\ddot{x})

ในทางปฏิบัติเราทำการวัดได้เพียง 2 สแตทเท่านั้นคือ มุมการแกว่ง และตำแหน่งของเครน ดังนั้นเราจึงจำเป็นต้องประมาณค่าสแตทที่เหลือจากค่าสแตทที่เรารู้ โดยเราใช้วิธีที่ ประมาณค่าสแตทโดยอาศัยหลักการฟิสิกส์

5.1 การควบคุมออปติมัล (Optimal Control)

การออกแบบระบบควบคุมออปติมัลคือการหากฎการควบคุม (Control Law) เพื่อกำหนดการตัดสินใจในการควบคุมในเวลาปัจจุบัน โดยมีเงื่อนไขบังคับบางประการเพื่อให้ระบบเบี่ยงเบนไปจากลักษณะที่ต้องการน้อยที่สุด ตัวที่วัดการทำงานของระบบที่ต้องการนี้เรียกว่า เกณฑ์การตัดสินใจ หรือดัชนีการทำงาน (Performance Index, J) ซึ่งเป็นฟังก์ชันที่เราเลือกให้เป็นตัวชี้ว่าการทำงานจริงของระบบใกล้เคียงกับการทำงาน ที่ต้องการมากน้อยเพียงใด ดังนั้นปัญหาการควบคุมออปติมัลจึงอยู่ที่ การเลือกดัชนีการทำงาน และกฎควบคุมออปติมัล (Optimal Control Law, u_k) เพื่อออกพิตไมซ์ฟังก์ชันดัชนีการทำงาน หรือส่วนใหญ่จะเป็นการหาค่ากฎการควบคุมที่ทำให้ค่าดัชนีการทำงานมี

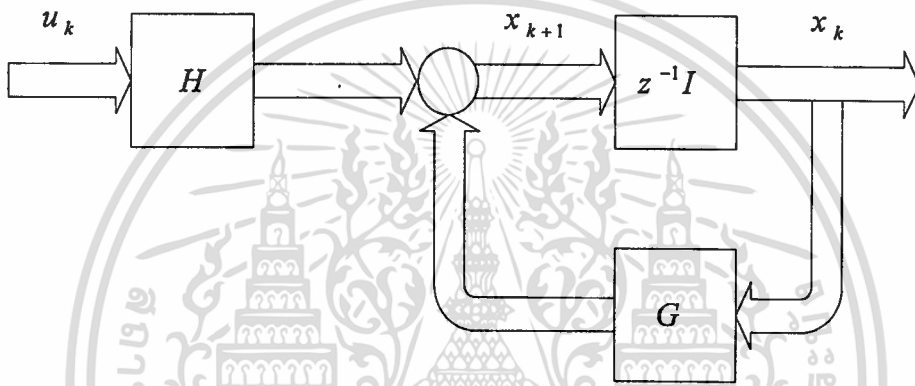
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าน้อยที่สุด (Minimization) นั้นเอง สำหรับการเลือกค่าดัชนีการทำงานนั้นขึ้นอยู่กับแต่ละปัญหา เราไม่สามารถกำหนดเกณฑ์ในการเลือกดัชนีการทำงานให้ตายตัวได้

กำหนดระบบที่สามารถควบคุมได้โดยสมบูรณ์

$$x_{k+1} = Gx_k + Hu_k \quad \text{-----}(5.1)$$

เมื่อ x_k เป็น n สเตทเวกเตอร์
 u_k เป็น r เวกเตอร์ควบคุม
 G เป็น $(n \times n)$ เมตริกซ์
 H เป็น $(n \times r)$ เมตริกซ์
 T หมายถึงทรานสโพล



รูปที่ 5.1 แสดงบล็อกไดอะแกรมของระบบเปิด

ในที่นี้เราเลือกใช้การควบคุมออปติมัลแบบเรกกูเลเตอร์เชิงเส้นที่สถานะอยู่ตัว (Steady State Linear Quadratic Regulator) ที่มีดัชนีการทำงานควอดราติกเป็น

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^T Q x_k + u_k^T R u_k] \quad \text{-----}(5.2)$$

Q เป็น $(n \times n)$ เมตริกซ์เฮอร์มิเซียน (Hermitian Matrix) หรือเมตริกซ์สมมาตรเลขจริง (Real Symmetric Matrix) positive definite หรือ positive semidefinite

R เป็น $(r \times r)$ เมตริกซ์เฮอร์มิเซียน (หรือเมตริกซ์สมมาตรเลขจริง) positive definite

เนื่องจากส่วนประกอบของระบบในทางปฏิบัติจะมีข้อจำกัดทางกายภาพ จึงต้องมีการกำหนดเงื่อนไขบังคับตัวแปรสเตท ตัวแปรควบคุม และสัญญาณควบคุม เพื่อจำกัดขนาดตัวแปรสเตทและตัวแปรควบคุม เมตริกซ์ Q และ R จะถูกเลือกเพื่อให้น้ำหนักความสำคัญของสเตทและสัญญาณอินพุตแต่ละตัวต่างๆกัน เมตริกซ์ Q จะถูกเลือกเพื่อจำกัดข้อผิดพลาด หรือขนาดของสเตทแต่ละตัว ยิ่งเราให้น้ำหนักความสำคัญกับสเตทไหนมาก (เมื่อเปรียบเทียบกับที่เราให้กับสเตทอื่น) ค่าผิดพลาดของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สแตทนั้นก็จะถูกจำกัดให้มีค่าน้อยลงเมื่อเทียบกับสแตทอื่น และเมตริกซ์ R จะถูกเลือกเพื่อจำกัดขนาดของสัญญาณควบคุมแต่ละตัว เช่นเดียวกันถ้าเราให้น้ำหนักความสำคัญกับอินพุตตัวไหนมาก (เมื่อเปรียบเทียบกับที่ให้อินพุตอื่น)

การหาค่ากฎควบคุมออปติคัลนั้นสามารถหาได้หลายวิธี ในที่นี้เราจะแสดงการหาค่ากฎการควบคุมออปติคัล โดยการใช้ฟังก์ชันลียาปูนอฟ (Liapunov Function)

5.1.1 การหากฎควบคุมออปติคัลสำหรับเรกกูเลเตอร์เชิงเส้นโดยใช้ฟังก์ชันลียาปูนอฟจากระบบที่สามารถควบคุมได้โดยสมบูรณ์

$$x_{k+1} = Gx_k + Hu_k$$

ต้องการหาเมตริกซ์เกน K สำหรับกฎควบคุมออปติคัล

$$u_k = -Kx_k \quad \text{-----}(5.3)$$

เมื่อ u_k คือการควบคุมออปติคัลที่ทำให้ดัชนีการทำงานควอดราติกต่อไปนี้น้อยที่สุด

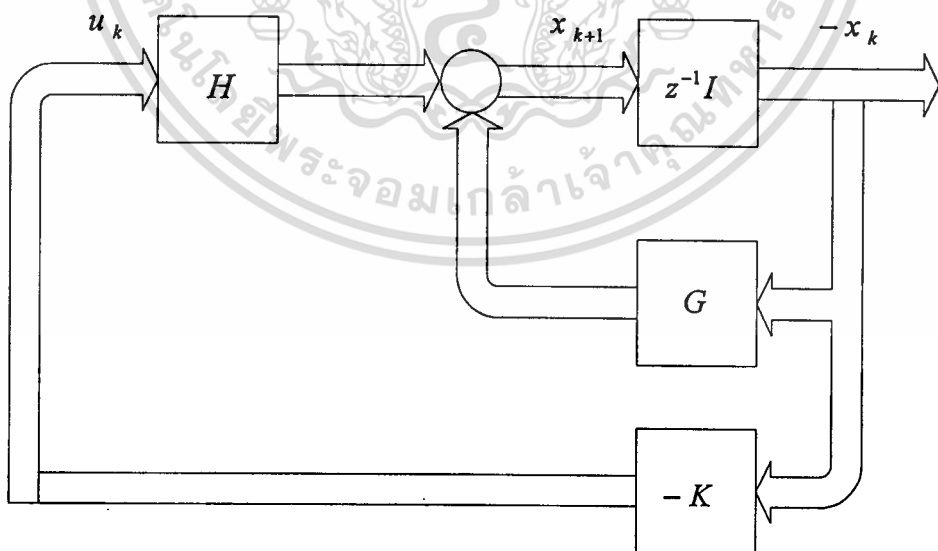
$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^T Q x_k + u_k^T R u_k]$$

เนื่องจาก Q เป็นเมตริกซ์เฮอริมิเทียน positive definite หรือ positive semidefinite และ R เป็นเมตริกซ์เฮอริมิเทียน positive definite ดังนั้นดัชนีการทำงาน J จึงเป็น positive definite

จากสมการของระบบวงเปิดด้วยการป้อนกลับ $u_k = -Kx_k$ ได้

$$x_{k+1} = (G - HK)x_k \quad \text{-----}(5.4)$$

และ $J = \frac{1}{2} \sum_{k=0}^{\infty} [x_k^T (Q + K^T R K) x_k] \quad \text{-----}(5.5)$



รูปที่ 5.2 แสดงบล็อกโคแอมแกรมของระบบเมื่อป้อนกลับด้วย $u_k = -Kx_k$

ในที่นี้ให้ $G - HK$ เป็นเมทริกซ์เสถียร ดังนั้นย่อมหาฟังก์ชันลิวาปูนอฟ ที่เป็น positive definite ได้และมีอนุพันธ์อันดับหนึ่ง เป็น negative definite ให้ฟังก์ชันลิวาปูนอฟคือ

$$V(x_k) = x_k^T P x_k$$

เมื่อ P เป็นเมทริกซ์เฮอร์มิเชียน (หรือสมมาตรเลขจริง) positive definite อนุพันธ์อันดับหนึ่งของ $V(x_k)$ คือ

$$\Delta V(x_k) = V(x_{k+1}) - V(x_k)$$

$$\Delta V(x_k) = x_{k+1}^T P x_{k+1} - x_k^T P x_k$$

จากสมการ (5.1)

$$\Delta V(x_k) = x_k^T \left[(G - HK)^T P (G - HK) - P \right] x_k$$

ซึ่งเป็น negative definite และ $x_k^T (Q + K^T R K) x_k$ เป็น positive definite เราจึงให้

$$x_k^T (Q + K^T R K) x_k = -x_k^T \left[(G - HK)^T P (G - HK) - P \right] x_k$$

สมมติว่าสมการดังกล่าวเป็นจริงสำหรับทุกค่าของ x_k จะได้

$$Q + K^T R K = -(G - HK)^T P (G - HK) + P$$

$$Q + G^T P G - P + K^T (R + H^T P H) K - (K^T H^T P G + G^T P H K) = 0$$

$$Q + G^T P G - P + \left[(R + H^T P H)^{\frac{1}{2}} K - (R + H^T P H)^{-\frac{1}{2}} H^T P G \right]^T \left[(R + H^T P H)^{\frac{1}{2}} K - (R + H^T P H)^{-\frac{1}{2}} H^T P G \right] - G^T P H (R + H^T P H)^{-1} H^T P G = 0 \quad \text{-----(5.6)}$$

การทำให้ค่าดัชนีการทำงานควอดเรติก น้อยที่สุดโดยคิดจากเกณฑ์กลับ K ก็คือ การเลือกค่าเกณฑ์ K ที่ทำให้ด้านซ้ายมือของสมการ(5.6)มีค่าน้อยที่สุด และเนื่องจากเทอม

$$\left[(R + H^T P H)^{\frac{1}{2}} K - (R + H^T P H)^{-\frac{1}{2}} H^T P G \right]^T \left[(R + H^T P H)^{\frac{1}{2}} K - (R + H^T P H)^{-\frac{1}{2}} H^T P G \right]$$

ไม่เป็นลบเสมอ ดังนั้นจึงจะมีค่าน้อยที่สุดเมื่อเป็นศูนย์เท่านั้น หรือเมื่อ

$$\begin{aligned} (R + H^T P H)^{\frac{1}{2}} K &= (R + H^T P H)^{-\frac{1}{2}} H^T P G \\ K &= (R + H^T P H)^{-1} H^T P G \quad \text{-----(5.7)} \end{aligned}$$

แทนค่าเกน K ที่ได้จากสมการ(5.7) ลงในสมการ(5.6) จะได้ค่า P ที่ทำให้ คำนีการทำให้ งานควบคุมคราดิกมีค่าน้อยที่สุด

$$P = Q + G^T P G - G^T P H (R + H^T P H)^{-1} H^T P G \quad (5.8)$$

สมการ(5.8)นี้เรียกว่า สมการรีคคาตี(Riccati Equation) สถานะอยู่ตัว

จากสมการ(5.7) และสมการ(5.8) เช่นเดียวกันจะเห็นว่าค่าของเมตริกซ์ P และค่าของ เมตริกซ์ K จะขึ้นอยู่กับค่าของเมตริกซ์กับค่าของเมตริกซ์ Q และเมตริกซ์ R ด้วย ดังนั้นถ้าสามารถ กำหนดเมตริกซ์ Q และเมตริกซ์ R ได้เหมาะสมกับระบบแล้ว เราก็จะได้เกนป้อนกลับ K ที่ทำให้ ระบบมีผลตอบสนองดีที่สุด

จากสมการ(5.7)และสมการ(5.8)ที่ได้แสดงให้เห็นว่า ถ้าระบบ สามารถควบคุมได้โดย สมบูรณ์ และเราสามารถวัดค่าจากทุกสเททได้ เมื่อเราหาระบบได้ และกำหนดเมตริกซ์ Q และ เมตริกซ์ R เพื่อให้น้ำหนักความสำคัญกับสเททอินพุตแต่ละตัวแล้ว เราจะสามารถแก้สมการ(5.8) หา ค่าเมตริกซ์ P ได้ และนำค่าเมตริกซ์ P ไปหาค่าเกน K ป้อนกลับสำหรับระบบเพื่อนำไปหาค่ากฎ ควบคุม ณ เวลาต่างๆได้

การหาค่าเกน K โดยวิธีการควบคุมออปติมัลแบบเรกกูเลเตอร์เชิงเส้นนี้อาจใช้โปรแกรม MATLAB ช่วยในการคำนวณได้(ดูภาคผนวก ข.3)

5.2 การนำระบบการควบคุมออปติมัลแบบเรกกูเลเตอร์เชิงเส้นมาใช้ในการควบคุมการเคลื่อนที่ของ เกรน

จากแบบจำลองทางคณิตศาสตร์ของระบบเกรน สมการ (4.21) ทดลองกำหนด เมตริกซ์ Q และ เมตริกซ์ R โดยให้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

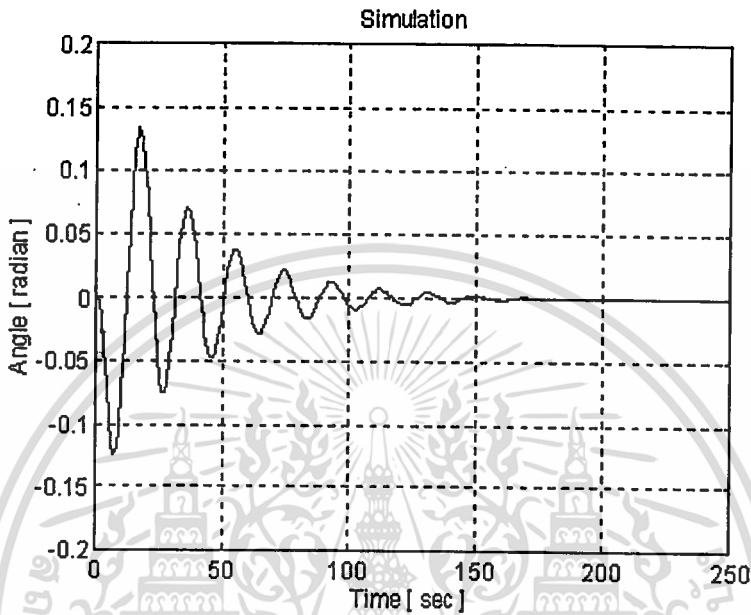
$$R = [100]$$

จากหัวข้อที่ 5.1 จะได้ เมตริกซ์ K ดังนี้

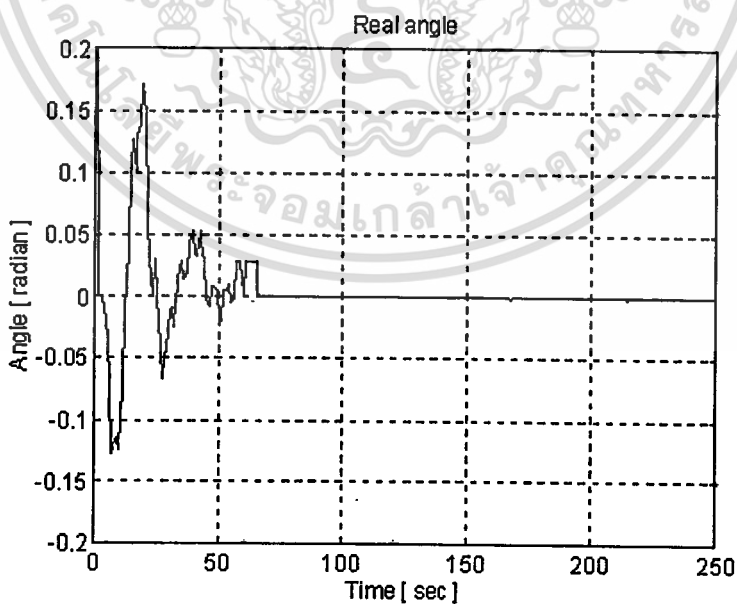
$$K = [-0.5435 \quad -0.2373 \quad 0.9062 \quad 0.3443 \quad 0.0605]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่าเมตริกซ์ K ที่ได้ไปทำการ Simulate โดยใช้โปรแกรม MATLAB (ดูภาคผนวก ข.4 และ ภาคผนวก ข.5) และทำการทดลองจริงกับระบบเครนโดยใช้โปรแกรม CMC3.EXE (ดูภาคผนวก ก) ได้ผลดังรูปต่อไปนี้

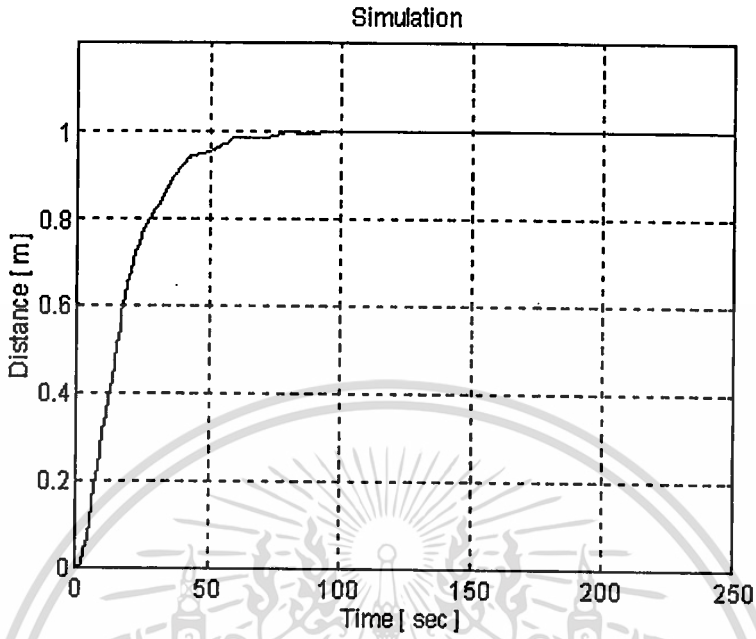


รูปที่ 5.3 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

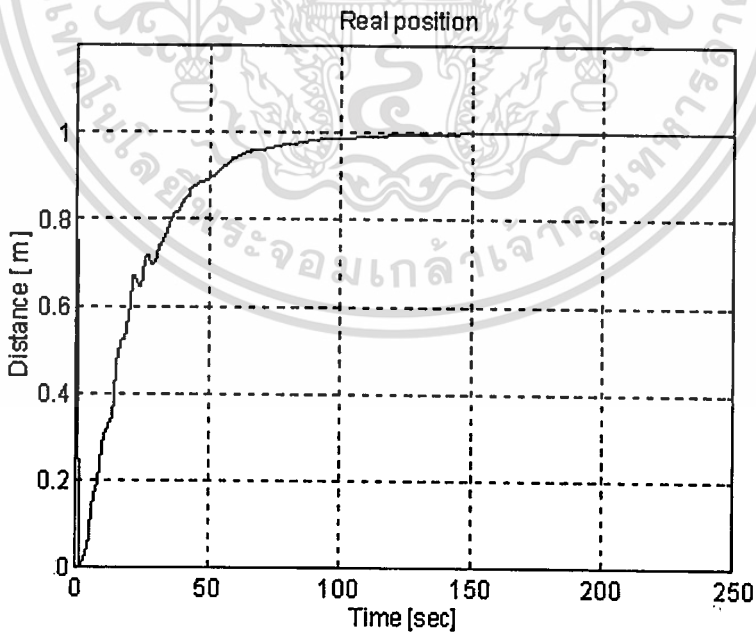


รูปที่ 5.4 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 5.6 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ผลการทดลอง

จะพิจารณาความยาวเชือกของภาระ 3 ระดับคือ 41.25cm 68.85 cm และ 89.75 cm โดยทำการปรับค่าเมตริกซ์ Q และเมตริกซ์ R ในแต่ละความยาวเชือก จะได้ค่า K และ ไอเกนแวลู (E)

6.1 การทดลองที่ความยาวเชือก 41.25 cm

การทดลองที่ 6.1.1 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

ได้ค่า $K = [-0.0300 \quad -0.1475 \quad 0.6472 \quad 0.2112 \quad 0.0563]$

$E = 0.5243$

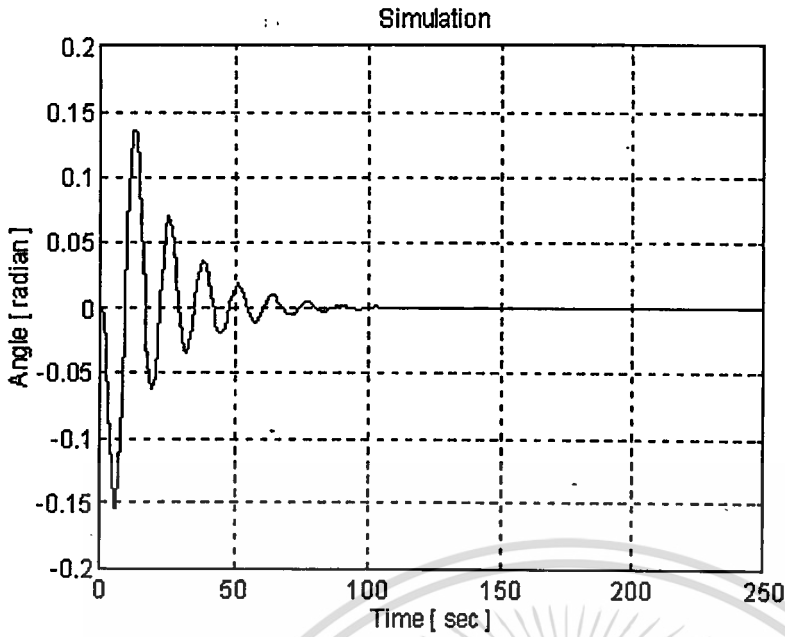
0.6538

0.9305

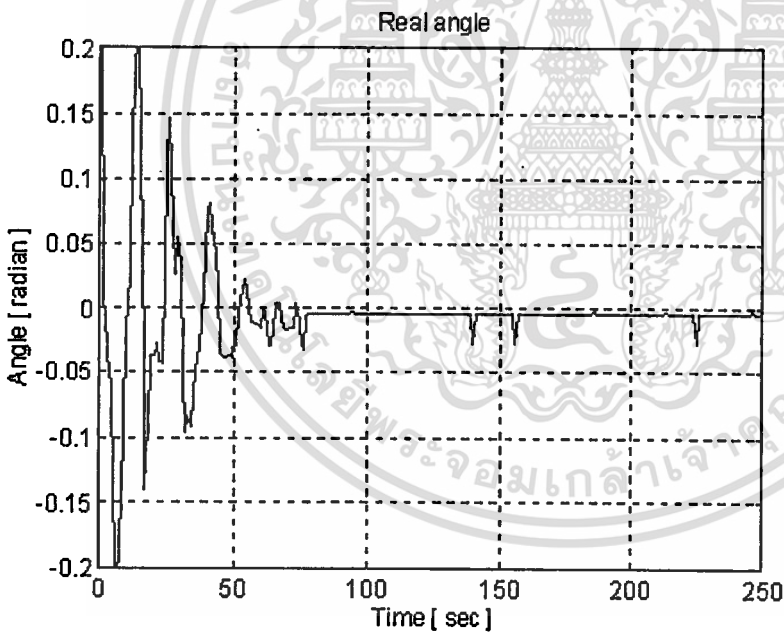
$0.8393 + 0.4465i$

$0.8393 - 0.4465i$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.1 และ 6.3 และผลจากการทดลองดังรูปที่ 6.2 และ 6.4

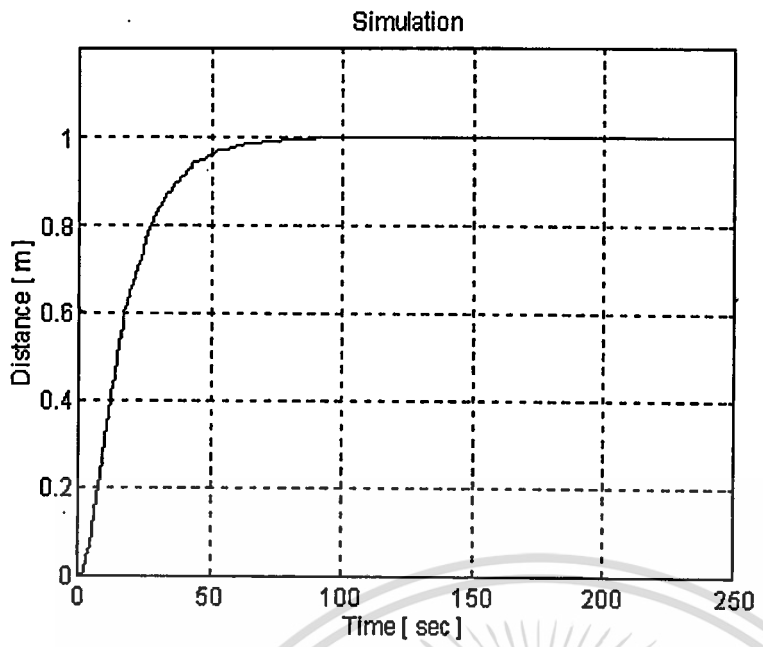


รูปที่ 6.1 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

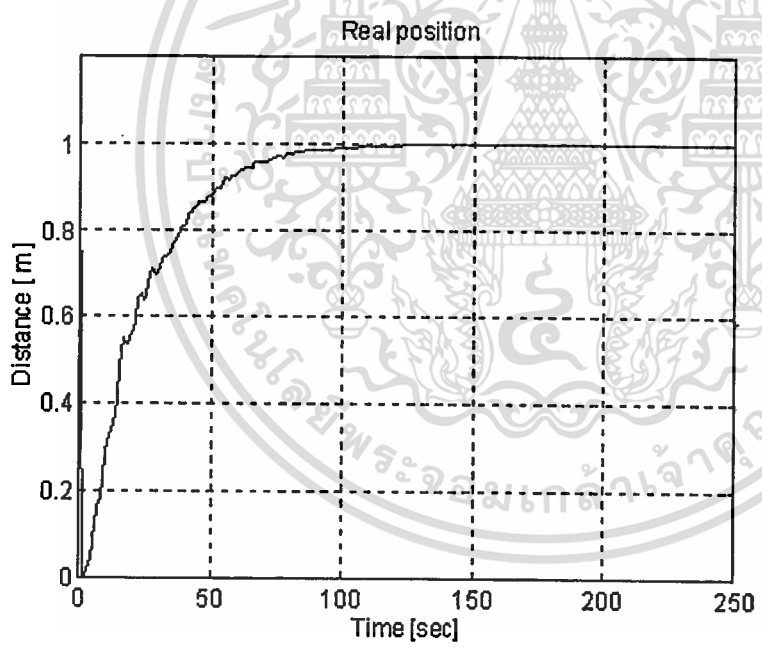


รูปที่ 6.2 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.3 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.4 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.1.2 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

$$\text{ได้ค่า } K = [0.0050 \quad -0.0788 \quad 0.6616 \quad 0.2231 \quad 0.0422]$$

$$E = 0.5283$$

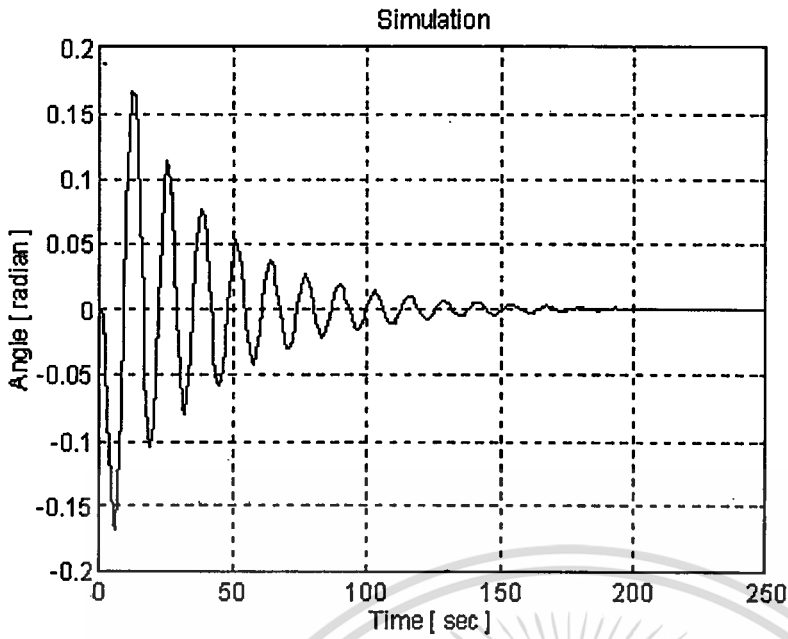
$$0.6461$$

$$0.9306$$

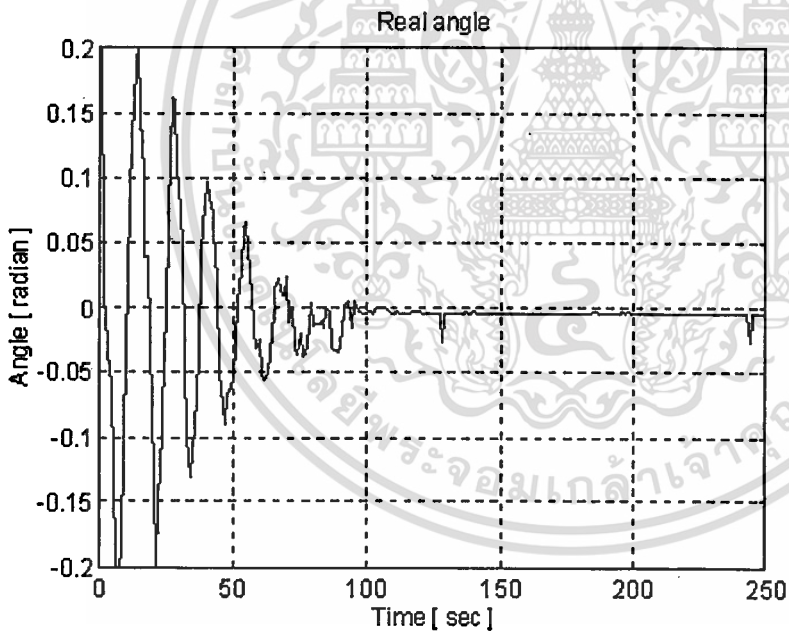
$$0.8606 + 0.4560i$$

$$0.8606 - 0.4560i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.5 และ 6.7 และผลจากการทดลองดังรูปที่ 6.6 และ 6.8

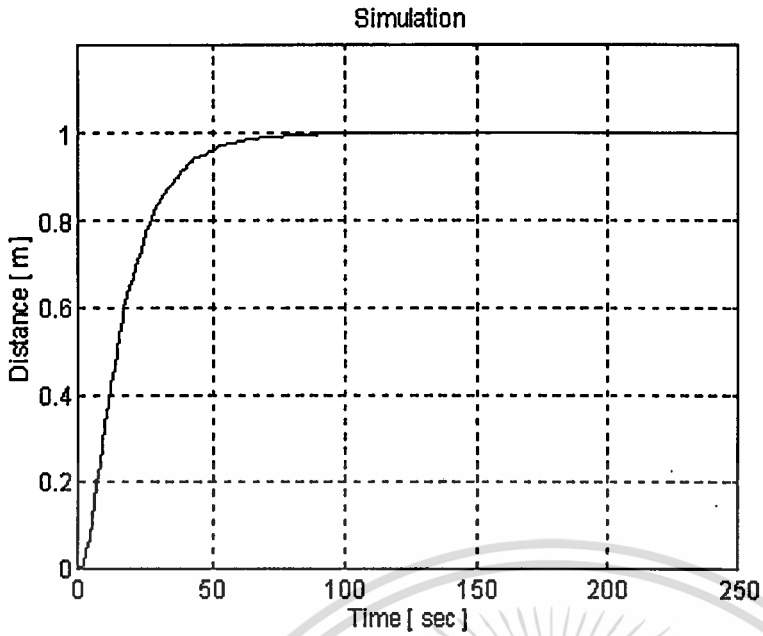


รูปที่ 6.5 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

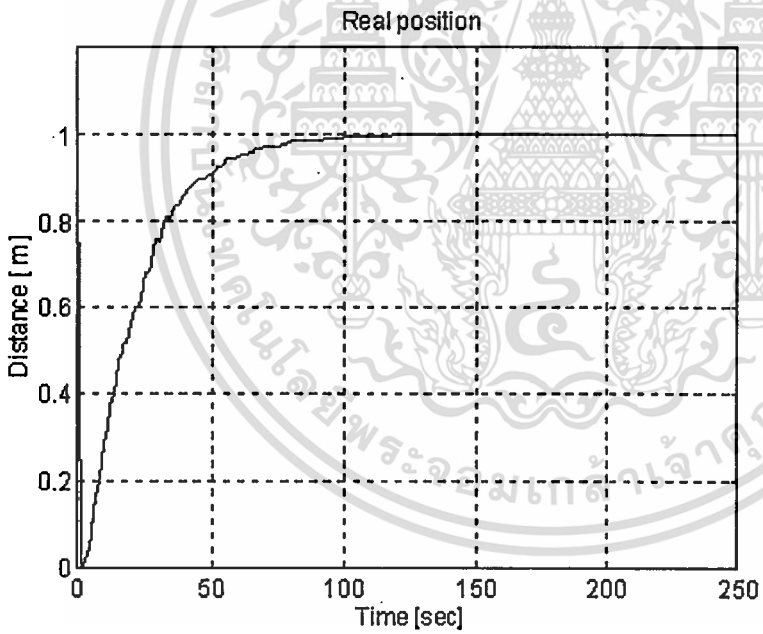


รูปที่ 6.6 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.7 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.8 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.1.3 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

$$\text{ได้ค่า } K = [0.0344 \quad -0.1506 \quad 0.2090 \quad 0.0353 \quad 0.0425]$$

$$E = 0.5259$$

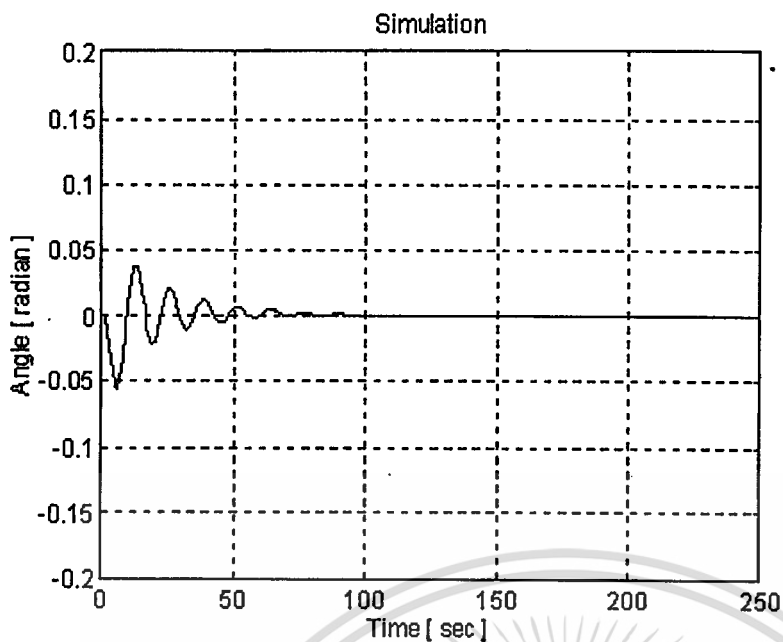
$$0.6471$$

$$0.8392 + 0.4465i$$

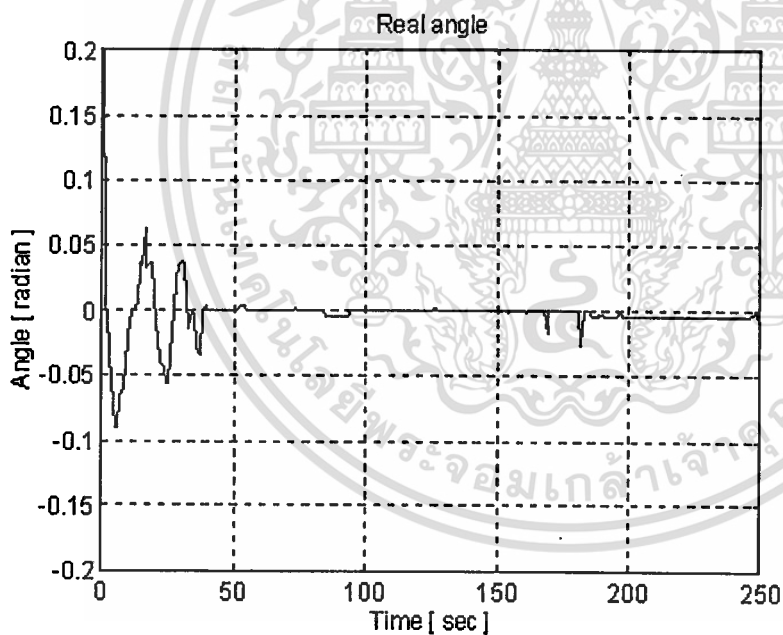
$$0.8392 - 0.4465i$$

$$0.9779$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.9 และ 6.11 และ
ผลจากการทดลองดังรูปที่ 6.10 และ 6.12

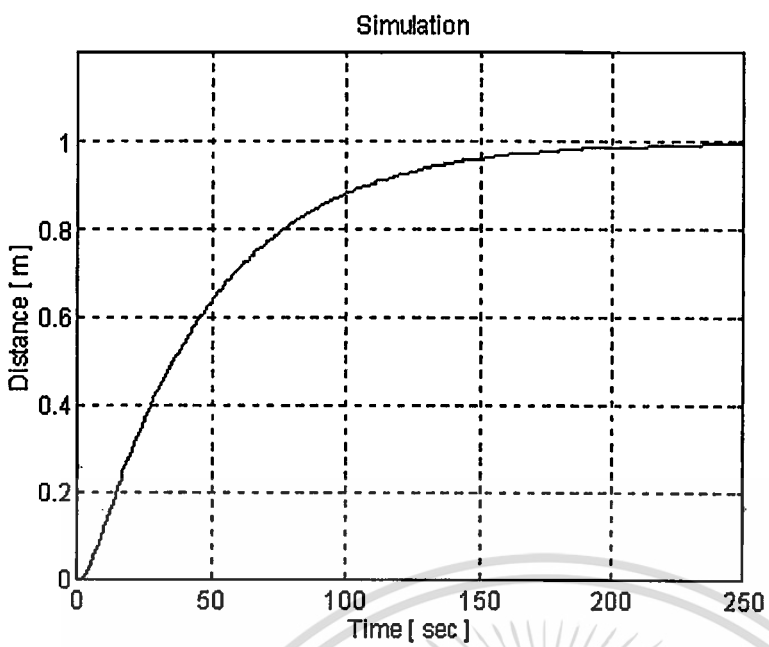


รูปที่ 6.9 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

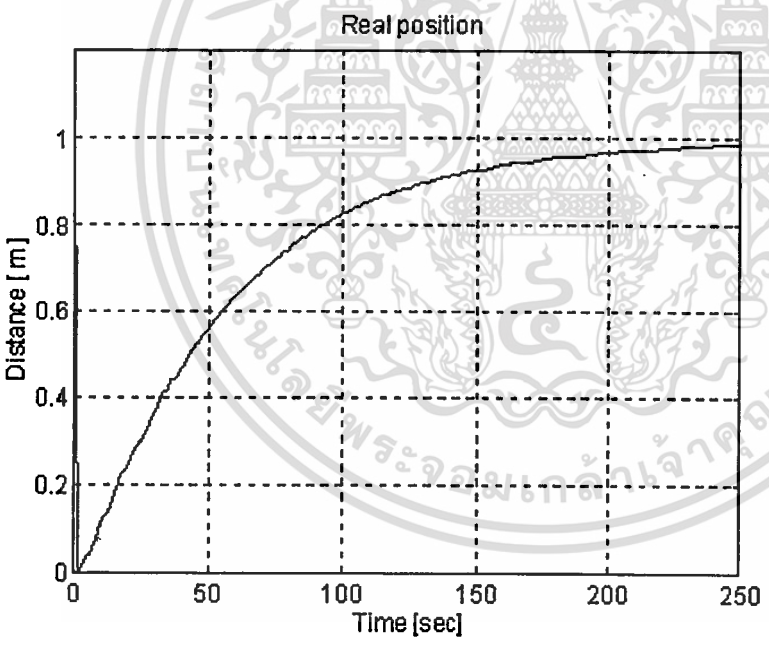


รูปที่ 6.10 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.11 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.12 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.1.4 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [50]$$

$$\text{ค่า } K = [-0.2562 \quad -0.2648 \quad 1.1832 \quad 0.4219 \quad 0.1040]$$

$$E = 0.4650$$

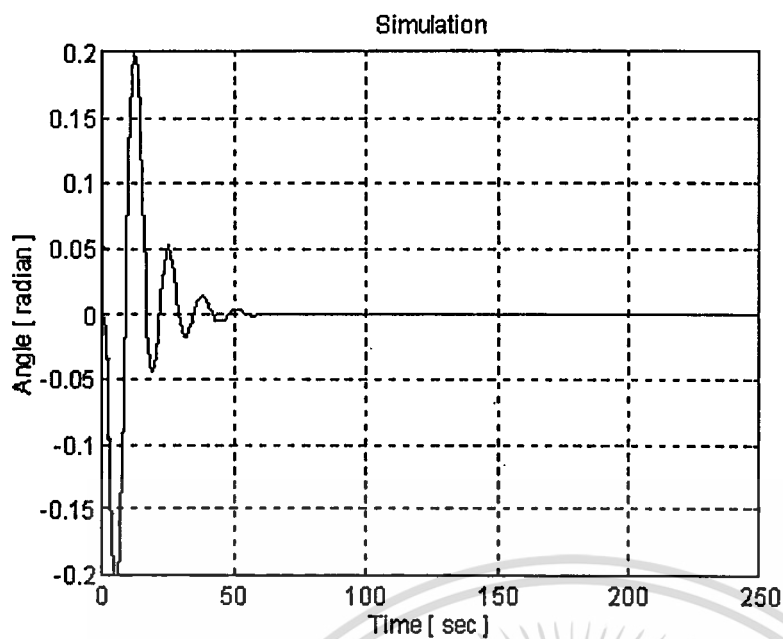
$$0.7401$$

$$0.8487$$

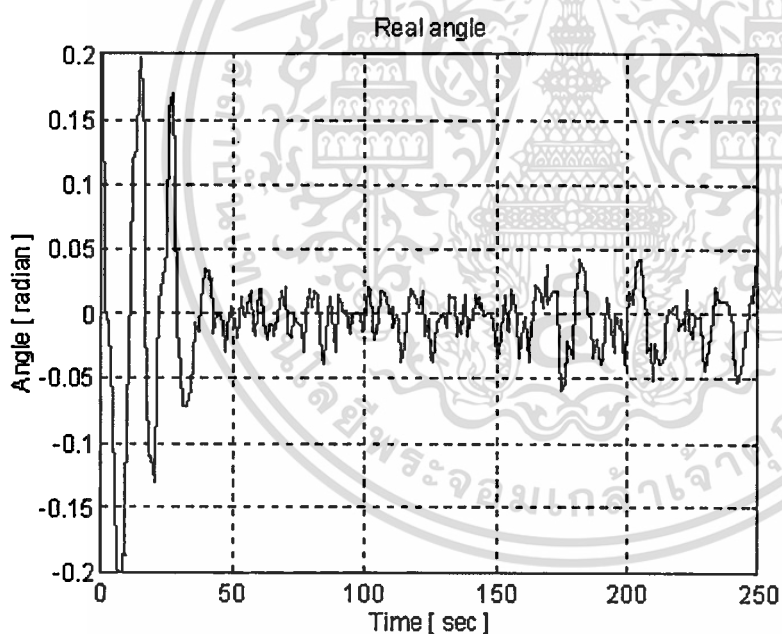
$$0.8005 + 0.4291i$$

$$0.8005 - 0.4291i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.13 และ 6.15 และผลจากการทดลองดังรูปที่ 6.14 และ 6.16

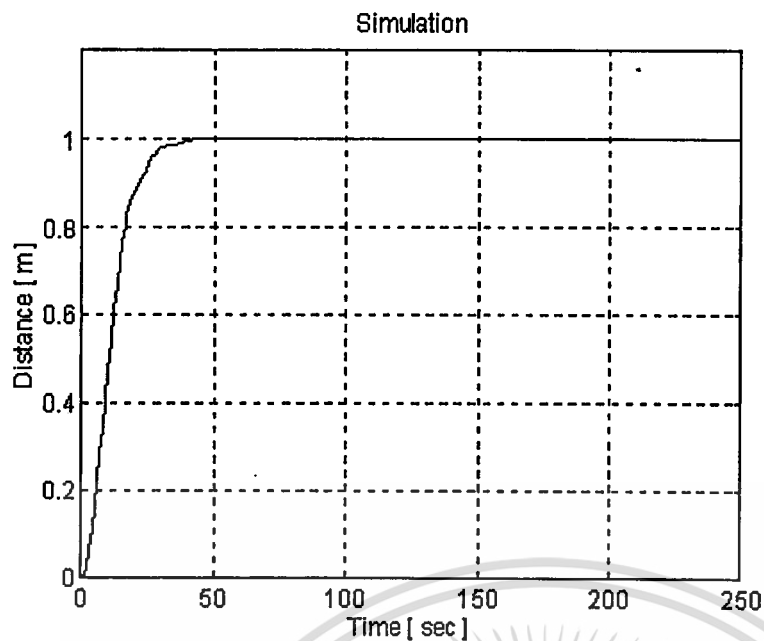


รูปที่ 6.13 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

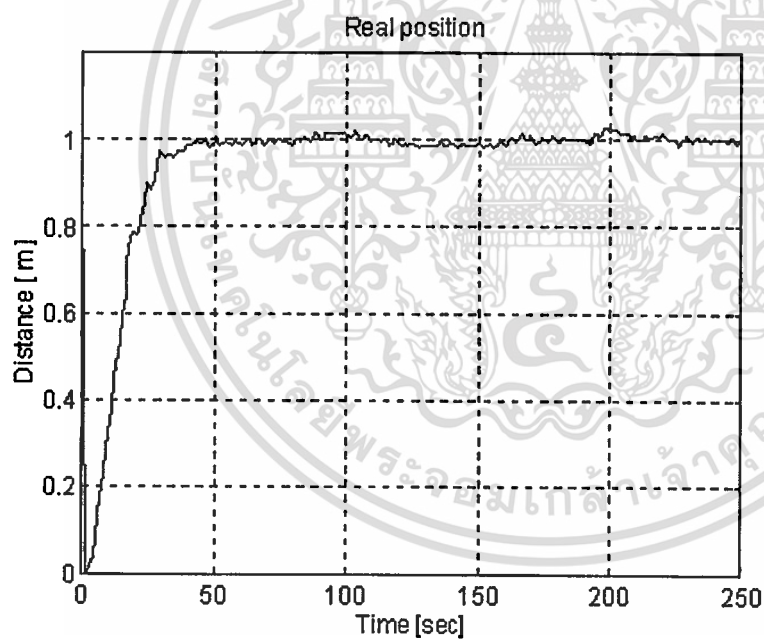


รูปที่ 6.14 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.15 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.16 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.1.5 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [20]$$

$$\text{ได้ค่า } K = [-0.6014 \quad -0.3649 \quad 1.6882 \quad 0.6526 \quad 0.1489]$$

$$E = 0.4004$$

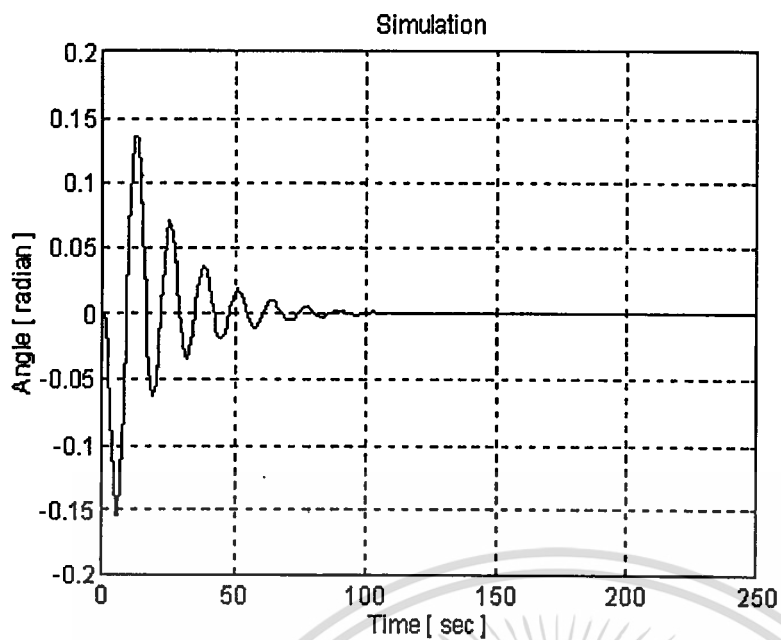
$$0.7992 + 0.0950i$$

$$0.7992 - 0.0950i$$

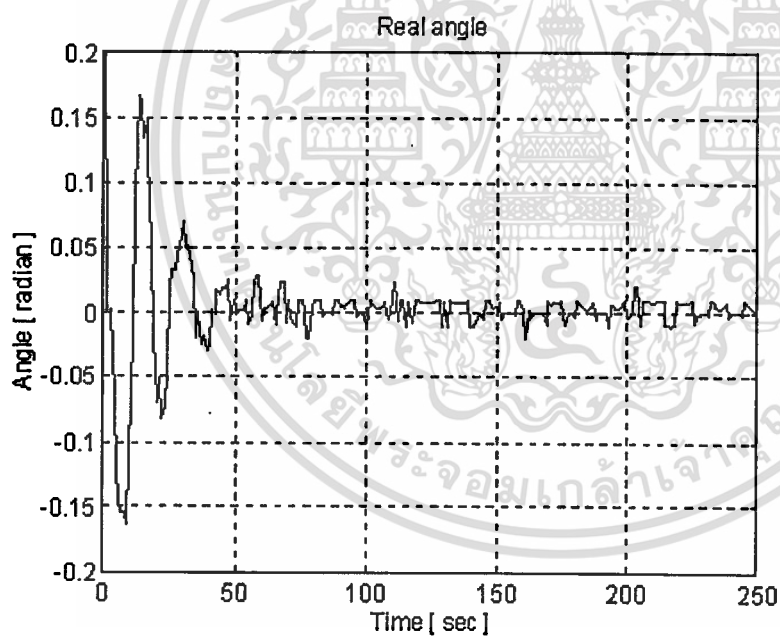
$$0.7645 + 0.4146i$$

$$0.7645 - 0.4146i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.17 และ 6.19 และผลจากการทดลองดังรูปที่ 6.18 และ 6.20

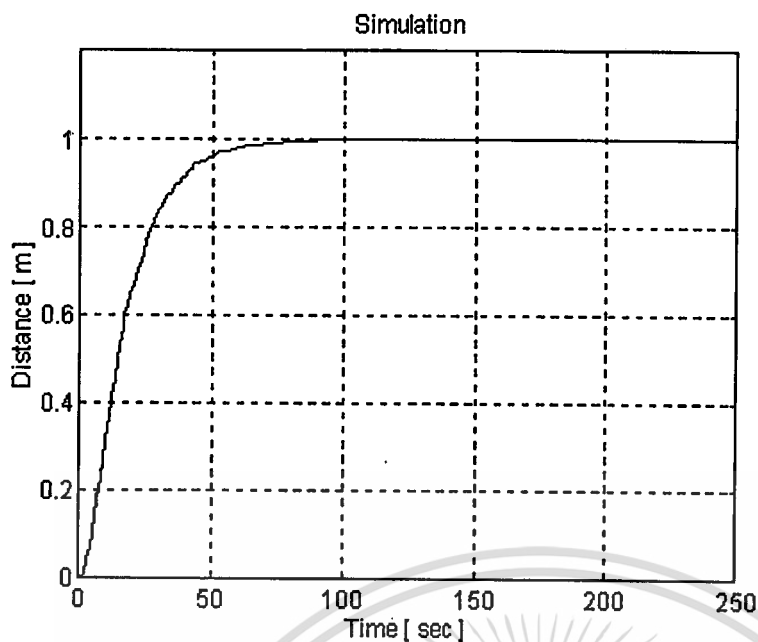


รูปที่ 6.17 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

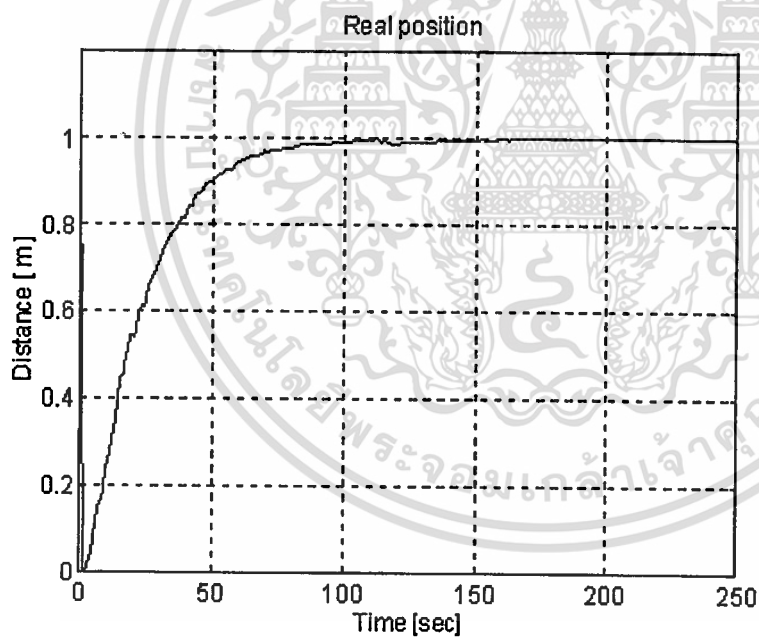


รูปที่ 6.18 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.19 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.20 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การทดลองที่ความยาวเชือก 68.85 cm

การทดลองที่ 6.2.1 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

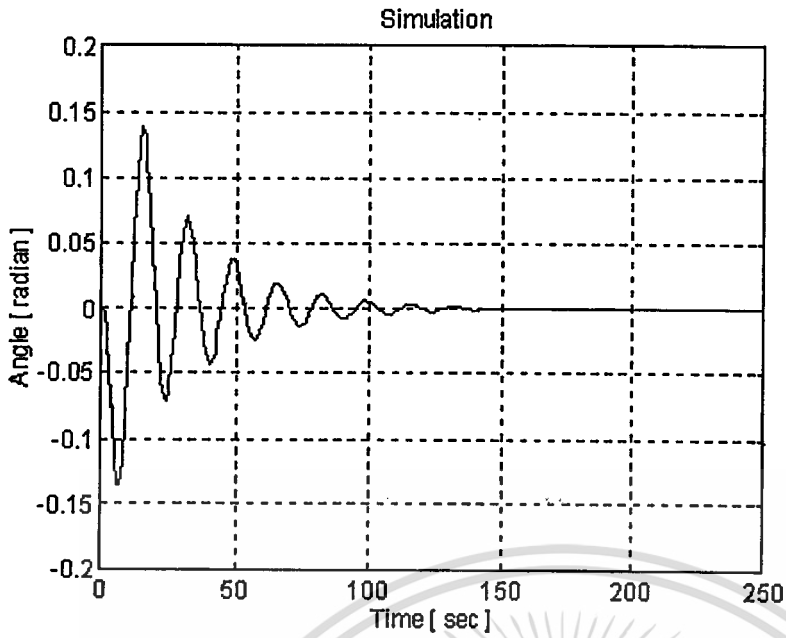
$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

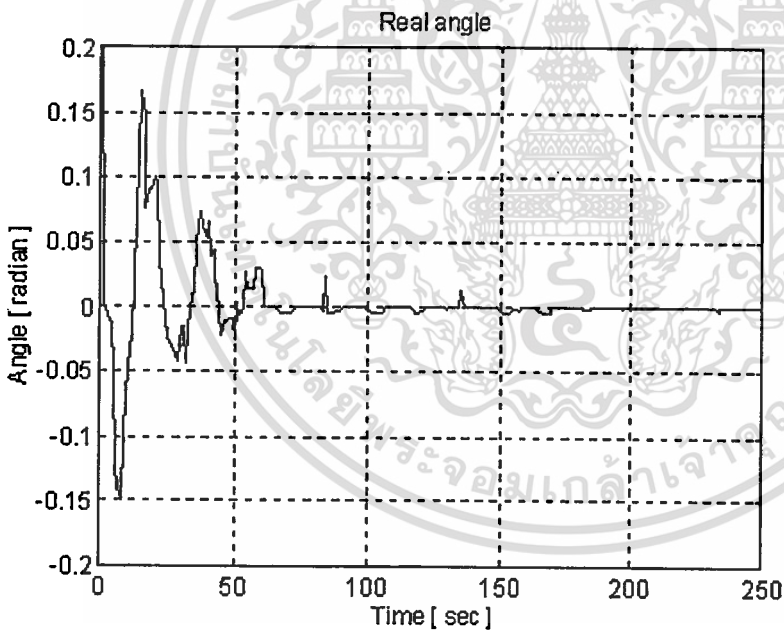
$$\text{ได้ค่า } K = [-0.2505 \quad -0.1736 \quad 0.6563 \quad 0.2321 \quad 0.0476]$$

$$E = \begin{matrix} 0.5256 \\ 0.6510 \\ 0.9305 \\ 0.8970 + 0.3558i \\ 0.8970 - 0.3558i \end{matrix}$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.21 และ 6.23 และผลจากการทดลองดังรูปที่ 6.22 และ 6.24

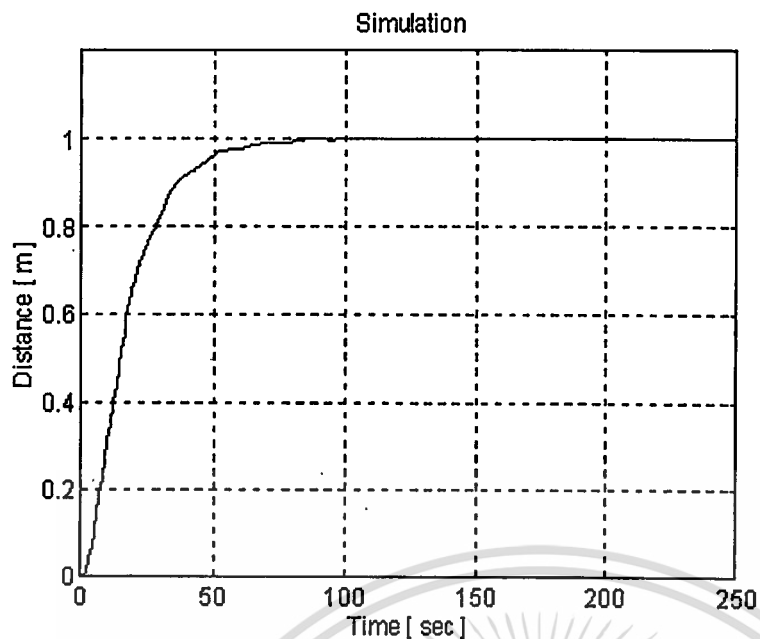


รูปที่ 6.21 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

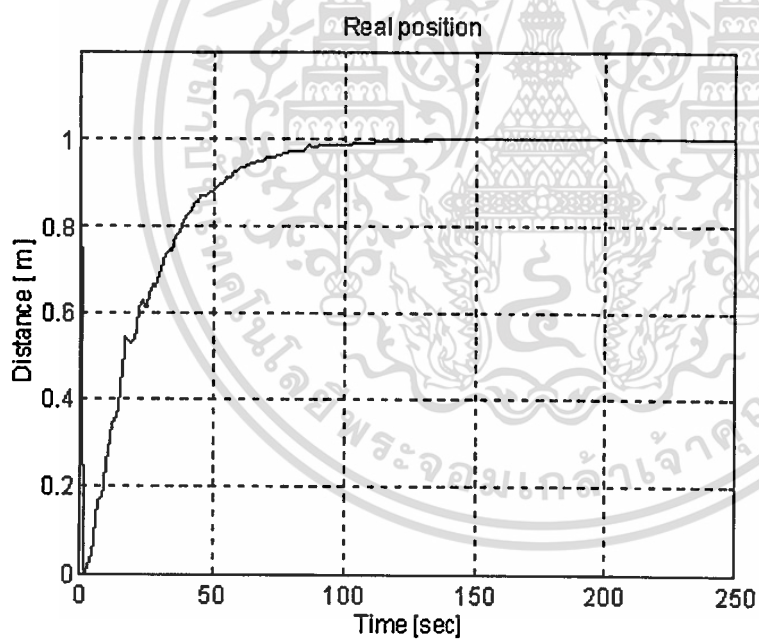


รูปที่ 6.22 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.23 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.24 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.2.2 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

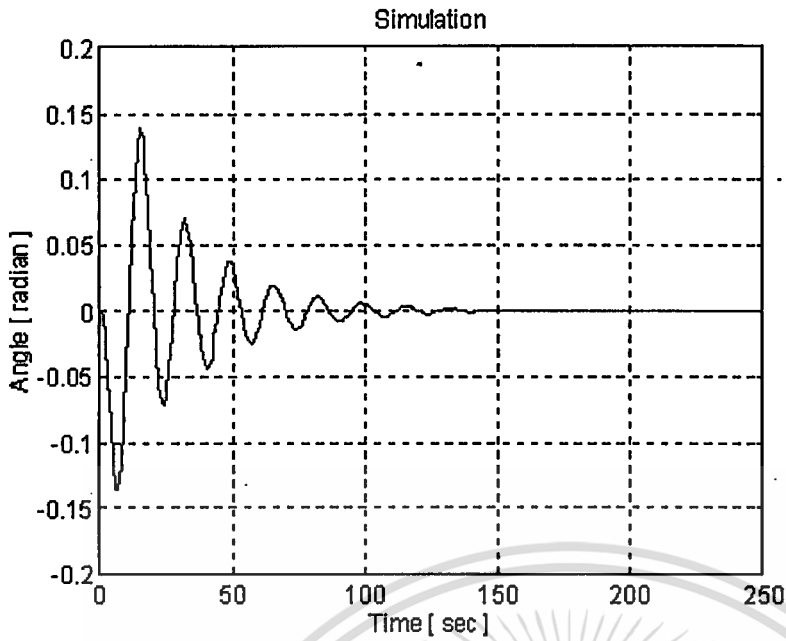
$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

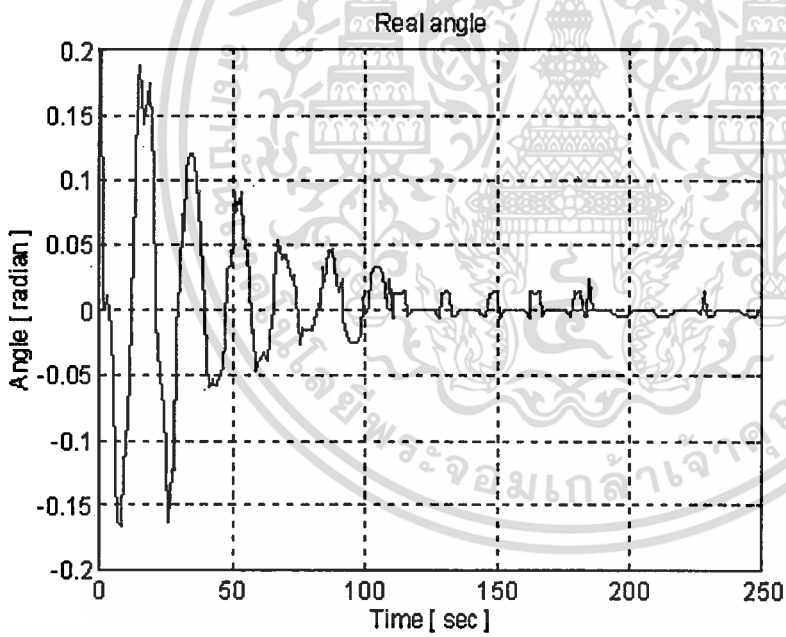
ได้ค่า $K = [-0.1046 \quad -0.0826 \quad 0.6680 \quad 0.2353 \quad 0.0361]$

$$E = \begin{matrix} 0.5275 \\ 0.6464 \\ 0.9306 \\ 0.9146 + 0.3623i \\ 0.9146 - 0.3623i \end{matrix}$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.25 และ 6.27 และผลจากการทดลองดังรูปที่ 6.26 และ 6.28

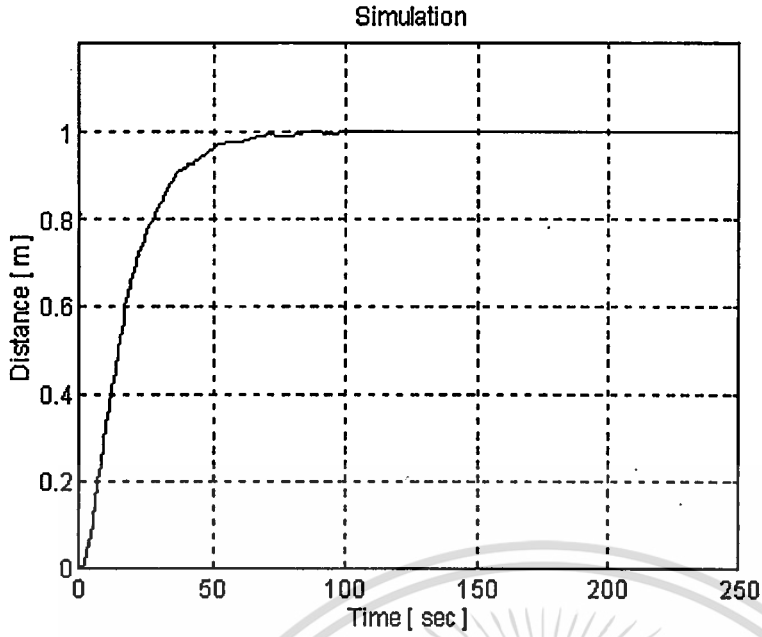


รูปที่ 6.25 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

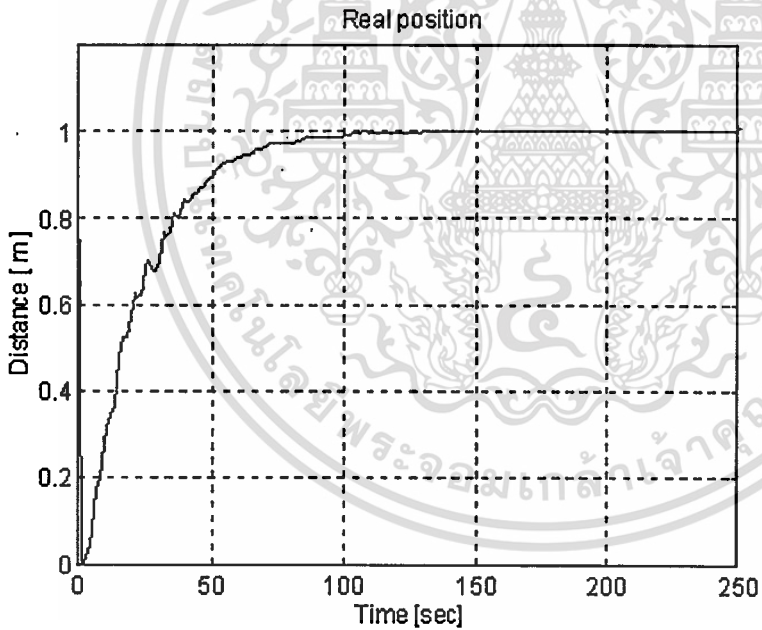


รูปที่ 6.26 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.27 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.28 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

การทดลองที่ 6.2.3 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

ได้ค่า $K = [-0.1766 \quad -0.1839 \quad 0.2120 \quad 0.0511 \quad 0.0336]$

$E = 0.5273$

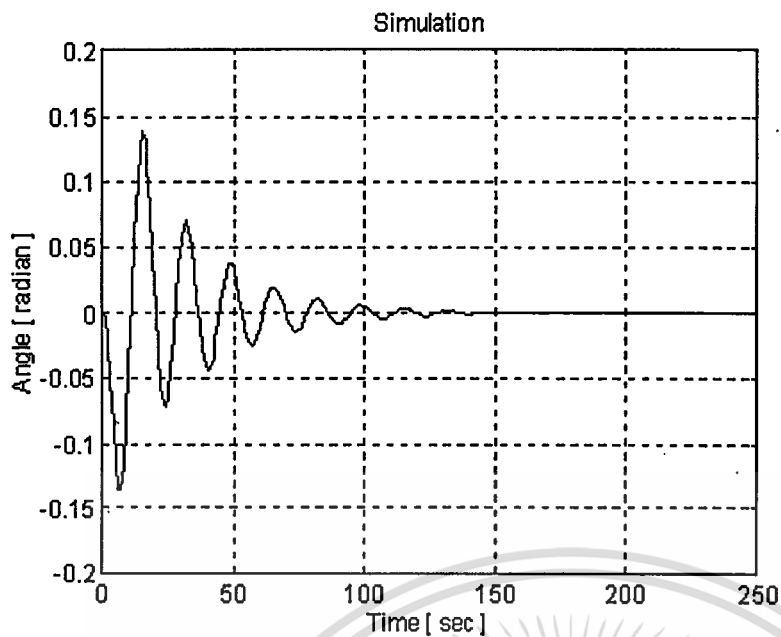
0.6444

$0.8968 + 0.3558i$

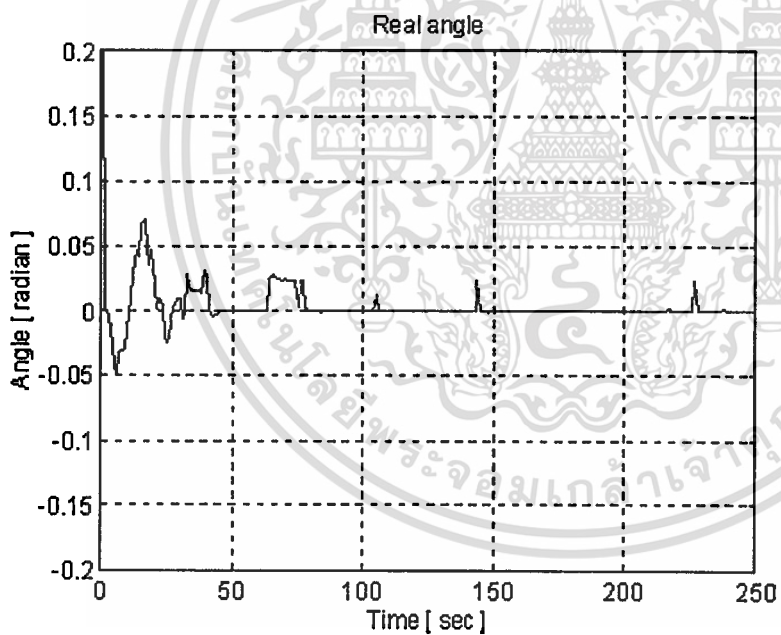
$0.8968 - 0.3558i$

0.9779

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.29 และ 6.31 และผลจากการทดลองดังรูปที่ 6.30 และ 6.32

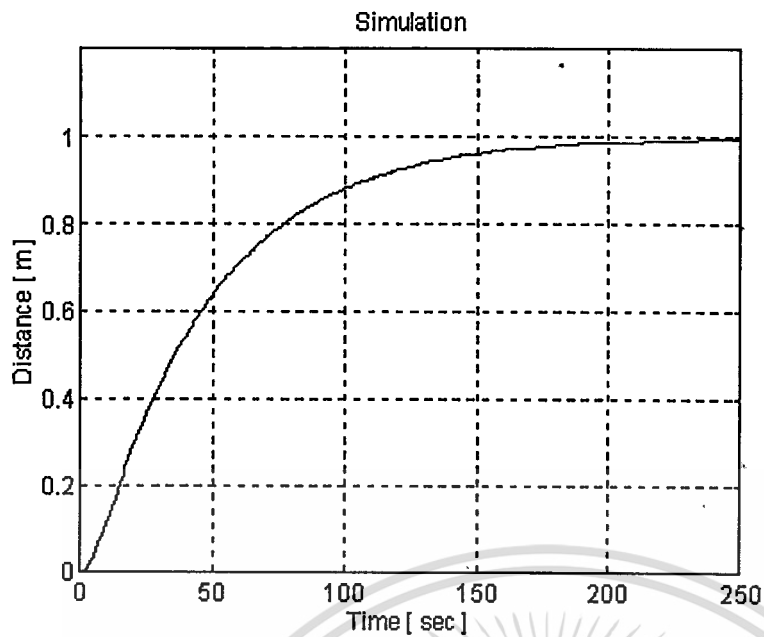


รูปที่ 6.29 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

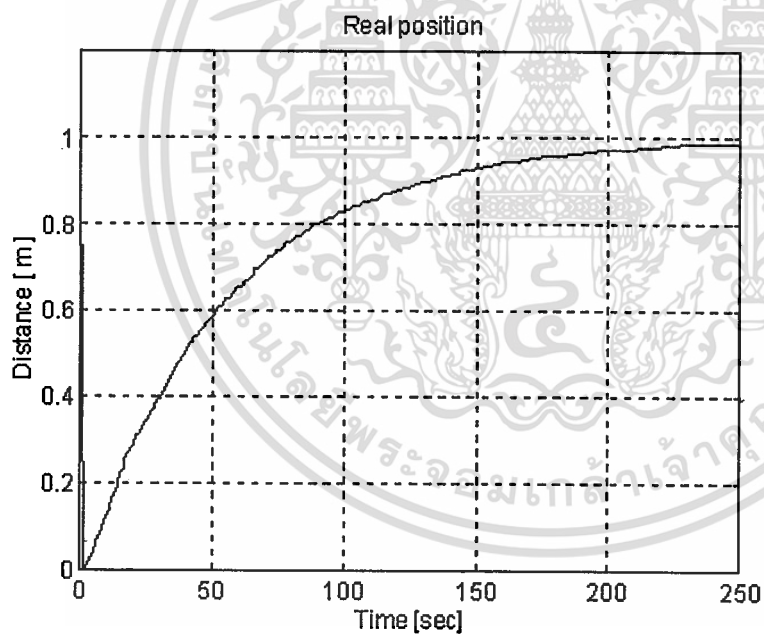


รูปที่ 6.30 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.31 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.32 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.2.4 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [50]$$

$$\text{ค่า } K = [-0.6579 \quad -0.2962 \quad 1.2137 \quad 0.4688 \quad 0.0893]$$

$$E = 0.4663$$

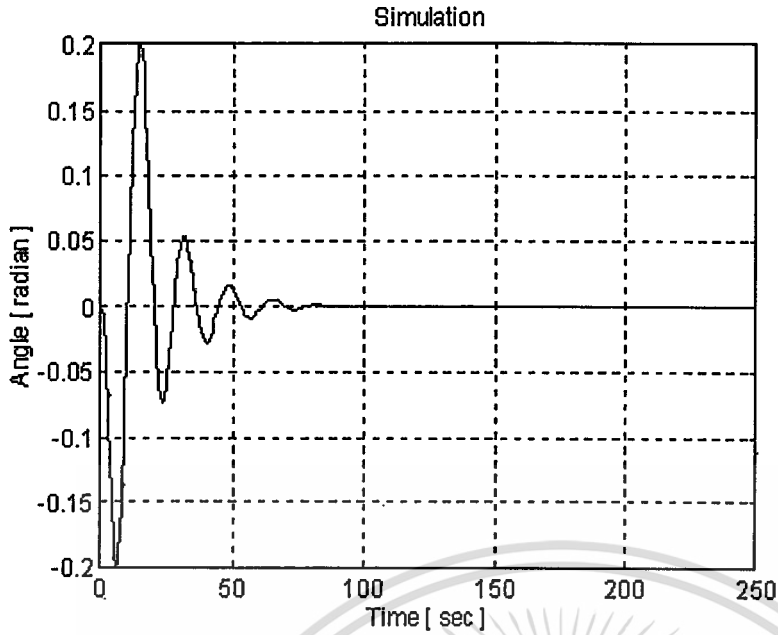
$$0.7329$$

$$0.8494$$

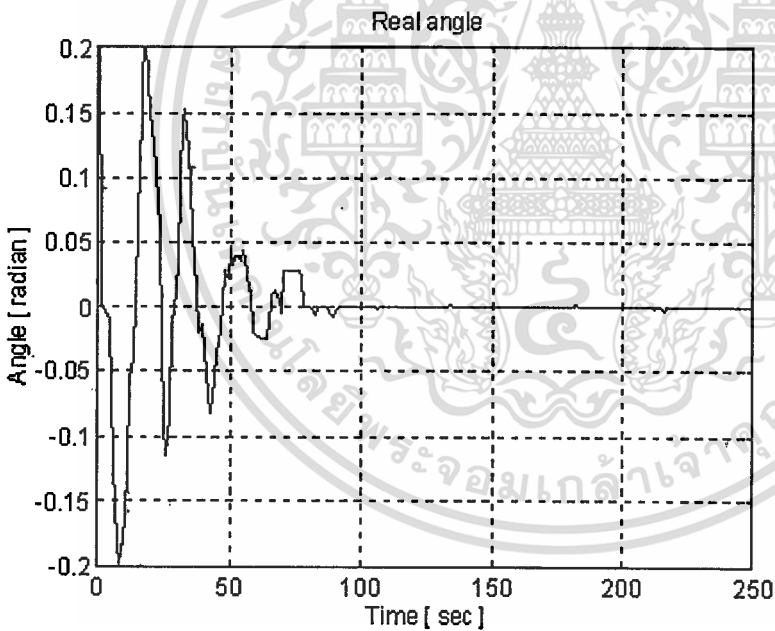
$$0.8686 + 0.3446i$$

$$0.8686 - 0.3446i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.33 และ 6.35 และผลจากการทดลองดังรูปที่ 6.34 และ 6.36

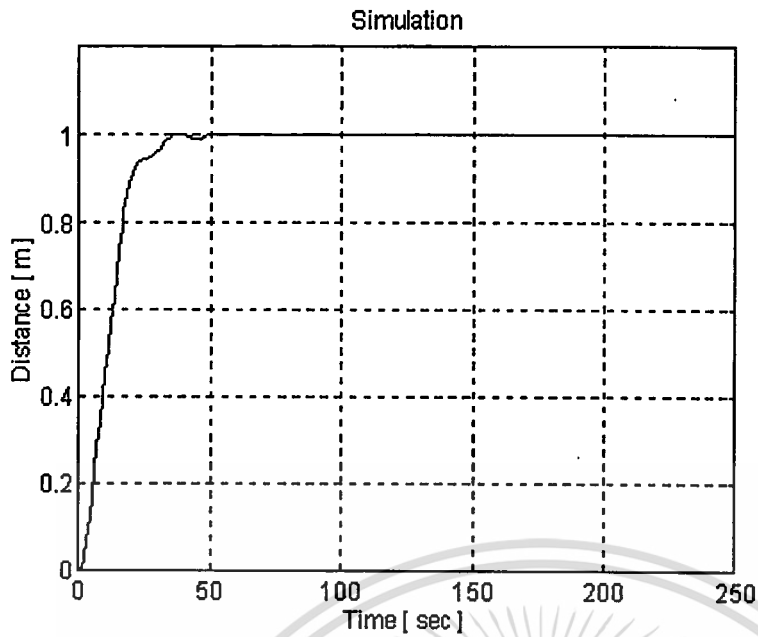


รูปที่ 6.33 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

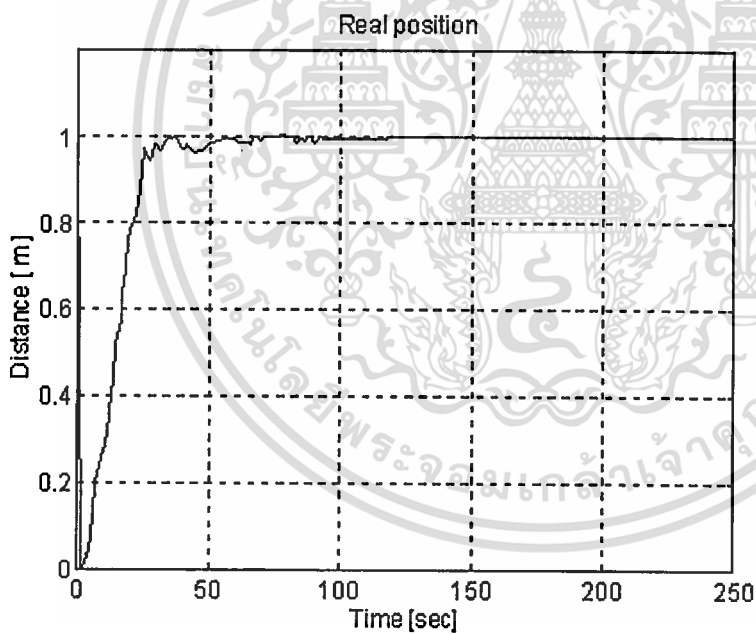


รูปที่ 6.34 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.35 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.36 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.2.5 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [20]$$

$$\text{ได้ค่า } K = [-1.1591 \quad -0.3887 \quad 1.7494 \quad 0.7282 \quad 0.1299]$$

$$E = 0.4009$$

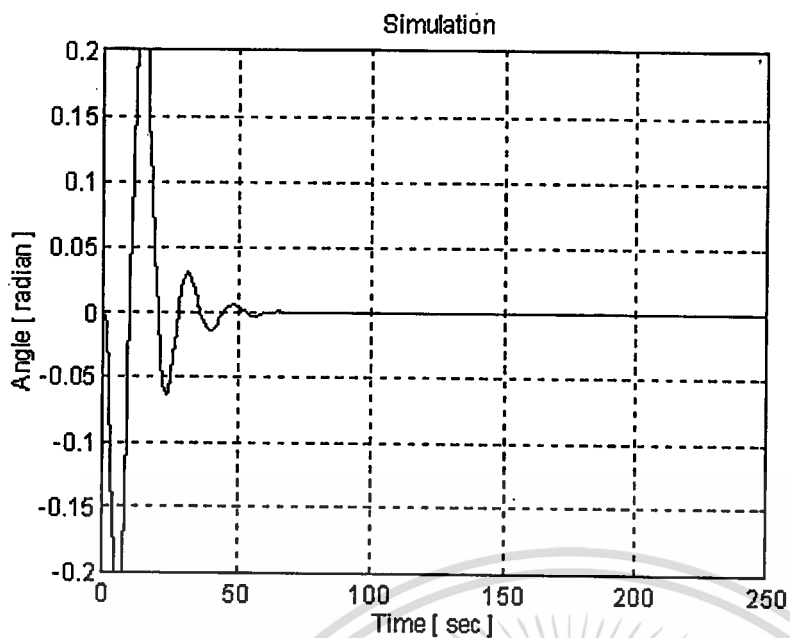
$$0.7942 + 0.0958i$$

$$0.7942 - 0.0958i$$

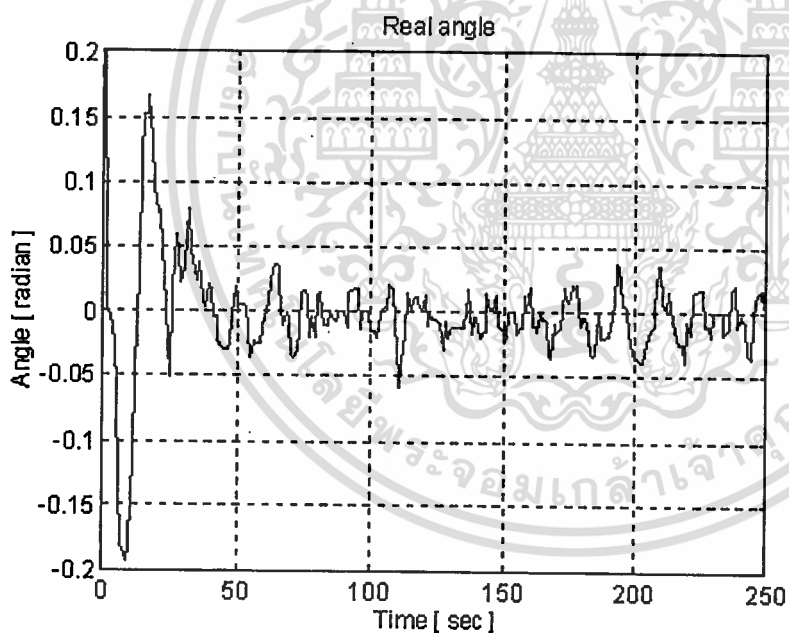
$$0.8424 + 0.3338i$$

$$0.8424 - 0.3338i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.37 และ 6.39 และผลจากการทดลองดังรูปที่ 6.38 และ 6.40

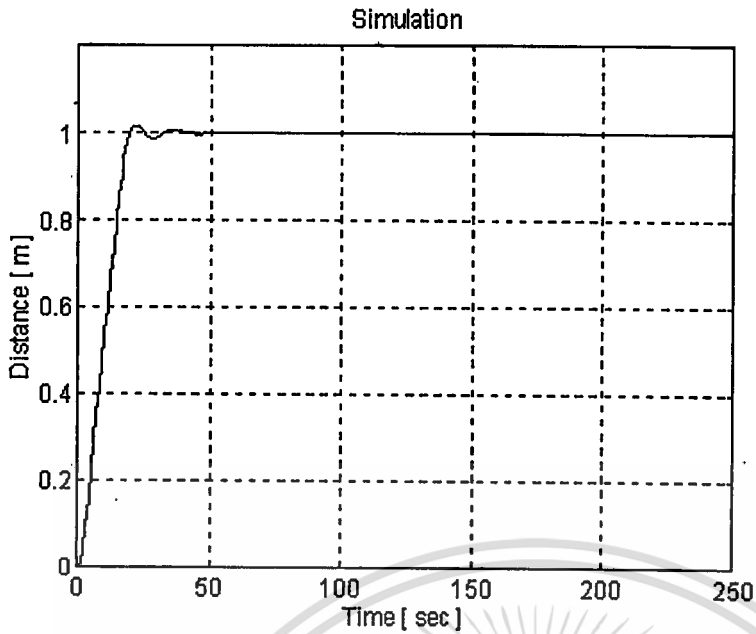


รูปที่ 6.37 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

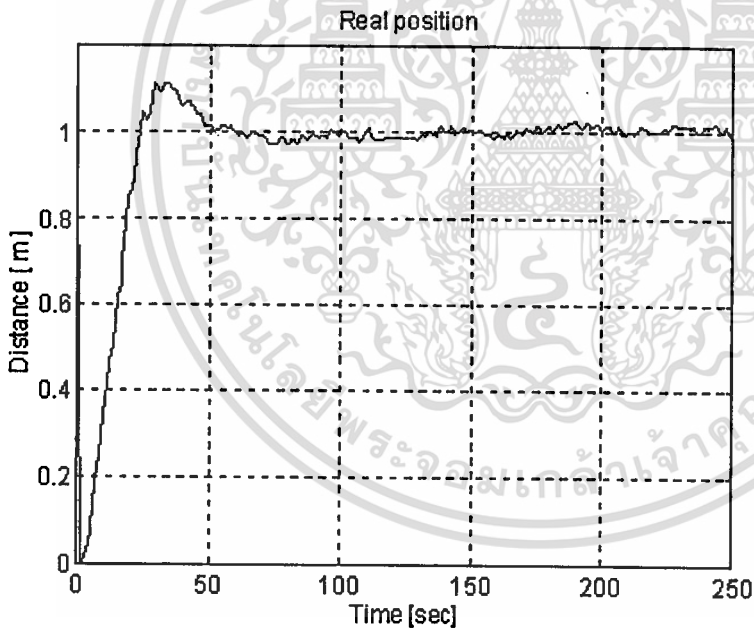


รูปที่ 6.38 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.39 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.40 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองที่ความยาวเชือก 89.75 cm

การทดลองที่ 6.3.1 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

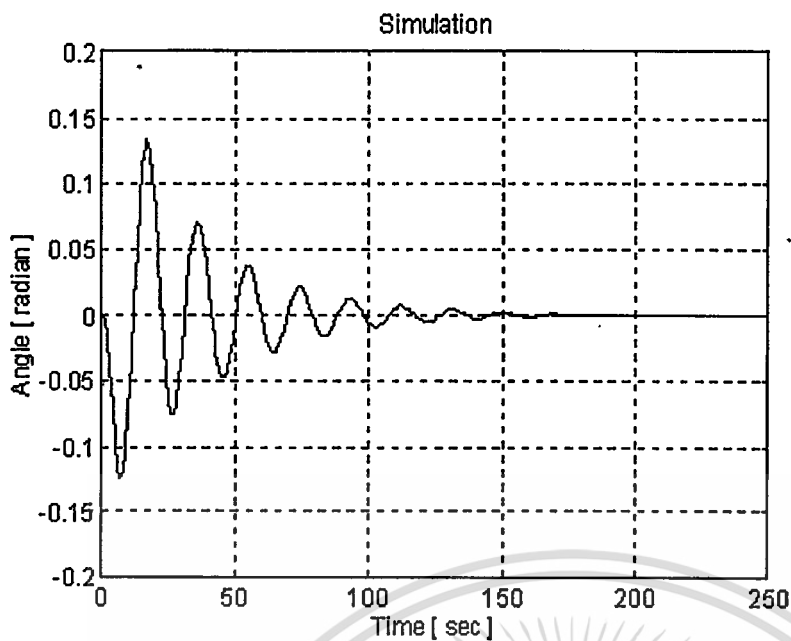
$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

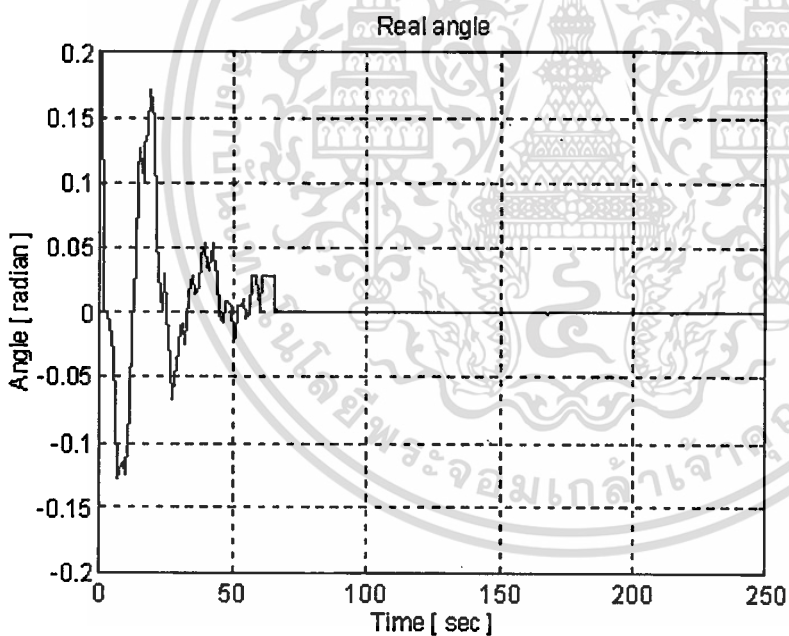
$$\text{ได้ค่า } K = [-0.3490 \quad -0.1819 \quad 0.6602 \quad 0.2413 \quad 0.0437]$$

$$E = \begin{matrix} 0.5260 \\ 0.6499 \\ 0.9305 \\ 0.9187 + 0.3152i \\ 0.9187 - 0.3152i \end{matrix}$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.41 และ 6.43 และผลจากการทดลองดังรูปที่ 6.42 และ 6.44

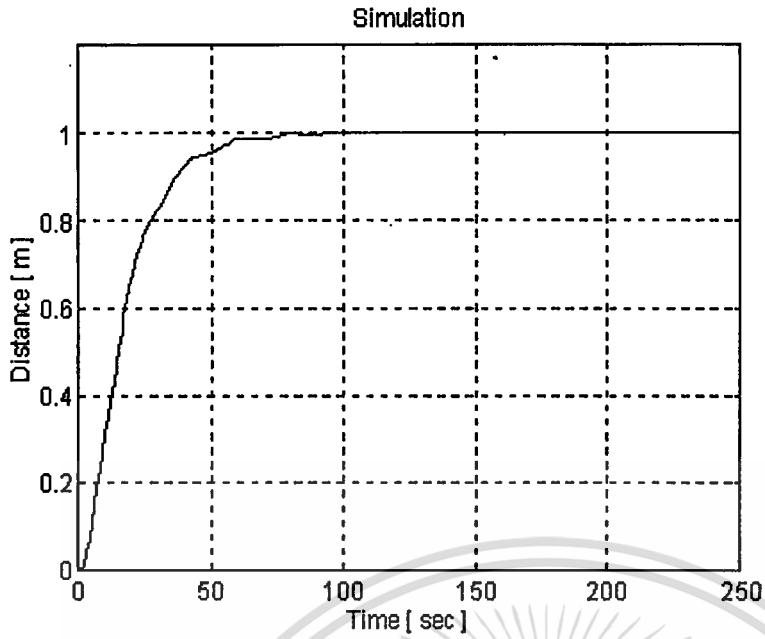


รูปที่ 6.41 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

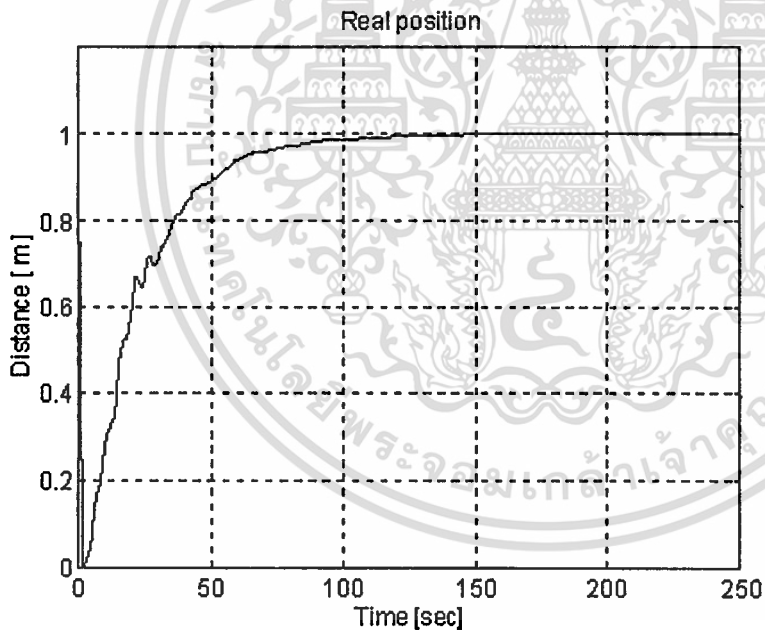


รูปที่ 6.42 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.43 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.44 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.3.2 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

$$\text{ได้ค่า } K = [-0.1449 \quad -0.0818 \quad 0.6704 \quad 0.2398 \quad 0.0337]$$

$$E = 0.5273$$

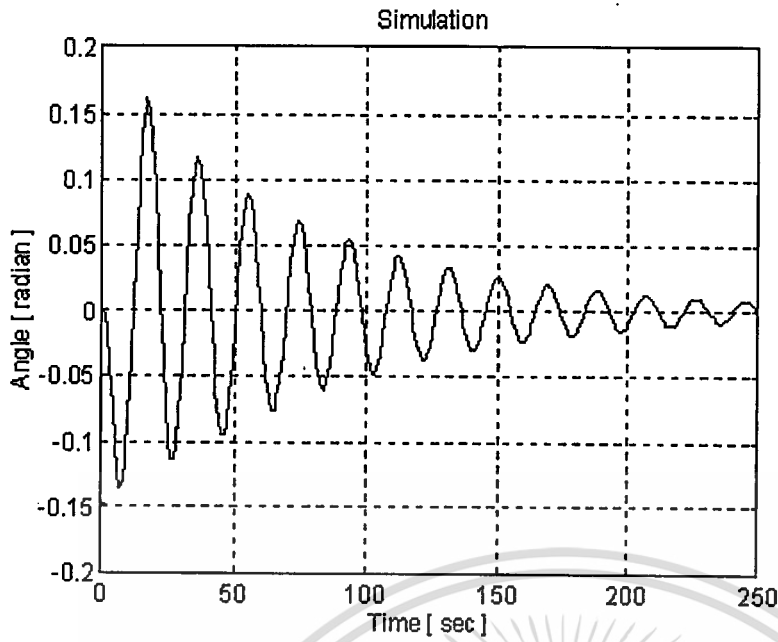
$$0.6465$$

$$0.9306$$

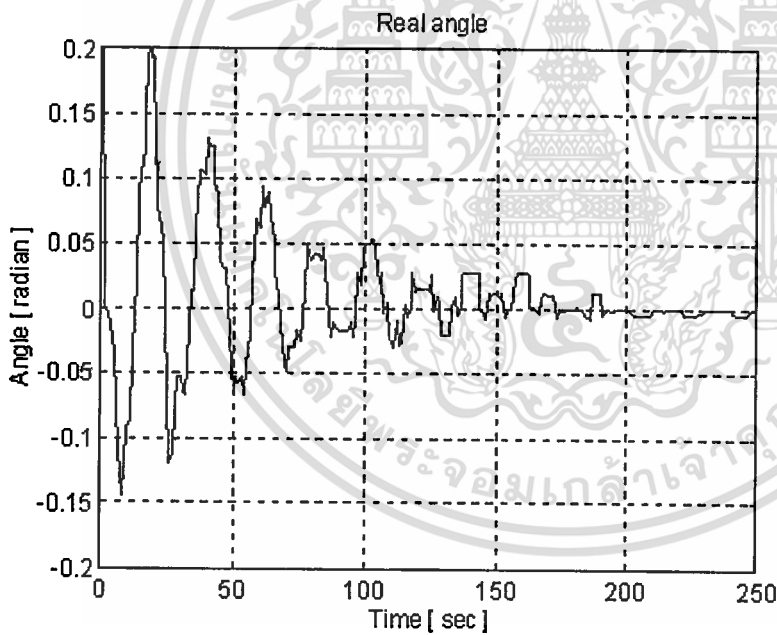
$$0.9340 + 0.3203i$$

$$0.9340 - 0.3203i$$

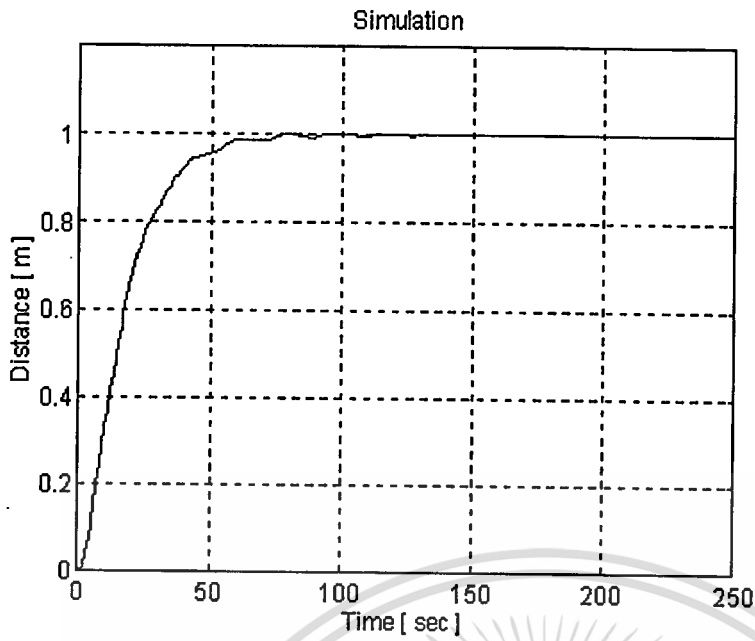
จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.45 และ 6.47 และผลจากการทดลองดังรูปที่ 6.46 และ 6.48



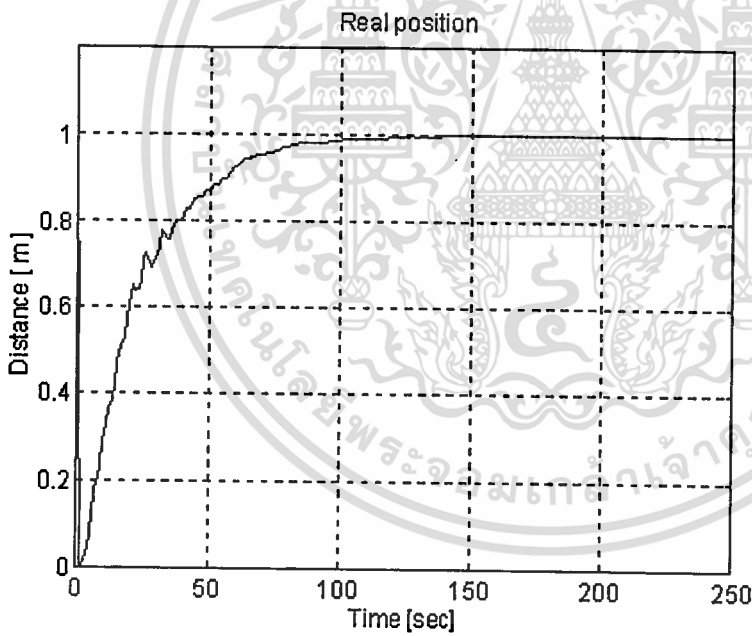
รูปที่ 6.45 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate



รูปที่ 6.46 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง



รูปที่ 6.47 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.48 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.3.3 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [200]$$

$$\text{ได้ค่า } K = [-0.2716 \quad -0.1985 \quad 0.2132 \quad 0.0580 \quad 0.0297]$$

$$E = 0.5277$$

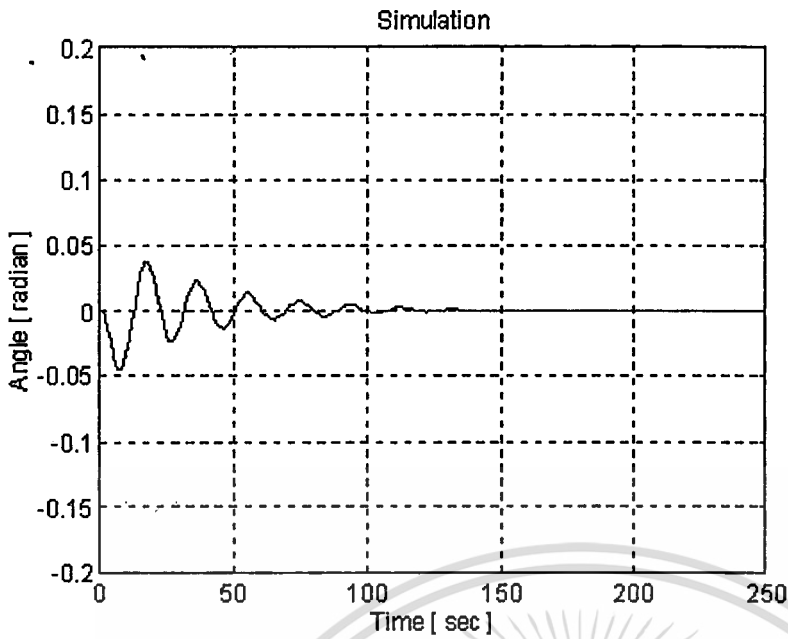
$$0.6433$$

$$0.9184 + 0.3151i$$

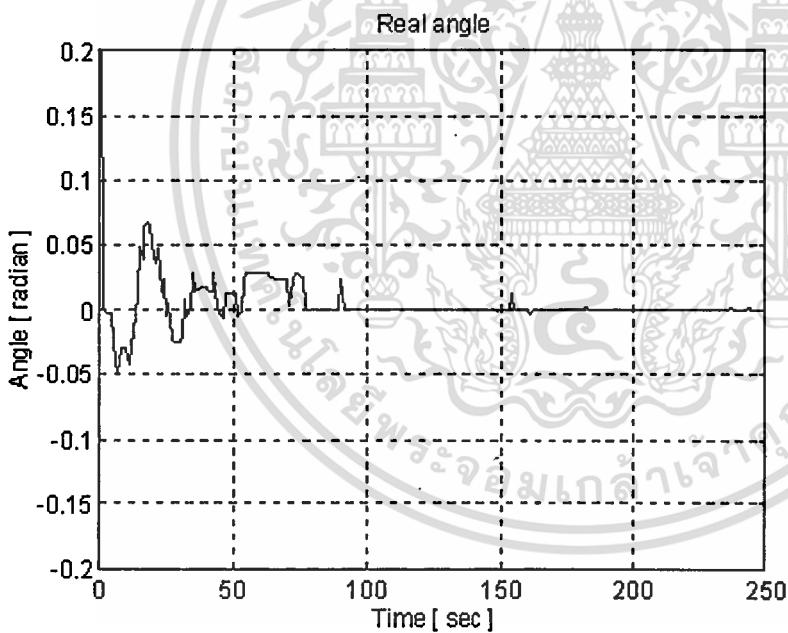
$$0.9184 - 0.3151i$$

$$0.9779$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.49 และ 6.51 และผลจากการทดลองดังรูปที่ 6.50 และ 6.52

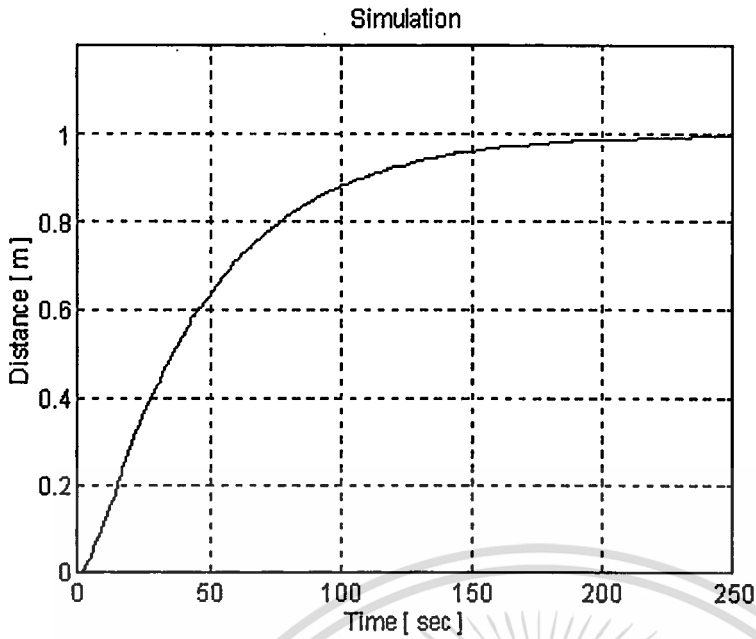


รูปที่ 6.49 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

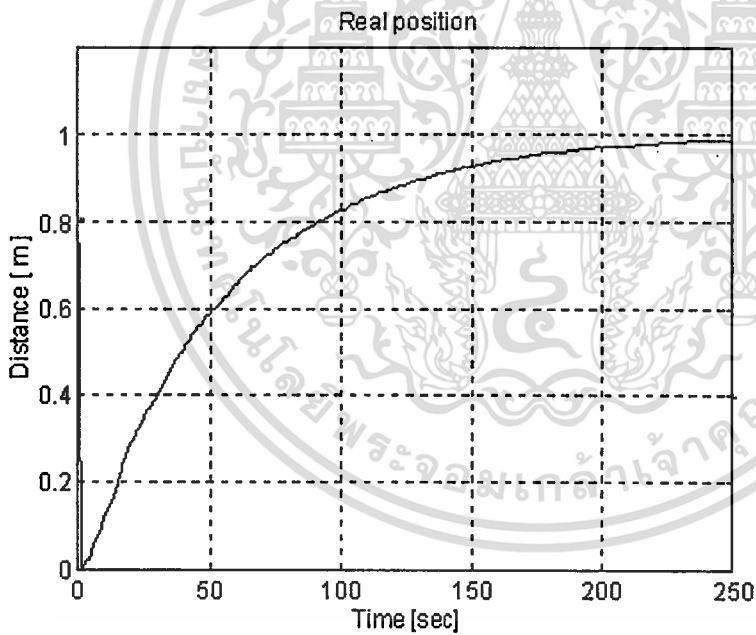


รูปที่ 6.50 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.51 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.52 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.3.4 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [50]$$

$$\text{ได้ค่า } K = [-0.8353 \quad -0.2983 \quad 1.2271 \quad 0.4893 \quad 0.0828]$$

$$E = 0.4667$$

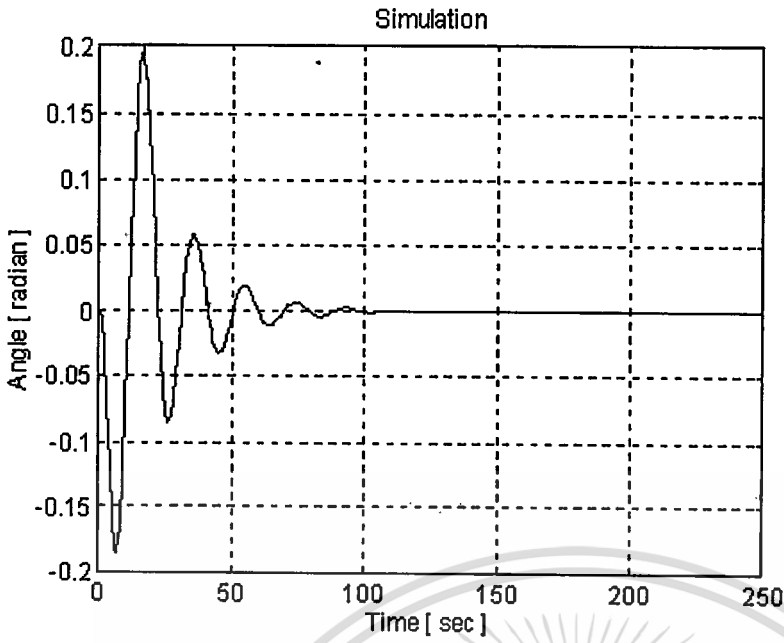
$$0.7294$$

$$0.8499$$

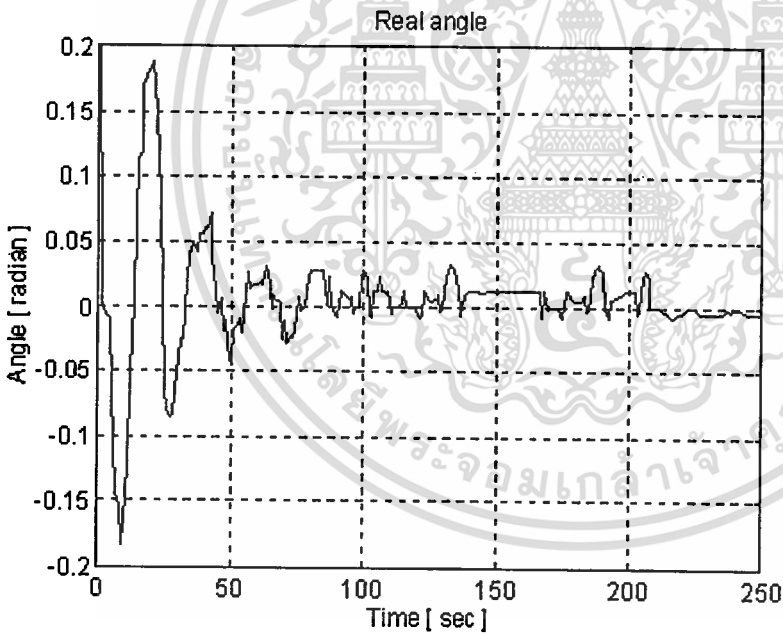
$$0.8955 + 0.3064i$$

$$0.8955 - 0.3064i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.53 และ 6.55 และผลจากการทดลองดังรูปที่ 6.54 และ 6.56

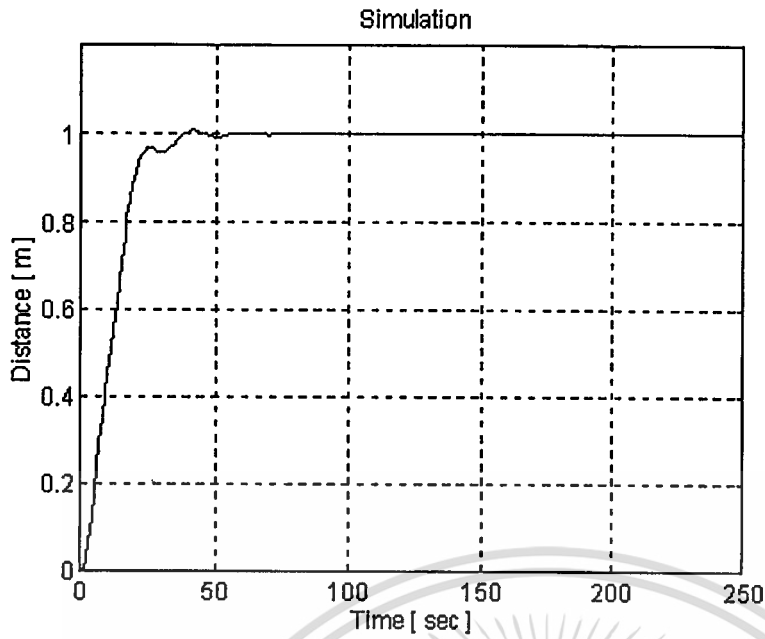


รูปที่ 6.53 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

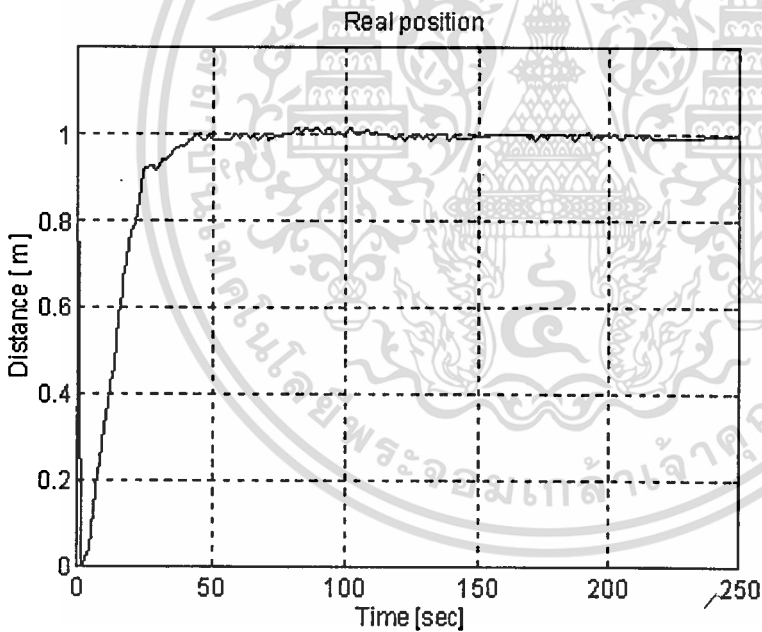


รูปที่ 6.54 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.55 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.56 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6.3.5 เมื่อเมตริกซ์ Q และเมตริกซ์ R มีค่าดังนี้

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = [20]$$

$$\text{ได้ค่า } K = [-1.3996 \quad -0.3755 \quad 1.7762 \quad 0.7608 \quad 0.1214]$$

$$E = 0.4010$$

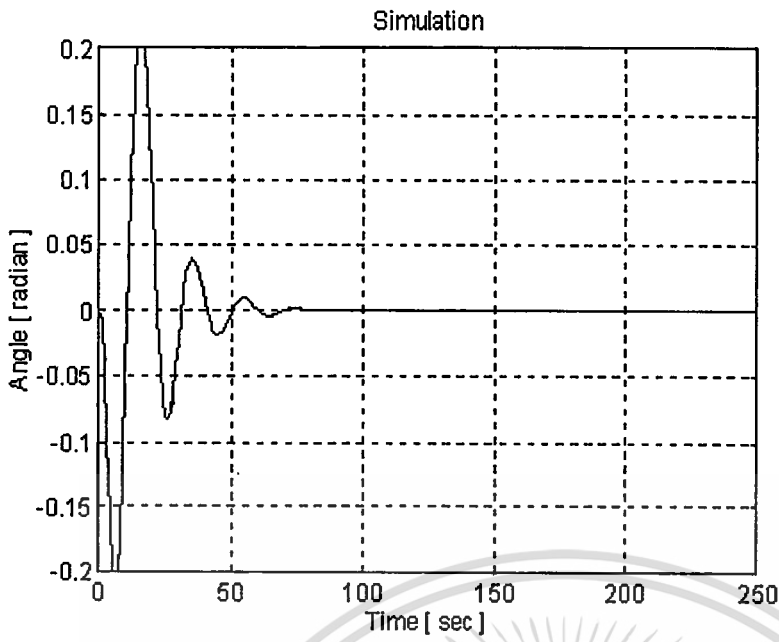
$$0.7909 + 0.0953i$$

$$0.7909 - 0.0953i$$

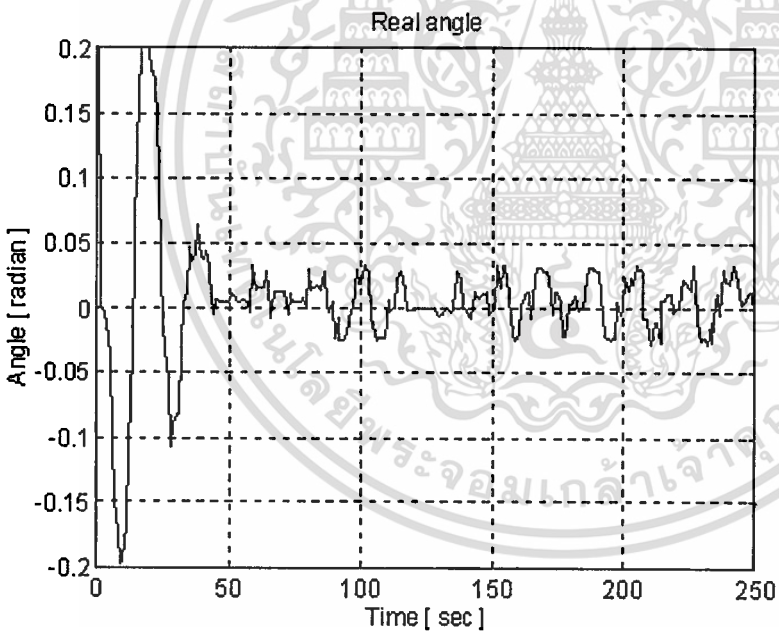
$$0.8746 + 0.2970i$$

$$0.8746 - 0.2970i$$

จะได้กราฟที่แสดงการเปรียบเทียบระหว่างผลที่ทำการ Simulate ดังรูปที่ 6.57 และ 6.59 และผลจากการทดลองดังรูปที่ 6.58 และ 6.60

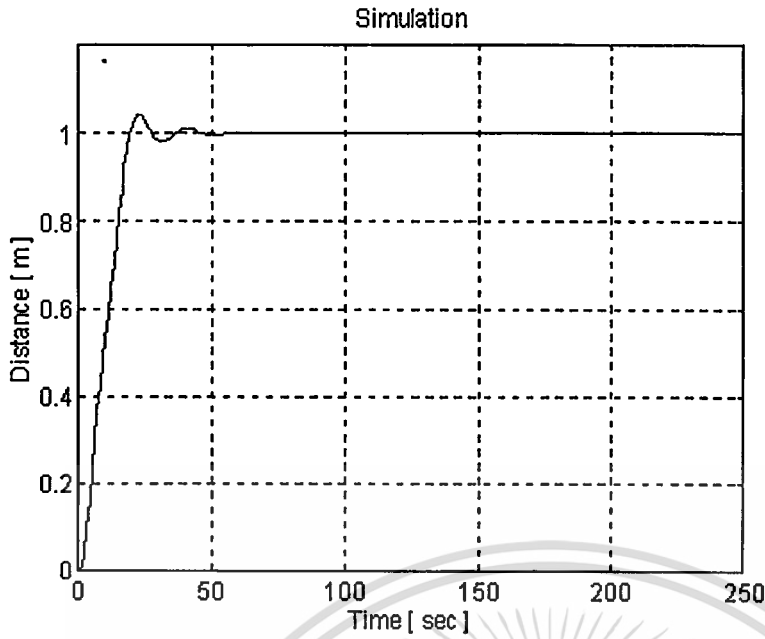


รูปที่ 6.57 แสดงผลตอบสนองของมุมที่ได้จากการ Simulate

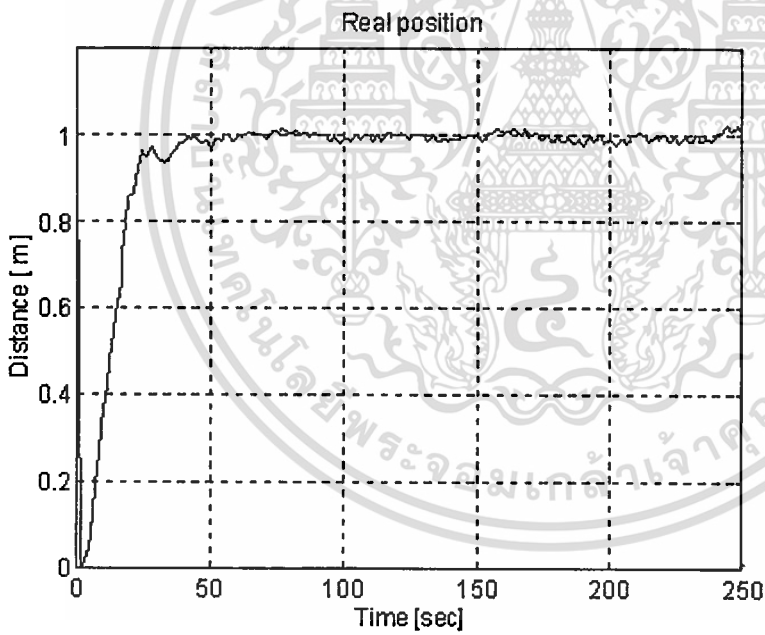


รูปที่ 6.58 แสดงผลตอบสนองของมุมที่ได้จากการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.59 แสดงผลตอบสนองของระยะทางที่ได้จากการ Simulate



รูปที่ 6.60 แสดงผลตอบสนองของระยะทางที่ได้จากการทดลอง

บทที่ 7

สรุปและวิจารณ์

จากการทดลองในส่วนของตัวควบคุมผลการทดลองที่ได้ จะมีความสัมพันธ์กับเมตริกซ์ Q และ R โดยแยกพิจารณาได้ดังนี้

การเปลี่ยนค่าเมตริกซ์ Q

หากเราให้ความสำคัญกับสเตทใดมาก ระบบจะพยายามกำจัดค่าผิดพลาดของสเตทนั้น พิจารณาการทดลองที่ 6.1.2 และ 6.1.3 การทดลองที่ 6.1.2 ได้ให้ความสำคัญกับสเตท x (ระยะทางของภาระ) มากกว่าสเตท θ (มุมของภาระ) ในขณะที่การทดลองที่ 6.1.3 ให้ความสำคัญต่อสเตท θ มากกว่าสเตท x เราสามารถเปรียบเทียบผลที่ได้ดังนี้

พิจารณาที่สเตท θ — การทดลองที่ 6.1.2 พบว่ามุมจะแกว่งมากกว่าการทดลองที่ 6.1.3 (เปรียบเทียบกราฟรูปที่ 6.6 และ 6.10)

พิจารณาที่สเตท x การทดลองที่ 6.1.2 พบว่าภาระเคลื่อนที่ได้เร็วกว่าการทดลองที่ 6.1.3 (เปรียบเทียบกราฟรูปที่ 6.8 และ 6.12)

การเปลี่ยนค่าเมตริกซ์ R

การเลือกค่าเมตริกซ์ R ส่งผลโดยตรงต่อระบบ พิจารณาสมการที่ (5.7) และ (5.8) จะพบว่า การลดค่าในเมตริกซ์ R จะทำให้ค่าเกน K เพิ่มขึ้น และทำให้ไอเกนแวลูของระบบปิดมีค่าจินตภาพเพิ่มขึ้น

พิจารณาจากการทดลองที่ 6.2.1 มีค่า R เป็น [200] ได้กราฟผลการทดลองดังรูปที่ 6.22 และ 6.24 เมื่อทำการลดค่า R เป็น [20] จะได้กราฟผลการทดลองดังรูปที่ 6.37 และรูปที่ 6.38 พบว่าการลดค่าในเมตริกซ์ R จะทำให้ระบบตอบสนองเร็วขึ้น แต่เสถียรภาพลดลง Over shoot เพิ่มขึ้น

การเปลี่ยนค่าความยาวเชือกที่แขวนภาระ

การเพิ่มความยาวเชือกที่แขวนภาระจะทำให้ระบบเข้าสู่สถานะคงตัว (Steady State) เร็วขึ้น เกิด Over shoot เพิ่มขึ้น สังเกตจากกราฟผลการทดลองที่ 6.1.5 และผลการทดลองที่ 6.3.5 เพื่อเปรียบเทียบ

ข้อเสนอแนะของระบบควบคุม

1. ควรเพิ่มส่วนควบคุมให้สามารถดิ่งภาระขึ้นลงได้
2. ควรทดลองนำตัวควบคุมแบบอื่นมาทำการควบคุม เพื่อหาตัวควบคุมที่เหมาะสมที่สุด เช่น Fuzzy Logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสนอแนะของตัวระบบจำลองเครน

1. ส่วนของเซ็นเซอร์ ที่ วัดมุม ควรเปลี่ยนจาก โพเทนทิ โอมิเตอร์ มาเป็น เอนโคเดอร์ เพื่อลดสัญญาณรบกวน และให้ความแม่นยำสูงกว่าเนื่องจากเอนโคเดอร์จะให้สัญญาณเป็นสัญญาณดิจิทัล
2. ควรหมั่นตรวจสอบแปร่งถ่านของมอเตอร์ ที่ขับเคลื่อนให้เคลื่อนที่ในแนวแกนนอน เพราะมอเตอร์ที่ใช้มีอายุการใช้งานมากแล้ว



หนังสืออ้างอิง

- (1) นิรุศด์ นาคสุข และ สุนิบุญ ตั้งวิญญู, ปรินญาณินพนธ์เรื่อง “ การควบคุมการเคลื่อนที่ของเครน ” ปีการศึกษา 2538 : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- (2) นรพล ตั้งตรงจิตต์ , ปัญจรัภย์ นิงสุนกิจ และ ภาคภูมิ สุกแสนถาวร , ปรินญาณินพนธ์เรื่อง “ การควบคุมการเคลื่อนที่ของเครน ” ปีการศึกษา 2539 : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- ชานินทร์ ถาวรศาสนวงศ์ และ ทินกร คู้ก , “ การอินเตอร์เฟส IBM/PC ” : หก.พีลิสต์เซ็นเตอร์, 2536
- ธันวา ศรีประโงม , “ การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม ” , พิมพ์ครั้งที่ 4 : โครงการนตำราวิชาการ มหาวิทยาลัยเทคโนโลยีมหานคร , 2538
- รศ. วิพันธ์ ปรึษาพานิช , “ การวิเคราะห์ระบบควบคุมเวลาติสครีต ” : , 2537
- Katsuhiko Ogata , “ Modern Control Engineering ” , 2nd ed. : Prentice Hall , Inc.
- Katsuhiko Ogata , “ Discrete Time Control Systems ” , 2nd ed. : Prentice Hall , Inc.
- Charlse L.Phillips , and H.Troy Nagle , “ Digital Control System Analysis and Design ” , 3rd ed : Prentice Hall , Inc.
- BahRam Shahian , and Micheal Hassul , “ Control System Design Using Matlab ” : Prentice Hall , Inc.
- “ Motorola Bipolar Power Transistor Data ” : National Semiconductor
- “ National Application Specific Analog Products Databooks ” : Motorola Inc.



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

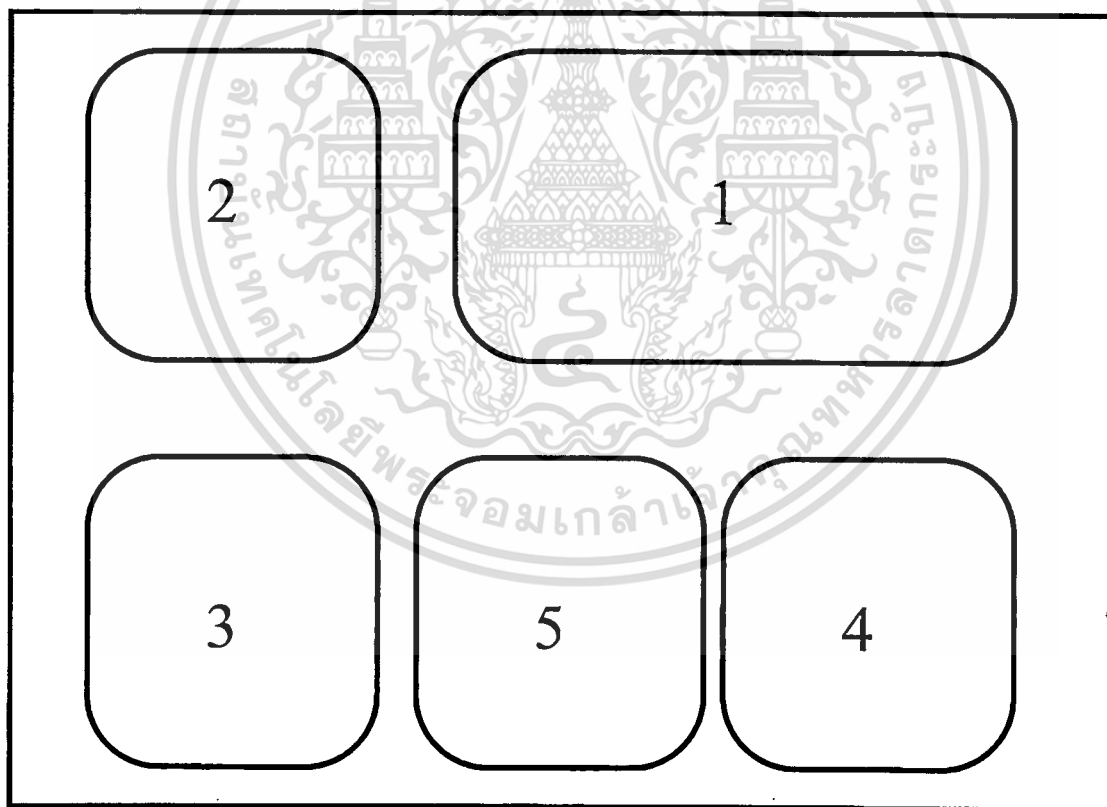


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
กล่องชุดควบคุมเกรน

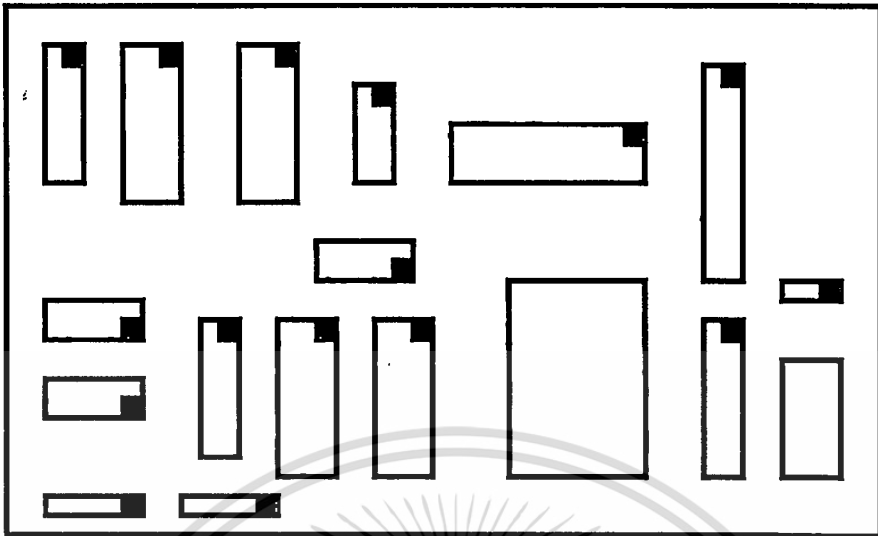
กล่องควบคุมเกรนประกอบไปด้วย

1. วงจรหลัก
2. วงจรตรวจวัดค่ามุม
3. วงจรป้องกันไฟเลี้ยง
4. วงจรขับเคลื่อนมอเตอร์ในแนวแกนนอน
5. วงจรขับเคลื่อนมอเตอร์ในแนวแกนตั้ง
6. วงจรแสดงผลด้วย LED (ติดอยู่กับฝากล่อง)



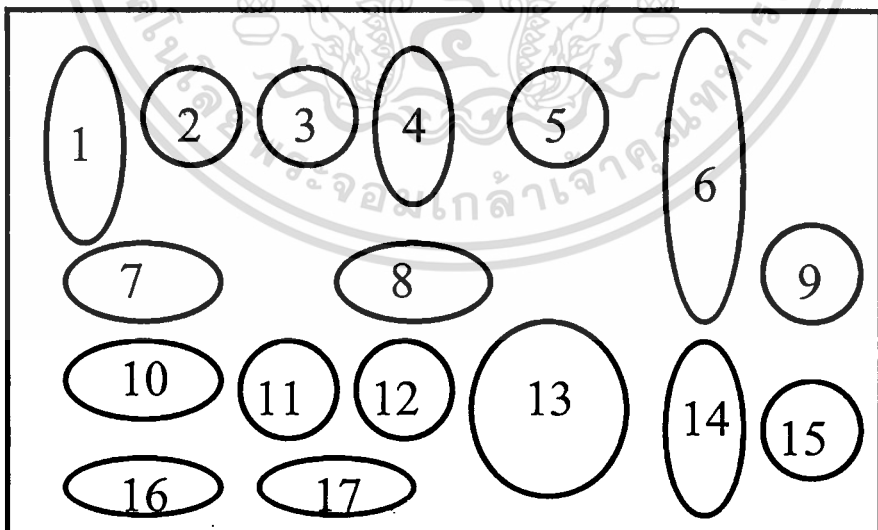
รูปที่ ก.1 แสดงตำแหน่งของวงจรภายในกล่องควบคุม

ก.1 วงจรหลัก



รูปที่ ก.2 แสดงวงจรหลักภายในกล่องควบคุม

โดยการต่อวงจรต่างๆให้สายที่เป็นเส้นสีแดง อยู่ทางด้านขวามือหรือทางด้านบนเสมอ (พิจารณาตำแหน่งที่ทำเครื่องหมายไว้) และการใส่ไอซีให้วางตำแหน่งขาที่หนึ่งของไอซีตรงกับที่ทำเครื่องหมายไว้



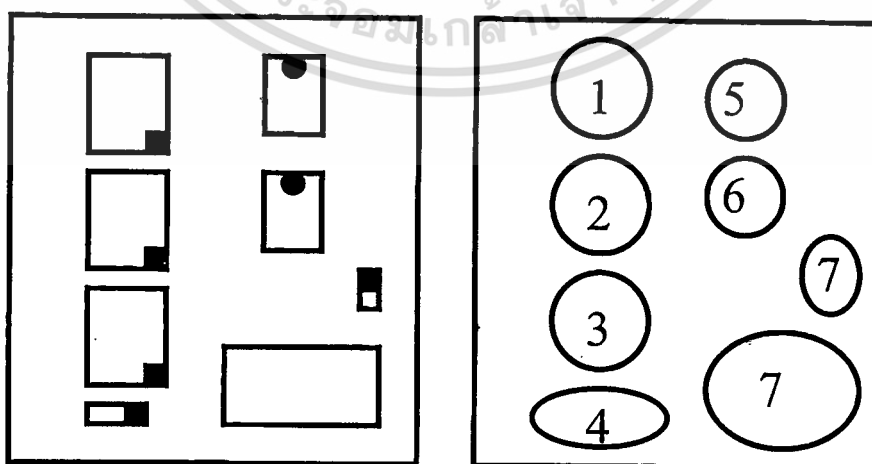
รูปที่ ก.3 แสดงตำแหน่งของอุปกรณ์บนวงจรหลัก

อุปกรณ์ต่างๆที่ต่ออยู่บนวงจรหลักให้สังเกตจากตำแหน่งของไฟเลี้ยงวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อห้ามการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
1. สวิตซ์เลือกโหมดของมอเตอร์ในแนวแกนนอน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. 74HC08
3. 74HC04
4. LED 1 -LED4 ต่ออยู่บนวงจรแสดงผลด้วย LED
5. 26LS32
6. DB 25
7. Limit Switch
8. ปุ่มบังคับทิศทาง
9. วงจรตรวจวัดค่านุ่ม
10. ไป LED5 - LED8 ต่ออยู่บนวงจรแสดงผลด้วย LED
11. 74LS08
12. 74LS86
13. วงจรกำเนิดสัญญาณขั้วมอเตอร์ด้วย LM555
14. ไป DB15
15. ไฟเลี้ยงวงจร
16. สัญญาณขั้วมอเตอร์ในแนวแกนตั้ง
17. สัญญาณขั้วมอเตอร์ในแนวแกนนอน

ก.2 วงจรตรวจวัดค่านุ่ม



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ ก.4 แสดงตำแหน่งของวงจรตรวจวัดค่านุ่มให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

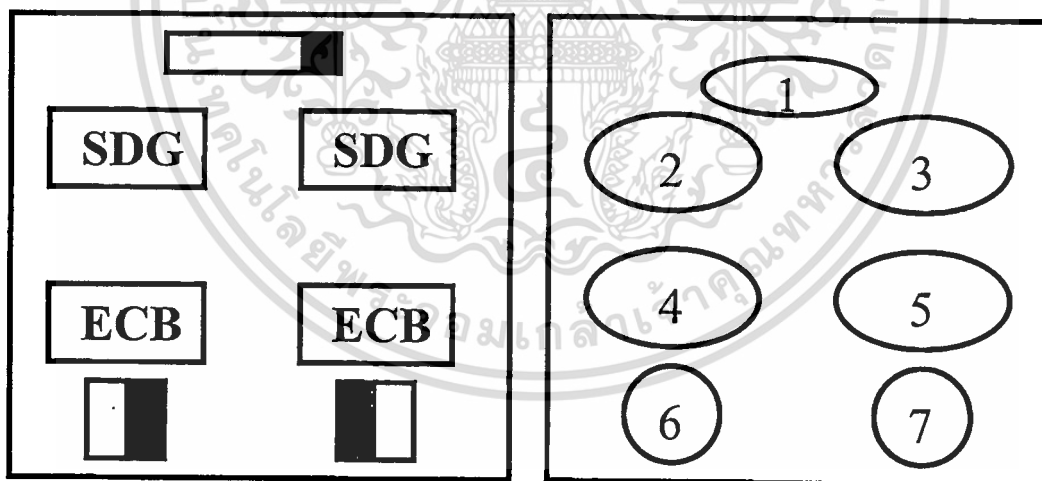
- 1.- 3. Op-Amp 741
4. สัญญาณจากวงจรหลัก
5. R1
6. R2
7. สัญญาณส่งไปยัง PCL714
8. ไฟเลี้ยง

วิธีตั้งค่ามุมเริ่มต้น

1. ปรับให้แนวลูกดิ่งตัวฉากกับพื้น โลก แล้วปรับ R2 จนกว่าค่าแรงดันที่จะจ่ายไปยัง การ์ด PCL741 มีค่าเท่ากับ 0 โวลต์
2. ตั้งค่ามุมของลูกดิ่งให้ได้ 45 องศา แล้วปรับ R1จนกว่าจะได้ค่าแรงดันเท่ากับ 5 โวลต์

ก.3 วงจรป้องกันไฟเลี้ยง อาศัยหลักการของไดโอดป้องกันการต่อไฟผิดทาง

ก.4 วงจรขับมอเตอร์



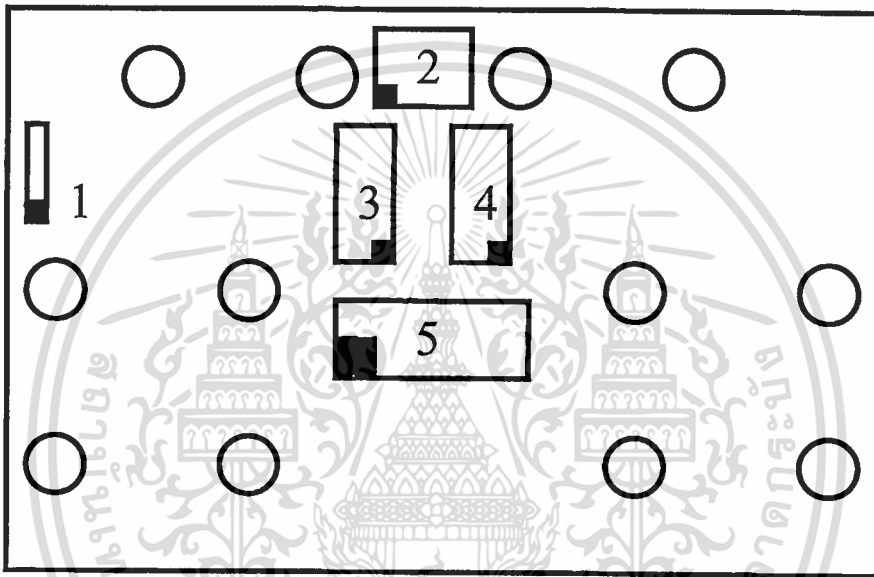
รูปที่ ก.5 แสดงตำแหน่งของอุปกรณ์บนวงจรขับมอเตอร์

การต่อวงจรขับมอเตอร์ใช้ตำแหน่งของตัวทรานซิสเตอร์ ไฟเลี้ยงและสัญญาณในตำแหน่งเดียวกัน จึงขอกว่าโดยรวมทั้งในแนวแกนนอนและแนวแกนตั้ง โดยยึดขาของทรานซิสเตอร์และมอเตอร์ตามรูปเป็นหลัก และให้สายไฟทุกเส้นที่เป็นสีแดงต่อเข้ากับขาที่ทำเครื่องหมายไว้ โดยต่ออุปกรณ์ตามรูปดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สัญญาณจากวงจรหลัก
- 2.,3. มอสเฟต
- 4.,5. ทรานซิสเตอร์
6. สัญญาณส่งไปขับมอเตอร์
7. ไฟแรงดันสูง

ก. 5 วงจรแสดงผลด้วย LED



รูปที่ ก.6 วงจรแสดงผลด้วย LED

จากรูปที่ ก.6 ประกอบไปด้วย

- LED 12 ดวง
- 1. ไฟเลี้ยงวงจร
- 2. DB-15 (ไปลิ้มิตสวิตช์)
- 3. 74HC04
- 4. 74HC08
- 5. 74LS245



ภาคผนวก ข
โปรแกรม MATLAB
ที่ใช้ในการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรม MATLAB ที่ใช้ในการคำนวณ

ข.1 โปรแกรมทดสอบการประมาณค่า ζ

นำรูปภาพ Step response ระบบจริงที่ได้มาหาค่าเวลา Settling time (T_s)

ทำการทดลองประมาณค่าที่เหมาะสม จากนั้นนำค่า ζ และ T_s แทนในโปรแกรมนี้ ให้พิจารณาว่ากราฟที่ได้จากโปรแกรมนี้ใกล้เคียงกับกราฟผลตอบสนอง จริงของ ระบบหรือไม่ ถ้าใกล้เคียงก็ถือว่าค่า ζ ที่กำหนดนั้นใช้ได้ แต่ถ้าไม่ใกล้เคียงก็ให้ทดลองประมาณค่า ζ ใหม่ อีกครั้ง

```

>>Ts      = 0.75
>>damp    = 1.0
>>Wn      = 4/(damp*Ts)
>>num     = Wn^2;
>>den     = [1      2*Wn*damp      Wn^2];
>>tep(num,den)
>>title('Step response')
>>ylabel('Velocity [m/sec]');
>>axis([0.0,      2.0,      0.0,      1.2      ]);
  
```

ข.2 โปรแกรม MATLAB ที่ใช้ในการ แปลงสมการสเตตต่อเนื่องเป็นสมการสเตตคิสคริต โดยที่สมการสเตตต่อเนื่องมีรูปแบบเป็น

$$\dot{x}(t) = Ax(t) + Bu(t)$$

และ สมการสเตตคิสคริตมีรูปแบบเป็น

$$x(k+1) = Gx(k) + Hu(k)$$

โดยต้องมีการกำหนดค่าคาบเวลาชักตัวอย่างของระบบ (T) ด้วย

```

>> T      = 0.1      %sec
>>[G,H] = c2d(A,B,T);
  
```

ข.5 โปรแกรม MATLAB ที่ใช้ในการSimulateเพื่อหา ผลตอบตำแหน่ง(x)ของภาระ

```

>>X = [0 0 1 0 0];
>>D = [0];
>>GG = G - H*K;
>>HH = H*K(3);
>>CC = C;
>>DD = D;
>>[num,den] = ss2tf(GG,HH,T,DD);
>>r = ones(1,601);
>>k = 0:600;
>>y = filter(num,den,r);
>>plot(k,y,'-',k,y,'-');
>>XMIN = 0;
>>XMAX = 250;
>>YMIN = 0.0;
>>YMAX = 1.2;
>>title('Simulation');
>>xlabel('Time [ sec ] ');
>>ylabel('Position [ m ]');
>>AXIS([XMIN XMAX YMIN YMAX])
>>AXIS(AXIS);
>>grid

```

ข.6 โปรแกรม MATLAB ที่ใช้ในการคำนวณค่าเกน K ของระบบคอน โดยโปรแกรมนี้จะนำพารามิเตอร์ต่างๆของระบบ มาคำนวณหาสมการสเตตต่อเนื่อง จากนั้นทำการแปลงเป็นสมการสเตตดิสครีต และทำการหาเกน K โดยผู้ใช้ต้องกำหนด ค่า Q และ R ลงไปด้วย โปรแกรมนี้จะนำเกน K ที่คำนวณได้ไปเก็บในไฟล์ ชื่อ “ C:DATAKF.DAT ” ให้โปรแกรม CMC3.EXE ค่า K ไปใช้ในการควบคุมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.3 โปรแกรม MATLAB ที่ใช้ในการหาค่าเมตริกซ์เกน K ด้วยวิธี LQR จาก ค่าของ G และ H ของระบบ ให้ทำการกำหนด เมตริกซ์ Q และ R ที่เหมาะสมแล้วแทนใน โปรแกรม จะได้ผลลัพธ์เป็นค่า ไอเก้น และ ค่าเมตริกซ์เกน K

```
>>Q = diag( [ 100 1 100 1 1 ] );
>>R = [200];
>>[K,S,E] = dlqr(G,H,Q,R);
```

ข.4 โปรแกรม MATLAB ที่ใช้ในการ Simulate เพื่อหา ผลตอบสนองมุมการแกว่ง(θ) ของ ภาระ

```
>>T = [1 0 0 0 0];
>>D = [0];
>>GG = G - H*K;
>>HH = H*K(3);
>>CC = C;
>>DD = D;
>>[num,den] = ss2tf(GG,HH,T,DD);
>>r = ones(1,601);
>>k = 0:600;
>>y = filter(num,den,r);
>>plot(k,y,'-',k,y,'-');
>>XMIN=0;
>>XMAX=250;
>>YMIN=-0.2;
>>YMAX=0.2;
>>title('Simulation');
>>xlabel('Time [ sec ] ');
>>ylabel('Angle [ radian ]');
>>AXIS([XMIN XMAX YMIN YMAX])
>>AXIS(AXIS);
>>grid
```

```

>>Ts = 0.75;
>>damp = 1.0;
>>Wn = 4/(damp*Ts);
>>num = Wn^2;
>>den = [1 2*Wn*damp Wn^2];
>>L = 0.4125;
>>g = 9.81;
>>A = [ 0 1 0 0 0;
      -g/L 0 0 0 -1/L;
      0 0 0 1 0;
      0 0 0 0 1;
      0 0 0 -Wn^2 -2*damp*Wn;];
>>B = [ 0;
      0;
      0;
      0;
      Wn^2;];
>>C = [ 0 0 1 0 0];
>>[G,H] = c2d(A,B,0.1);
>>Q = diag( [ 100 1 100 1 1 ] );
>>R = [200];
>>[K,S,E] = dlqr(G,H,Q,R);
>>E %แสดงค่า Eigen value
>>K %แสดงค่า เกน K ที่คำนวณได้
>>fid = fopen('c:\data\Kf.dat','wb+'); %เก็บค่าเกน K ลงไฟล์
>>fwrite(fid,K(1),'float');
>>fwrite(fid,K(2),'float');
>>fwrite(fid,K(3),'float');
>>fwrite(fid,K(4),'float');
>>fwrite(fid,K(5),'float');
>>fclose(fid);

```

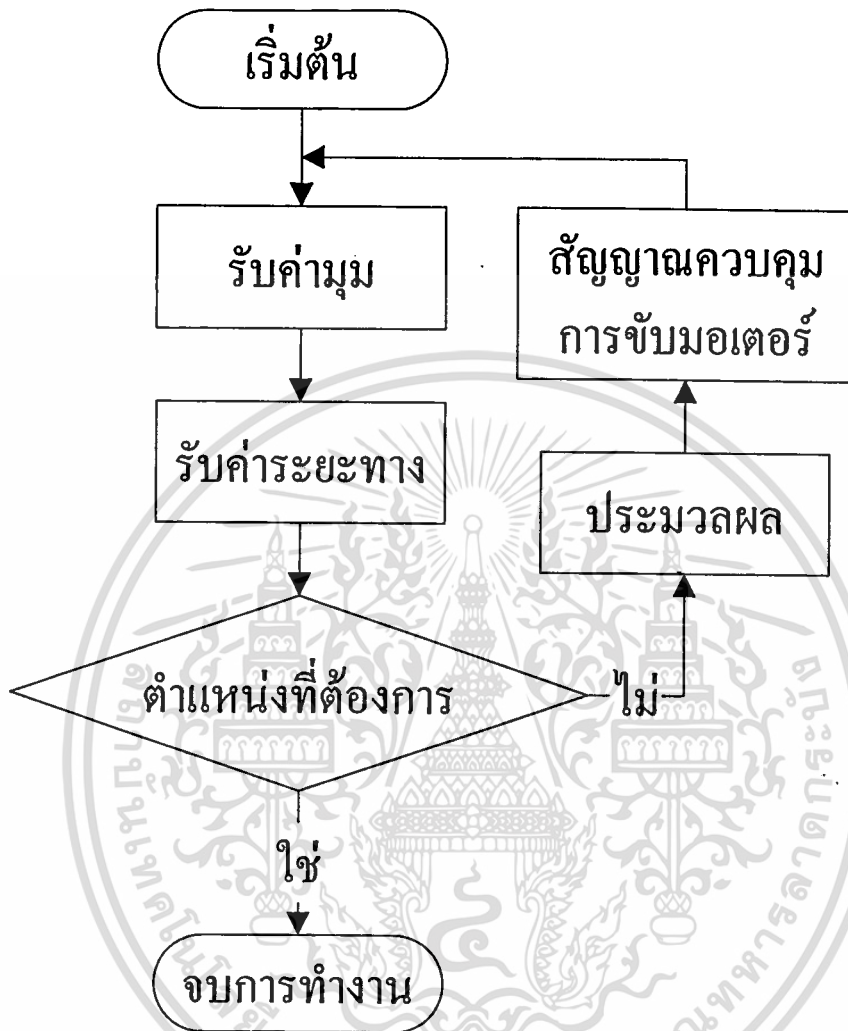


ภาคผนวก ค
โปรแกรมควบคุม
การเคลื่อนที่ของ Керน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

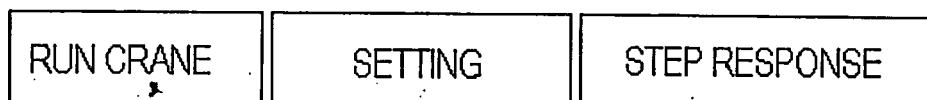
ภาคผนวก ค

โปรแกรมควบคุมการเคลื่อนที่ของเครน



รูปที่ ค.1 Flow Chart ของ โปรแกรมการควบคุมการเคลื่อนที่ของเครน

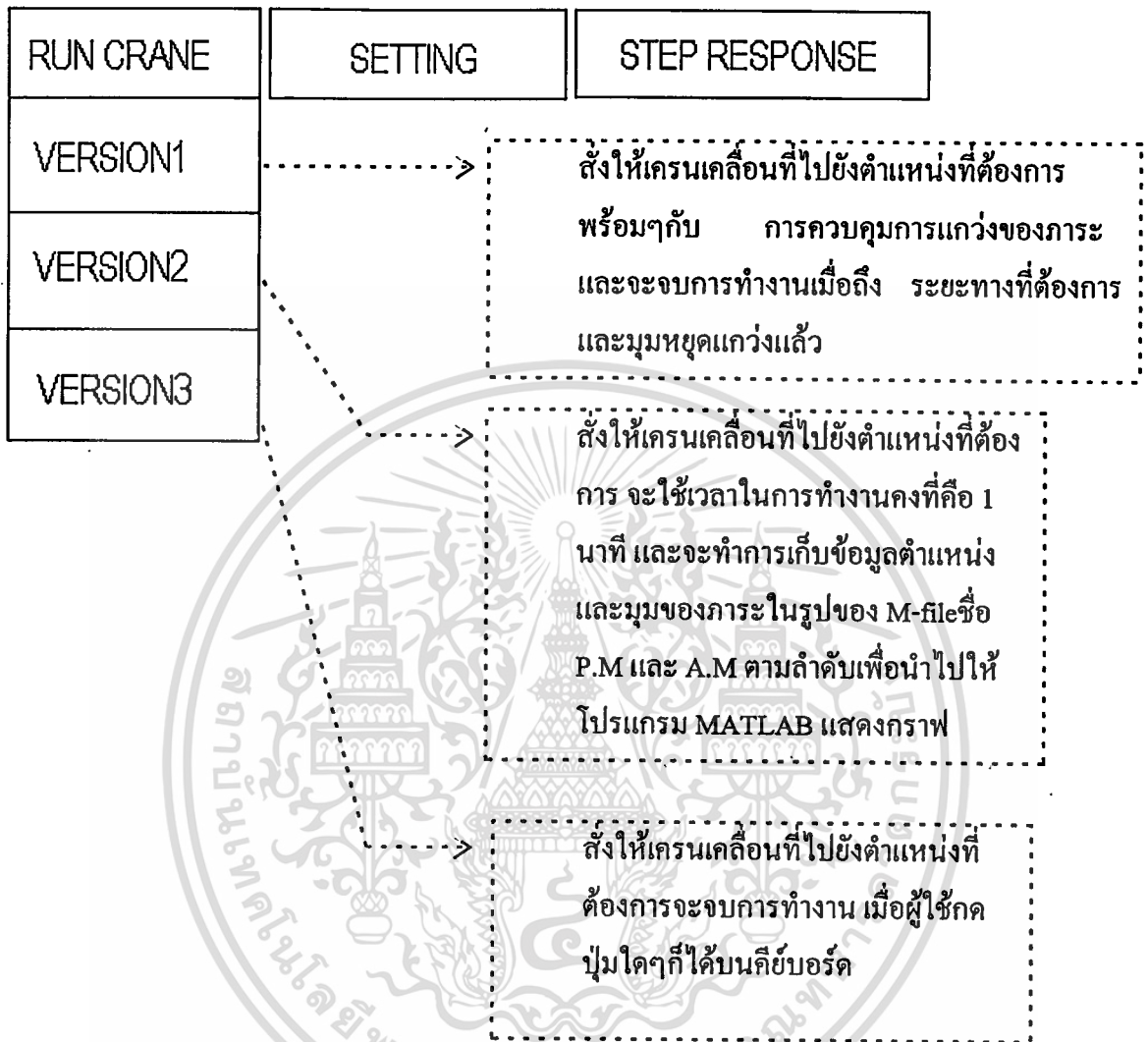
โปรแกรมที่ใช้มีชื่อว่า “CMC3” มีการติดต่อกับผู้ใช้ในลักษณะของเมนูดังรูปที่ ค.2



รูปที่ ค.2 ลักษณะเมนูของโปรแกรมCMC3.EXE

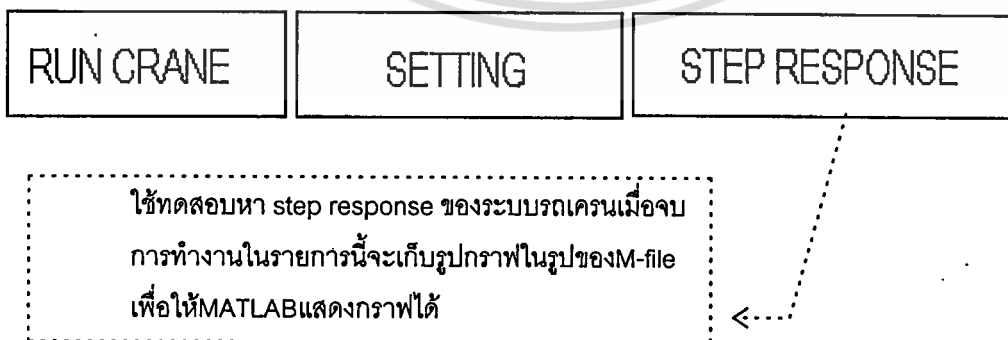
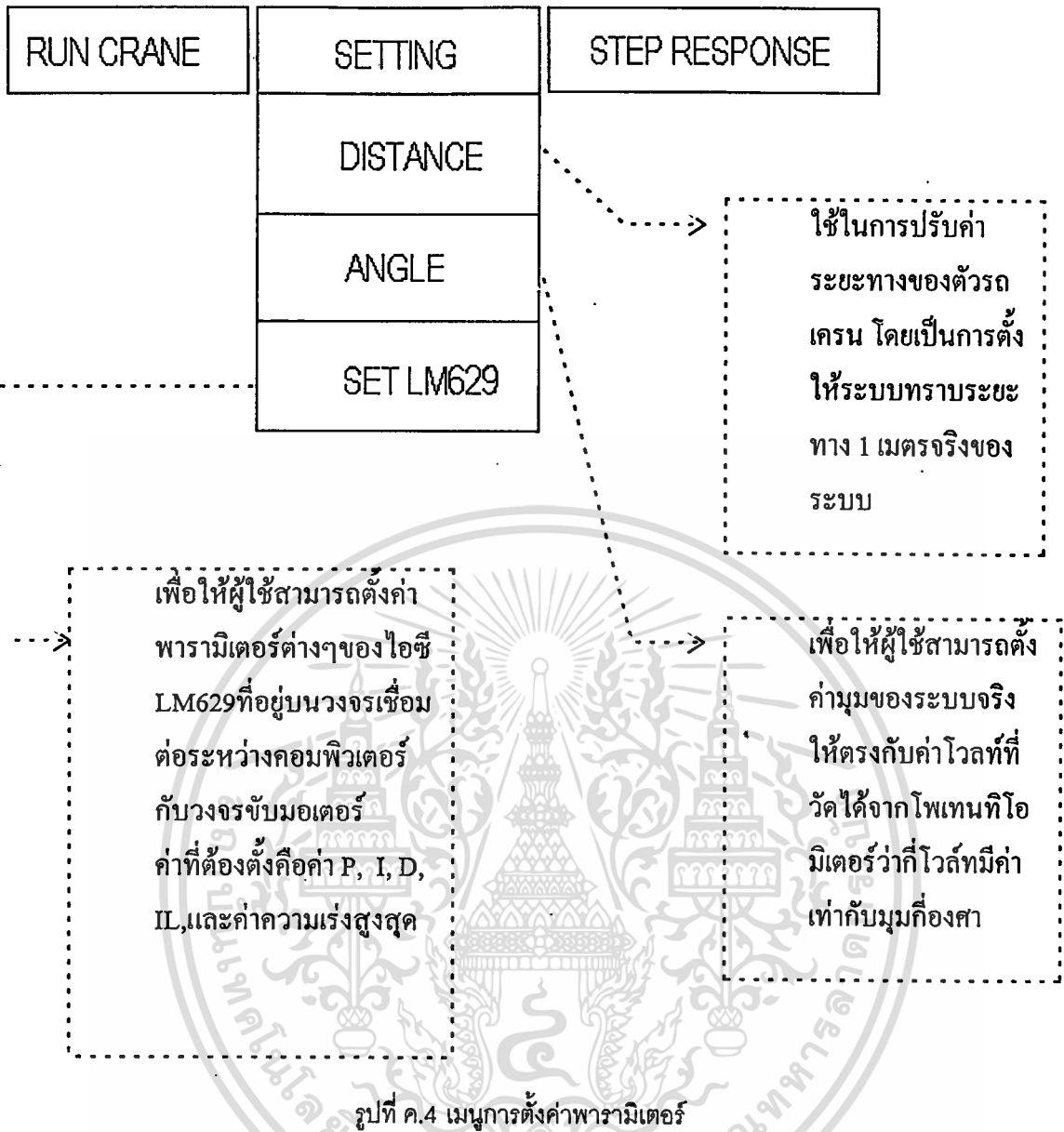
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรม CMC3 ทำได้โดยการเลือกเมนูด้วยปุ่มลูกศร ขึ้น-ลง-ซ้าย-ขวา ของคีย์บอร์ด คอมพิวเตอร์ เพื่อเลื่อนไปยังรายการที่ต้องการ รายการต่างๆมีดังต่อไปนี้



รูปที่ ค.3 เมนูการเลือกเวอร์ชันในการเริ่มโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Source Code

โปรแกรมที่ใช้ทั้งหมดประกอบด้วย 10 โปรแกรมด้วยกันคือ

1. CMC3.CPP เป็นตัวสร้างหน้าจอเมนูให้ผู้ใช้สามารถเรียกใช้งานโปรแกรมย่อยอื่นๆได้
2. VERSION1.CPP โปรแกรมรับค่าระยะทางจากผู้ใช้และส่งค่าไปควบคุมระบบตามทฤษฎี โดยโปรแกรมจะหยุดทำงานเมื่อภาระถึงระยะทางที่ตั้งและมุมหยุดแกว่ง
3. VERSION2.CPP โปรแกรมรับค่าระยะทางจากผู้ใช้และส่งค่าไปควบคุมระบบตามทฤษฎี โดยโปรแกรมจะหยุดทำงานครบเวลา 1 นาที โดยเมื่อจบการทำงานโปรแกรมจะสร้าง M-file ชื่อ “a.m” และ “p.m” ให้ MATLAB อ้างอิงเพื่อแสดงกราฟ
4. VERSION3.CPP โปรแกรมรับค่าระยะทางจากผู้ใช้และส่งค่าไปควบคุมระบบตามทฤษฎี โดยโปรแกรมจะหยุดทำงานเมื่อผู้ใช้กดปุ่มใดๆบนคีย์บอร์ด
5. DISTANCE.CPP โปรแกรมปรับค่าระยะทางให้ตรงกับความเป็นจริง
6. ANGLE.CPP โปรแกรมปรับค่ามุมให้ตรงกับความเป็นจริง
7. SETLM629.CPP โปรแกรมตั้งค่า P, I, D, IL ให้กับ
8. STEPTEST.CPP โปรแกรมส่งค่าความเร็วคงที่ไปยังระบบเพื่อทดสอบ step response และบันทึกค่าใน M-file ชื่อ “c_goto2.m”
9. C_TEST2.H เป็นเฮดเดอร์ไฟล์ใช้ติดต่อกับไอซี LM629
10. ADC.H เป็นเฮดเดอร์ไฟล์ใช้ติดต่อกับการ์ด PCL714

CMC3.CPP

```

#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <bios.h>
#include <process.h>
#include <stdlib.h>

//----- DEFINE BUTTON -----//
#define UP_ARROW      0x4800
#define DOWN_ARROW   0x5000
#define LEFT_ARROW   0x4B00
#define RIGHT_ARROW  0x4D00
#define RETURN       0x1C0D
#define ESCAPE       0x011B
typedef struct
{
    int row;
    int col;
}event_type;

void WRITE_MAIN_MENU(event_type item);
event_type CHECK_KEY_MAP_IN_PULL_DOWN_MENU(event_type item);
void DRAW_PULL_DOWN_IN_COL1(void);
void DRAW_PULL_DOWN_IN_COL2(void);
void DRAW_PULL_DOWN_IN_COL3(void);

event_type CHECK_CHEGNGE_ROW(event_type item);
event_type CHANGE_COLUMN(event_type item);
void DISPLAY_PULL_DOWN_MENU(event_type item);
void CALL_SUBROUTINE(event_type item);

void main(void)
{
    event_type item;
    item.row = 1;
    item.col = 1;
    int key;
    int event;
    clrscr();

    WRITE_MAIN_MENU(item); //item = PULL_DOWN_IN_COL1(item);

    item = CHECK_KEY_MAP_IN_PULL_DOWN_MENU(item);
    getch();
    clrscr();
}
//-----//

void WRITE_MAIN_MENU(event_type item)
{
    _setcursortype(_NOCURSOR);
    textcolor(9);
    gotoxy(1,1);
    printf("_____ \n"); //13x3
    gotoxy(1,2);
    printf("| \n");
    gotoxy(1,3);
    printf("_____ \n");
    gotoxy(2,2);
    printf("_____ \n");
    if( (item.col == 1) && (item.row == 1) )
    {
        textcolor(4);
    }
    else
    {
        textcolor(11);
    }
    printf(" RUN CRANE");

    textcolor(9);
    gotoxy(14,1);
    printf("_____ \n"); //13x3
    gotoxy(14,2);
    printf("| \n");
    gotoxy(14,3);
    printf("_____ \n");
    gotoxy(15,2);
    printf("_____ \n");
    if( (item.col == 2) && (item.row == 1) )
    {
        textcolor(4);
    }
    else
    {
        textcolor(11);
    }
    printf(" SETTING ");

    textcolor(9);
    gotoxy(27,1);
    printf("_____ \n"); //17x3
    gotoxy(27,2);
    printf("| \n");
    gotoxy(27,3);
    printf("_____ \n");
    gotoxy(28,2);
    printf("_____ \n");
    if( (item.col == 3) && (item.row == 1) )
    {
        textcolor(4);
    }
    else
    {
        textcolor(11);
    }
    printf(" STEP RESPONSE");
} //end WRINT_MAIN_MENU()

event_type CHECK_KEY_MAP_IN_PULL_DOWN_MENU(event_type item)
{
    int key;
    for(;;)
    {
        key = bioskey(0);
        clrscr();
        WRITE_MAIN_MENU(item);
        switch(key)

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        case DOWN_ARROW : item.row++;
                        item = CHECK_CHEGNGE_ROW(item);
                        WRITE_MAIN_MENU(item);
                        DISPLAY_PULL_DOWN_MENU(item);
                        break;

        case UP_ARROW : item.row--;
                        item = CHECK_CHEGNGE_ROW(item);
                        WRITE_MAIN_MENU(item);
                        DISPLAY_PULL_DOWN_MENU(item);
                        break;

        case LEFT_ARROW : item.col--;
                        item = CHANGE_COLUMN(item);
                        WRITE_MAIN_MENU(item);
                        DISPLAY_PULL_DOWN_MENU(item);
                        break;

        case RIGHT_ARROW: item.col++;
                        item = CHANGE_COLUMN(item);
                        WRITE_MAIN_MENU(item);
                        DISPLAY_PULL_DOWN_MENU(item);
                        break;

        case RETURN : clrscr();
                        //textcolor()
                        //textmode(BW80);
                        setcursortype(NORMALCURSOR);
                        CALL_SUBROUTINE(item);
                        item.row = 1;
                        //item.col = 1;
                        WRITE_MAIN_MENU(item);
                        break;

        case ESCAPE : _setcursortype(NORMALCURSOR);
                        clrscr();
                        exit(0);
                        break;
    };

    //end for(;;)

    return item;
//end CHECK_KEY_MAP_IN_PULL_DOWN_MENU()

void DRAW_PULL_DOWN_IN_COL1(void)
{
    int key;

    textcolor(9);
    gotoxy(1,3);
    printf("-----|\\n "); //13x3
    gotoxy(1,4);
    printf("|\\n ");
    gotoxy(1,5);
    printf("-----|\\n ");
    gotoxy(1,6);
    printf("|\\n ");
    gotoxy(1,7);
    printf("-----|\\n ");
    gotoxy(1,8);
    printf("|\\n ");
    gotoxy(1,9);
    printf("-----|\\n ");

    gotoxy(2,4);
    textcolor(11);
    printf(" VERSION 1 ");
    gotoxy(2,6);
    textcolor(11);
    printf(" VERSION 2 ");
    gotoxy(2,8);
    textcolor(11);
    printf(" VERSION 3 ");
}

void DRAW_PULL_DOWN_IN_COL2(void)
{
    textcolor(9);
    gotoxy(14,3);
    printf("-----|\\n "); //13x3
    gotoxy(14,4);
    printf("|\\n ");
    gotoxy(14,5);
    printf("-----|\\n ");
    gotoxy(14,6);
    printf("|\\n ");
    gotoxy(14,7);
    printf("-----|\\n ");
    gotoxy(14,8);
    printf("|\\n ");
    gotoxy(14,9);
    printf("-----|\\n ");

    gotoxy(15,4);
    textcolor(11);
    printf(" DISTANCE ");
    gotoxy(15,6);
    textcolor(11);
    printf(" ANGLE ");
    gotoxy(15,8);
    textcolor(11);
    printf(" SET LM629");
}

void DRAW_PULL_DOWN_IN_COL3(void)
{
}

event_type CHECK_CHEGNGE_ROW(event_type item)
{
    switch(item.col)
    {
        case 1 : switch(item.row)
                {
                    case 0 : item.row = 4;
                            break;
                    case 5 : item.row = 1;
                            break;
                };
                break;

        case 2 : switch(item.row)
                {
                    case 0 : item.row = 4;
                            break;
                    case 5 : item.row = 1;
                            break;
                };
                break;
    };
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

case 1 : break;
case 2 : spawnl(P_WAIT, "\\b427\\tc\\1\\distance.exe", NULL);
        clrscr();
        //SET_DISTANCE();
        break;
case 3 : spawnl(P_WAIT, "\\b427\\tc\\1\\angle.exe", NULL);
        clrscr();
        //SET_ANGLE();
        break;
case 4 : spawnl(P_WAIT, "\\b427\\tc\\1\\setlm629.exe", NULL);
        clrscr();
        //SET_LM629_PID();
        break;
    };
    break;
case 3 : switch(item.row)
    {
        case 1 : spawnl(P_WAIT, "\\b427\\tc\\1\\steptest.exe", NULL);
                //RUN_STEP();
                clrscr();
                break;
    };
    break;
}; //end switch(item.col)
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION1.CPP

```

//version1.cpp ==> n42.Cpp ==> lqr.c with new state state space

//----- STANDARD HEADER FILES -----//
#include<dos.h>
#include<conio.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include <bios.h>
#include <stdlib.h>
#include <process.h>

//----- OUR HEADER FILE -----//
#include "c:\b427\tc\include\adc.h"
#include "c:\b427\tc\include\c_test2.h"

//----- DEFINE CONSTANT MACRO -----//
#define PERIOD 0.1
#define LOW1 2500 //value to load to conuter2 8253
#define LOW2 2500
#define PI 3.14159265359
#define X 16
#define Y 5
#define MAXDATA 4

#define cranefile "c:\\data\\N42.dat" //response data file
#define crane M "c:\\data\\m\\p.m" //m file for running in matlab
#define crane_ang_M "c:\\data\\m\\a.m"
#define Kf_source "c:\\data\\Kf.dat"
#define POTENTIO_FILE "c:\\data\\potentio.dat" //response data file

struct feedBackGain
{
    float angle;
    float difAngle;
    float pos;
    float difPos;
    float difDifPos;
};

typedef struct{
    float xMax; //Max Time Operation (sec)
    float yMax; //Set Point Position (cm)
    float sampling; //Sampling Period (sec)
    char *heading; //Heading of Graph (string)
    char *xLabel; //Label of X Axis (string)
    char *yLabel; //Label of Y Axis (string)
}graphInfo;
typedef struct{
    float secPpix;
    int pixPsampling;
}scaleX;
typedef struct{
    float cmPpix;
}scaleY;
typedef struct{
    scaleX x;
    scaleY y;
}axisScale;

struct Conv_type
{
    float posC2M;
    float velM2C;
};

float READ_DATA(void);
Conv_type OPEN_CONVERT_DATA(void);
void main(void)
{
//----- DEFINE VARIABLE -----//
clrscr();
Conv_type convert;
//convert
convert.posC2M = OPEN_CONVERT_DATA();
convert.velM2C = 691176.5596;
printf("\nposC2M = %e",convert.posC2M);
printf("\nvelM2C = %e",convert.velM2C);
getch();

word adc;
int mode = PACER; //select PACER Mode to use as Sampling Time of Process
int base = 0x200; //select address H200 to use as BASE Address
int ch = 12; //select channel 12 to use
counterRegister counter;
float offset;
//----- read ratio from potentio.dat -----//

float volt2angle = 7.5;//READ_DATA();// set from angle.exe
printf("\n volt2angle = %4.4f",volt2angle);
getch();
//-----//

graphInfo info;
axisScale scale;
float maxDistance = 1.0; //Set Maximum Value In Y-Axis of Graph
float samplingTime,maxTime=60.0; //Set Maximum Value In X-Axis of Graph

int k = 0;
long time = 0;
long startProcess = 0, startLoop = 0, end = 0;

float velocity = 0.0;
float loadPos;
double radian;

float P1,P2,P3;

FILE *datafile;
FILE *data_M;
FILE *data_ang_M;
FILE *Kf_file;

float volt[MAXDATA],realAngle[MAXDATA],realPos[MAXDATA];
float realDifAngle[MAXDATA],realDifPos[MAXDATA],realDifDifPos[MAXDATA];

float pos[MAXDATA], angle[MAXDATA];
float difPos[MAXDATA],difDifPos[MAXDATA],difAngle[MAXDATA];
float controlLaw[MAXDATA];

```

เอกสารนี้เป็นลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float f1, f2, f3, f4, f5;
float Ko[7][4], L[7][4];
float G[7][7], H[7][3];
float errorPos[MAXDATA], errorAngle[MAXDATA];

float estimAngle[MAXDATA], estimPos[MAXDATA], estimDifAngle[MAXDATA];
float estimDifPos[MAXDATA], estimDifDifPos[MAXDATA];
float ePos[MAXDATA], eAngle[MAXDATA];

float filAngle[MAXDATA];
float a1[5], b1[5], a2[5], b2[5], a3[5], b3[5], a4[5], b4[5], a5[5], b5[5], a6[5], b6[5];

float angleRef = 0.0; //Set Value of ANGLE REFERENCE
char ON_THE_LEFT = 0;

//----- INITIAL FEEDBACK GAIN -----//
//----- CALCULATE FORM Q = diag([100 1 10000 100 10]), R = diag([100 100]) see Example in chapter 5

Kf_file = fopen(Kf_source, "rb");
fread(&Kf, sizeof(struct feedBackGain), 1, Kf_file);
clrscr();
printf("\nKf.angle           = %4.4f", Kf.angle);
printf("\nKf.difAngle        = %4.4f", Kf.difAngle);
printf("\nKf.pos                 = %4.4f", Kf.pos);
printf("\nKf.difPos              = %4.4f", Kf.difPos);
printf("\nKf.difDifPos           = %4.4f", Kf.difDifPos);
getch();
clrscr();

fclose(Kf_file);

//----- receive destination point -----//
float posRef; // = 1.0; //Set Value of POSITION REFERENCE
char y_n_e = 'n';
do
{
    printf("\n Enter distance :           m.");
    gotoxy(18, 2);
    scanf("%f", &posRef);
    printf("\n distance = %4.4f (Yes/No/Exit)", posRef);
    cscanf("%c", &y_n_e);

    if( (y_n_e=='e') || (y_n_e=='E') )
    {
        exit(1);
    }
    clrscr();
}while( (y_n_e!='y') && (y_n_e!='Y') );

float display_errorPos[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
int display_k[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
float display_controlLaw[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
int j = 0;
long count = 0;
float rad = 0;
float ang = 0;

//----- START PROGRAM -----//
clrscr();
//----- Set info of graph -----//
info.xMax = maxTime;
info.yMax = maxDistance;
info.sampling = PERIOD;
info.heading = "Response of Load Position ";

info.xLabel = "Time (sec)";
info.yLabel = "Distance (cm)";
samplingTime = info.sampling;

//----- INITIAL DATA -----//
for(k=0; k<MAXDATA; k++)
{
    volt[k] = 0; //Voltage Reading from ADC On PCI-714 Card
    realAngle[k] = 0; //Real Angle Value Convert from Voltage
    realPos[k] = 0; //Real Position Value Reading from IM629
    realDifAngle[k] = 0; //Value Compute On Physics Relation
    realDifPos[k] = 0; //Value Compute On Physics Relation
    realDifDifPos[k] = 0; //Value Compute On Physics Relation
    angle[k] = 0; //Value After Measurement Update
    difAngle[k] = 0; //Value After Measurement Update
    pos[k] = 0; //Value After Measurement Update
    difPos[k] = 0; //Value After Measurement Update
    difDifPos[k] = 0; //Value After Measurement Update
    estimAngle[k] = 0; //Value Before Measurement Update
    estimPos[k] = 0; //Value Before Measurement Update
    estimDifAngle[k] = 0; //Value Before Measurement Update
    estimDifPos[k] = 0; //Value Before Measurement Update
    estimDifDifPos[k] = 0; //Value Before Measurement Update
    controlLaw[k] = 0; //Control Law
    errorPos[k] = 0;
    errorAngle[k] = 0;
    ePos[k] = 0; //realPos-estimatePos
    eAngle[k] = 0; //realAngle-estimateAngle
    filAngle[k] = 0; //Angle Vale That has Filtered from Band Pass Filter
}

// PAGE 141
//----- BAND-PASS FILTER FOR FILTER NOISE FORM ANGLE -----//
a1[1]=1.0000;a1[2]=0.0309;a1[3]=-0.9691;
b1[1]=0.9845;b1[2]=0.0000;b1[3]=-0.9845;

counter=COUNTER_REGISTER(PERIOD*1000000);
//----- INITIAL LM629 -----//
INIT_LM629(); //initial lm629
clrscr();

SET_PID_FILTER(50, 7, 28, 20000);
SET_ACCELERATION_OUT(0x10F); //0x100F);

//----- INITIAL 8253 COUNTER -----//
SET_COUNTER(base, counter); //load sampling time and start counting

startProcess = clock(); //read start Process clock
startLoop = startProcess; //read start Loop clock

//----- START LOOP CONTROL -----//
clrscr();
gotoxy(1, 3);
printf(" k      errorPos[k]    controlLaw[k]    ang");
k = 0;
clock_t prev_time, pres_time;
prev_time = clock();
//----- FIND OFFSET VOLTAGE -----//
TRIG(mode, base, ch, counter, k); //select channel and trig ADC to read data
adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time

offset = ((adc.high*256)+adc.low)-8192;
offset = (offset*10)/16384; //convert data to voltage value
char begin_step1 = 1;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

display_k[14] = display_k[15];
display_k[15] = display_k[16];
display_k[16] = display_k[17];
display_k[17] = display_k[18];
display_k[18] = display_k[19];
display_k[19] = k;

display_errorPos[0] = display_errorPos[1];
display_errorPos[1] = display_errorPos[2];
display_errorPos[2] = display_errorPos[3];
display_errorPos[3] = display_errorPos[4];
display_errorPos[4] = display_errorPos[5];
display_errorPos[5] = display_errorPos[6];
display_errorPos[6] = display_errorPos[7];
display_errorPos[7] = display_errorPos[8];
display_errorPos[8] = display_errorPos[9];
display_errorPos[9] = display_errorPos[10];
display_errorPos[10] = display_errorPos[11];
display_errorPos[11] = display_errorPos[12];
display_errorPos[12] = display_errorPos[13];
display_errorPos[13] = display_errorPos[14];
display_errorPos[14] = display_errorPos[15];
display_errorPos[15] = display_errorPos[16];
display_errorPos[16] = display_errorPos[17];
display_errorPos[17] = display_errorPos[18];
display_errorPos[18] = display_errorPos[19];
display_errorPos[19] = errorPos[1];

display_controlLaw[0] = display_controlLaw[1];
display_controlLaw[1] = display_controlLaw[2];
display_controlLaw[2] = display_controlLaw[3];
display_controlLaw[3] = display_controlLaw[4];
display_controlLaw[4] = display_controlLaw[5];
display_controlLaw[5] = display_controlLaw[6];
display_controlLaw[6] = display_controlLaw[7];
display_controlLaw[7] = display_controlLaw[8];
display_controlLaw[8] = display_controlLaw[9];
display_controlLaw[9] = display_controlLaw[10];
display_controlLaw[10] = display_controlLaw[11];
display_controlLaw[11] = display_controlLaw[12];
display_controlLaw[12] = display_controlLaw[13];
display_controlLaw[13] = display_controlLaw[14];
display_controlLaw[14] = display_controlLaw[15];
display_controlLaw[15] = display_controlLaw[16];
display_controlLaw[16] = display_controlLaw[17];
display_controlLaw[17] = display_controlLaw[18];
display_controlLaw[18] = display_controlLaw[19];
display_controlLaw[19] = controlLaw[1];
}

//----- ROTATE DATA -----//
realAngle[3] = realAngle[2];
filAngle[3] = filAngle[2];

realAngle[2] = realAngle[1];
filAngle[2] = filAngle[1];
realPos[2] = realPos[1];
realDifPos[2] = realDifPos[1];
//----- CHECK OUT -----//
if( fabs(errorPos[1]) <= 0.002 //check errorPos
{
    pos_check_stop = 1;
}
else
{
    pos_check_stop = 0;
}

if(fabs(difAngle[1]) <= 0.001 //check difAngle
{
    dif_ang_check_stop = 1;
}
else
{
    dif_ang_check_stop = 0;
}
if( fabs(filAngle[1]) <= 0.04 )
{
    ang_check_stop = 1;
}
else
{
    ang_check_stop = 0;
}

if( (pos_check_stop==1)&&(ang_check_stop==1)&&(dif_ang_check_stop==1) )
{
    set_point_check_stop = 1;
}
else
{
    set_point_check_stop = 0;
}

if( (set_point_check_stop==1)|| (bioskey(1)) )
{
    EXIT = 1;
}
else
{
    EXIT = 0;
}

//-----//
}while(EXIT == 0); //END CONTROL LOOP
VELOCITY_OUT(0, convert.velM2C);
INIT_IM629();
//cliEscr();
end = clock(); //read clock when end routine
time = (long)((end-startProcess)/CLK_TCK*1000000);
//gotoxy(X-6,Y);
printf("\nThis Process use time = %10.0ld microsec ",time);
//-----//
printf("\n\nPress any key to quit program .");
getch();
}

float READ_DATA(void)
{
    struct angle_type
    {
        float angle;
        float volt;
        float ratio;
    }data;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FILE *potentio_file;
potentio_file = fopen(POTENTIO_FILE,"rb");
fread(&data,sizeof(struct angle_Type),1,potentio_file);
fclose(potentio_file);

clrscr();
textcolor(7);
printf("%4.4f degrees = %4.4f volt \n(ratio = %4.4f / %4.4f = %4.4f)",data.angle, data.volt, data.angle, data.volt,
data.ratio );

return data.ratio;
}

Conv_type OPEN_CONVERT_DATA(void)
{
Conv_type convert;
FILE *data_file;
data_file = fopen(ConvertFile,"rb");
fread(&convert, sizeof(struct Conv_type),1,data_file);
fclose(data_file);

return convert;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION2.CPP

```

//----- STANDARD HEADER FILES -----//
#include<dos.h>
#include<conio.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include <bios.h>
#include <stdlib.h>
#include <process.h>

//----- OUR HEADER FILE -----//
#include "c:\b427\tc\include\adc.h"
#include "c:\b427\tc\include\c_test2.h"

//----- DEFINE CONSTANT MACRO -----//
#define PERIOD 0.1
#define LOW1 2500 //value to load to conuter2 8253
#define LOW2 2500
#define PI 3.14159265359
#define X 16
#define Y 5
#define MAXDATA 600

#define crane_M "c:\\data\\m\\p.m" //m file for running in matlab
#define crane_ang_M "c:\\data\\m\\a.m"
#define Kf source "c:\\data\\Kf.dat"
#define POTENTIO_FILE "c:\\data\\potentio.dat" //response data file

struct feedBackGain
{
    float angle;
    float difAngle;
    float pos;
    float difPos;
    float difDifPos;
};

typedef struct{
    float xMax; //Max Time Operation (sec)
    float yMax; //Set Point Position (cm)
    float sampling; //Sampling Period (sec)
    char *heading; //Heading of Graph (string)
    char *xLabel; //Label of X Axis (string)
    char *yLabel; //Label of Y Axis (string)
}graphInfo;
typedef struct{
    float secPpix;
    int pixPsampling;
}scaleX;
typedef struct{
    float cmPpix;
}scaleY;
typedef struct{
    scaleX x;
    scaleY y;
}axisScale;

struct Conv_type
{
    float posC2M;
    float velM2C;
}convert;

int Stop = 1;

float READ_DATA(void);
Conv_type OPEN_CONVERT_DATA(void);
void main(void)
{
//----- DEFINE VARIABLE -----//
clrscr();
//Conv_type convert;
//convert = OPEN_CONVERT_DATA();
convert.posC2M = 3.236455E-5;
convert.velM2C = 691176.5593;
printf("\nposC2M = %f",convert.posC2M);
printf("\nvelM2C = %f",convert.velM2C);
getch();

word adc;
int mode = PACER; //select PACER Mode to use as Sampling Time of Process
int base = 0x200; //select address H220 to use as BASE Address
int ch = 12; //select channel 12 to use
counterRegister counter;
float offset;
float volt2angle = 7.5;///READ DATA();
printf("\nvolt2angle = %4.4f",volt2angle);
getch();

graphInfo info;
axisScale scale;
float maxDistance = 1.0; //Set Maximum Value In Y-Axis of Graph
float samplingTime,maxTime=60.0; //Set Maximum Value In X-Axis of Graph

int k = 0;
long time = 0;
long startProcess = 0, startLoop = 0, end = 0;

float velocity = 0.0;
float loadPos;
double radian;

// Page 139
float P1[MAXDATA], P2[MAXDATA], P3[MAXDATA];

FILE *datafile;
FILE *data_M;
FILE *data_ang_M;
FILE *Kf_file;

float volt [MAXDATA], realAngle [MAXDATA], realPos [MAXDATA];
float realDiFAngle [MAXDATA], realDifPos [MAXDATA], realDifDifPos [MAXDATA];

float pos [MAXDATA], angle [MAXDATA];
float difPos [MAXDATA], difDifPos [MAXDATA], difAngle [MAXDATA];
float controlLaw [MAXDATA];

float f1, f2, f3, f4, f5;
float Ko[7] [4], L[7] [4];
float G[7] [7], H[7] [3];
float errorPos [MAXDATA], errorAngle [MAXDATA];

```

เอกสารนี้จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float ePos[MAXDATA], eAngle[MAXDATA];

float filAngle[MAXDATA];
float a1[5], b1[5], a2[5], b2[5], a3[5], b3[5], a4[5], b4[5], a5[5], b5[5], a6[5], b6[5];

float angleRef = 0.0; //Set Value of ANGLE REFERENCE

//----- INITIAL FEEDBACK GAIN -----//
//----- CALCULATE FORM Q = diag([100 1 10000 100 10]), R = diag([100 100]) see Example in chapter 5

feedBackGain Kf;
Kf_file = fopen(Kf_source, "rb");
fread(&Kf, sizeof(struct feedBackGain), 1, Kf_file);
fclose(Kf_file);
clrscr();
printf("\nKf.angle = %4.4f", Kf.angle);
printf("\nKf.difAngle = %4.4f", Kf.difAngle);
printf("\nKf.pos = %4.4f", Kf.pos);
printf("\nKf.difPos = %4.4f", Kf.difPos);
printf("\nKf.difDifPos = %4.4f", Kf.difDifPos);
getch();
clrscr();

//----- receive destination point -----//
float posRef; // = 1.0; //Set Value of POSITION REFERENCE
char y_n_e = 'n';
do
{
    //clrscr();
    //
    printf("\n Enter distance : m.");
    gotoxy(18, 2);
    scanf("%f", &posRef);
    printf("\n distance = %4.4f (Yes/No/Exit)", posRef);
    cscanf("%c", &y_n_e);

    if( (y_n_e=='e') || (y_n_e=='E') )
    {
        exit(1);
    }
    clrscr();
}while( (y_n_e=='y') && (y_n_e=='Y') );

//-----//

float display_errorPos[20] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 };
int display_k[20] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 };
float display_controlLaw[20] = { 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 };
int j = 0;
long count = 0;
float rad = 0;
float ang = 0;

//----- START PROGRAM -----//
clrscr();
//----Set Info of graph -----//
info.xMax = maxTime;
info.yMax = maxDistance;
info.sampling = PERIOD;
info.heading = "Response of Load Position ";

// PAGE 140
info.xLabel = "Time (sec)";
info.yLabel = "Distance(cm)";
samplingTime = info.sampling;

//----- INITIAL DATA -----//
for(k=0; k<MAXDATA; k++)
{
    volt[k] = 0; //Voltage Reading from ADC on PCL-714 Card
    realAngle[k] = 0; //Real Angle Value Converse from Voltage
    realPos[k] = 0; //Real Position Value Reading from LM629
    realDifAngle[k] = 0; //Value Compute On Physics Relation
    realDifPos[k] = 0; //Value Compute On Physics Relation
    realDifDifPos[k] = 0; //Value Compute On Physics Relation
    angle[k] = 0; //Value After Measurement Update
    difAngle[k] = 0; //Value After Measurement Update
    pos[k] = 0; //Value After Measurement Update
    difPos[k] = 0; //Value After Measurement Update
    difDifPos[k] = 0; //Value After Measurement Update
    controlLaw[k] = 0; //Control Law
    errorPos[k] = 0;
    errorAngle[k] = 0;
    eAngle[k] = 0; //realAngle-estimateAngle
    filAngle[k] = 0; //Angle Vale That has Filtered from Band Pass Filter
}

// PAGE 141
//----- BAND-PASS FILTER FOR FILTER NOISE FORM ANGLE -----//
a1[1]=1.0000;a1[2]=0.0309;a1[3]= -0.9691;
b1[1]=0.9845;b1[2]=0.0000;b1[3]= -0.9845;

counter=COUNTER_REGISTER(PERIOD*1000000);
//----- INITIAL LM629 -----//
INIT_LM629(); //initial lm629
clrscr();

SET_PID_FILTER(50,7,28,20000 );
SET_ACCELERATION_OUT(0x10F); //0x100F;
//----- INITIAL 8253 COUNTER -----//
SET_COUNTER(base, counter); //load sampling time and start counting

startProcess = clock(); //read start Process clock
startLoop = startProcess; //read start Loop clock
//----- START LOOP CONTROL -----//
clrscr();
gotoxy(1, 3);
printf(" k errorPos[k] controlLaw[k] ang");
k = 0;
clock_t prev_time, pres_time;
prev_time = clock();

do
{
    //----- READ ANGLE FROM ADC -----//
    k++;
    TRIG(mode, base, ch, counter, k); //select channel and trig ADC to read data
    adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time

    //PAGE 143
    volt[k] = ((adc.high*256+adc.low)-8192);
    volt[k] = (volt[k]*10)/16384; //converse data to voltage value
    if(k==1) offset = volt[1];
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ การศึกษาเท่านั้น ไม่สามารถเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

volt[k] = volt[k] - offset; //automatic adjust offset
realAngle[k] = ((volt[k]*volt2angle)*2*PI)/360; //from voltage to "radian"

//----- FILTER ANGLE BEFORE COMPUTE -----//
if(k==1)
{
    filAngle[k] = bl[1]*realAngle[k];
}
if(k==2)
{
    filAngle[k] = bl[1]*realAngle[k] + bl[2]*realAngle[k-1] - a1[2]*filAngle[k-1];
}
if((k!=1)&&(k!=2))
{
    filAngle[k]=bl[1]*realAngle[k]+bl[2]*realAngle[k-1] + bl[3]*realAngle[k-2]-
a1[2]*filAngle[k-1]-a1[3]*filAngle[k-2];
}
//----- READ POSITION FROM LM629 -----//
realPos[k] = READ_REAL_POS(convert.posC2M); //read real position from lm629

//----- STATE ESTIMATOR CALCULUS DifAngle DifPos DifDifPos FROM PHYSICS RELATION FOR LQR -----//
if(k==1)
{
    realDifPos[k] = 0;
    realDifDifPos[k] = 0;
    realDifAngle[k] = 0;
}
else
{
    realDifPos[k]=(realPos[k] - realPos[k-1])/info.sampling;
    realDifDifPos[k]=(realDifPos[k]-realDifPos[k-1])/info.sampling;
    realDifAngle[k] = (realAngle[k]-realAngle[k-1])/info.sampling;
}
/*
if(k==2)
{
    realDifDifPos[k] = 0;
}
else
{
    realDifDifPos[k] = (realDifPos[k] - realDifPos[k-1])/info.sampling;
}
*/
errorAngle[k] = angleRef - filAngle[k];
errorPos[k] = posRef - realPos[k];

//----- COMPUTE CONTROL LAW (VELOCITY COMMAND) -----//
//----- ESTIMATE STATE FROM PHYSICS RELATION -----//
f2 = realDifAngle[k]*Kf.difAngle;
f3 = realDifPos[k]*Kf.difPos;
f4 = realDifDifPos[k]*Kf.difDifPos;
P1[k] = Kf.angle*errorAngle[k];
P2[k] = Kf.pos*errorPos[k];
P3[k] = f2+f3+f4;

//controlLaw[k] = (Kf.angle*errorAngle[k]) + (Kf.pos*errorPos[k])-(f2+f3+f4);
controlLaw[k] = P1[k] + P2[k] - P3[k];
//----- SEND VELOCITY COMMAND TO SPEED CONTROL -----//
VELOCITY_OUT(controlLaw[k],convert.velM2C); //send command to Speed Control
//*****
samplingTime = samplingTime + info.sampling; //update new time

//----- display to screen -----//
pres_time = clock();
if( (pres_time - prev_time)/CLK_TCK ) >= 0.05 )
{
    clrscr();
    printf("\n k distance controlLaw radian ");
    for(j = 0; j<=19; j++)
    {
        gotoxy(1,5+j);
        printf("%3d %6.4f %6.4f %2.4f",
        ",display_k[j],display_errorPos[j],display_controlLaw[j],filAngle[k]; //errorAngle[k];
    }
    prev_time = clock();
    display_k[0] = display_k[1];
    display_k[1] = display_k[2];
    display_k[2] = display_k[3];
    display_k[3] = display_k[4];
    display_k[4] = display_k[5];
    display_k[5] = display_k[6];
    display_k[6] = display_k[7];
    display_k[7] = display_k[8];
    display_k[8] = display_k[9];
    display_k[9] = display_k[10];
    display_k[10] = display_k[11];
    display_k[11] = display_k[12];
    display_k[12] = display_k[13];
    display_k[13] = display_k[14];
    display_k[14] = display_k[15];
    display_k[15] = display_k[16];
    display_k[16] = display_k[17];
    display_k[17] = display_k[18];
    display_k[18] = display_k[19];
    display_k[19] = k;

    display_errorPos[0] = display_errorPos[1];
    display_errorPos[1] = display_errorPos[2];
    display_errorPos[2] = display_errorPos[3];
    display_errorPos[3] = display_errorPos[4];
    display_errorPos[4] = display_errorPos[5];
    display_errorPos[5] = display_errorPos[6];
    display_errorPos[6] = display_errorPos[7];
    display_errorPos[7] = display_errorPos[8];
    display_errorPos[8] = display_errorPos[9];
    display_errorPos[9] = display_errorPos[10];
    display_errorPos[10] = display_errorPos[11];
    display_errorPos[11] = display_errorPos[12];
    display_errorPos[12] = display_errorPos[13];
    display_errorPos[13] = display_errorPos[14];
    display_errorPos[14] = display_errorPos[15];
    display_errorPos[15] = display_errorPos[16];
    display_errorPos[16] = display_errorPos[17];
    display_errorPos[17] = display_errorPos[18];
    display_errorPos[18] = display_errorPos[19];
    display_errorPos[19] = errorPos[k];

    display_controlLaw[0] = display_controlLaw[1];
    display_controlLaw[1] = display_controlLaw[2];
    display_controlLaw[2] = display_controlLaw[3];
    display_controlLaw[3] = display_controlLaw[4];
    display_controlLaw[4] = display_controlLaw[5];
    display_controlLaw[5] = display_controlLaw[6];
    display_controlLaw[6] = display_controlLaw[7];
    display_controlLaw[7] = display_controlLaw[8];
    display_controlLaw[8] = display_controlLaw[9];
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        display_controllaw[9] = display_controllaw[10];
        display_controllaw[10] = display_controllaw[11];
        display_controllaw[11] = display_controllaw[12];
        display_controllaw[12] = display_controllaw[13];
        display_controllaw[13] = display_controllaw[14];
        display_controllaw[14] = display_controllaw[15];
        display_controllaw[15] = display_controllaw[16];
        display_controllaw[16] = display_controllaw[17];
        display_controllaw[17] = display_controllaw[18];
        display_controllaw[18] = display_controllaw[19];
        display_controllaw[19] = controlLaw[k];
    }
}
//-----
}while(samplingTime < maxTime); //END CONTROL LOOP
VELOCITY_OUT(0,convert.velM2C);
INIT_IM629();
clrscr();
end = clock(); //read clock when end routine
time = (long){(end-startProcess)/CLK_TCK*1000000};
gotoxy(X-6,Y);
printf("This Process use time = %10.0ld microsec ",time);
//-----
//----- WRITE POSITION M-FILE -----//
data_M = fopen(crane_M,"wt+");
fprintf(data_M,"\\nx = ["];
for(k = 0; k < MAXDATA; k++)
{
    fprintf(data_M,"\\n %3d",k);
}
fprintf(data_M,"\\n");
fprintf(data_M,"\\ny = ["];
for(k = 0; k < MAXDATA; k++)
{
    fprintf(data_M,"\\n %4.6f",posRef - errorPos[k] );
}
fprintf(data_M,"\\n");
fprintf(data_M,"\\nplot(x,y,'b');");
fprintf(data_M,"\\naxis([ 0.0, 250.0, 0.0, 1.2]);");
fprintf(data_M,"\\ntitle('Real position');");
fprintf(data_M,"\\nylabel('Distance [ m ]');");
fprintf(data_M,"\\nxlabel('Time [sec]');");
fclose(data_M); //close file
//-----WRITE ANGLE M-FILE -----//
data_ang_M = fopen(crane_ang_M,"wt+");
fprintf(data_ang_M,"\\nx = ["];
for(k = 0; k < MAXDATA; k++)
{
    fprintf(data_ang_M,"\\n %3d",k);
}
fprintf(data_ang_M,"\\n");
fprintf(data_ang_M,"\\ny = ["];
for(k = 0; k < MAXDATA; k++)
{
    fprintf(data_ang_M,"\\n %4.6f",realAngle[k]);
}
fprintf(data_ang_M,"\\n");
fprintf(data_ang_M,"\\nplot(x,y,'b');");
fprintf(data_ang_M,"\\naxis([ 0.0, 250.0, -0.2, 0.2]);");
fprintf(data_ang_M,"\\ntitle('Real angle');");
fprintf(data_ang_M,"\\nylabel('Angle [ radian ]');");
fprintf(data_ang_M,"\\nxlabel('time [ sec ]');");
fclose(data_ang_M);

gotoxy(15,22);
printf("Writing data to file is ready. \\n");

gotoxy(15,24);
printf("Press any key to quit program .");
getch();

} //End Of Program

float READ_DATA(void)
{
    struct angle_type
    {
        float angle;
        float volt;
        float ratio;
    }data;

    FILE *potentio file;
    potentio file = fopen(POTENTIO_FILE,"rb");
    fread(&data,sizeof(struct angle_type),1,potentio_file);
    fclose(potentio_file);

    clrscr();
    textcolor(7);
    printf("%4.4f degrees = %4.4f volt \\n(ratio = %4.4f / %4.4f = %4.4f)",data.angle, data.volt, data.angle, data.volt,
data.ratio );

    return data.ratio;
}

Conv_type OPEN_CONVERT_DATA(void)
{
    Conv_type convert;
    FILE *data_file;
    data_file = fopen(ConvertFile,"rb");
    fread(&convert, sizeof(struct Conv_type),1,data_file);
    fclose(data_file);

    return convert;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VERSION3.CPP

```

//----- STANDARD HEADER FILES -----//
#include<dos.h>
#include<conio.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include<bios.h>
#include <stdlib.h>
#include <process.h>

//----- OUR HEADER FILE -----//
#include "c:\b427\tc\include\adc.h"
#include "c:\b427\tc\include\c_test2.h"

//----- DEFINE CONSTANT MACRO -----//
#define PERIOD 0.1
#define LOW1 2500 //value to load to conuter2 8253
#define LOW2 2500
#define PI 3.14159265359
#define X 16
#define Y 5
#define MAXDATA 4

#define cranefile "c:\\data\\N42.dat" //response data file
#define crane_M "c:\\data\\m\\p.m" //m file for running in matlab
#define crane_ang_M "c:\\data\\m\\a.m"
#define Kf_source "c:\\data\\Kf.dat"
#define POTENTIO_FILE "c:\\data\\potentio.dat" //response data file
struct feedBackGain
{
    float angle;
    float difAngle;
    float pos;
    float difPos;
    float difDifPos;
};

typedef struct{
    float xMax; //Max Time Operation (sec)
    float yMax; //Set Point Position (cm)
    float sampling; //Sampling Period (sec)
    char *heading; //Heading of Graph (string)
    char *xLabel; //Label of X Axis (string)
    char *yLabel; //Label of Y Axis (string)
}graphInfo;
typedef struct{
    float secPpix;
    int pixPsampling;
}scaleX;
typedef struct{
    float cmPpix;
}scaleY;
typedef struct{
    scaleX x;
    scaleY y;
}axisScale;
struct Conv_type
{
    float posC2M;
    float velM2C;
};

int Stop = 1;

float READ_DATA(void);
Conv_type OPEN_CONVERT_DATA(void);
void main(void)
{
//----- DEFINE VARIABLE -----//
clrscr();
Conv_type convert;
//convert = OPEN_CONVERT_DATA();
convert.posC2M = 3.236455E-5;
convert.velM2C = 691176.5596;

printf("\nposC2M = %e", convert.posC2M);
printf("\nvelM2C = %e", convert.velM2C);
getch();

word adc;
int mode = PACER; //select PACER Mode to use as Sampling Time of Process
int base = 0x200; //select address H220 to use as BASE Address
int ch = 12; //select channel 12 to use
counterRegister counter;
float offset;
float volt2angle = 7.5;//READ DATA(); //value converse froms voltage to angle in degree
printf("\nvolt2angle = %4.4f",volt2angle);
getch();

graphInfo info;
axisScale scale;
float maxDistance = 1.0; //Set Maximum Value In Y-Axis of Graph
float samplingTime,maxTime=60.0; //Set Maximum Value In X-Axis of Graph

int k = 0;
long time = 0;
long startProcess = 0, startLoop = 0, end = 0;

float velocity = 0.0;
float loadPos;
double radian;

// Page 139
float F1,F2,F3;

FILE *datafile;
FILE *data_M;
FILE *data_ang_M;
FILE *Kf_file;

float volt[MAXDATA],realAngle[MAXDATA],realPos[MAXDATA];
float realDifAngle[MAXDATA],realDifPos[MAXDATA],realDifDifPos[MAXDATA];

float pos[MAXDATA], angle[MAXDATA];
float difPos[MAXDATA],difDifPos[MAXDATA],difAngle[MAXDATA];
float controlLaw[MAXDATA];

```

เอกสารนี้เป็นงานวิจัยสำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float f1, f2, f3, f4, f5;
float Ko[7][4], L[7][4];
float G[7][7], H[7][3];
float errorPos[MAXDATA], errorAngle[MAXDATA];

float estimAngle[MAXDATA], estimPos[MAXDATA], estimDifAngle[MAXDATA];
float estimDifPos[MAXDATA], estimDifDifPos[MAXDATA];
float ePos[MAXDATA], eAngle[MAXDATA];

float filAngle[MAXDATA];
float a1[5], b1[5], a2[5], b2[5], a3[5], b3[5], a4[5], b4[5], a5[5], b5[5], a6[5], b6[5];

float angleRef = 0.0; //Set Value of ANGLE REFERENCE

//----- INITIAL FEEDBACK GAIN -----//
//----- CALCULATE FORM Q = diag([100 1 10000 100 10]), R = diag([100 100]) see Example in chapter 5

Kf_file = fopen(Kf_source, "rb");
fread(&Kf, sizeof(struct feedBackGain), 1, Kf_file);
fclose(Kf_file);

//----- receive destination point -----//
float posRef; // = 1.0; //Set Value of POSITION REFERENCE
char y_n_e = 'n';
do
{
    //clrscr();
    //
    printf("\n Enter distance :          m.");
    gotoxy(18, 2);
    scanf("%f", &posRef);
    printf("\n distance = %4.4f (Yes/No/Exit)", posRef);
    cscanf("%c", &y_n_e);

    if( (y_n_e=='e') || (y_n_e=='E') )
    {
        exit(1);
    }
    clrscr();
}while( (y_n_e!='y') && (y_n_e!='Y') );

//getch();

float display_errorPos[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
int display_k[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
float display_controlLaw[20] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
int j = 0;
long count = 0;
float rad = 0;
float ang = 0;

//----- START PROGRAM -----//
//----- Set Info of graph -----//
info.xMax = maxTime;
info.yMax = maxDistance;
info.sampling = PERIOD;
info.heading = "Response of Load Position ";

// PAGE 140
info.xLabel = "Time (sec)";
info.yLabel = "Distance (cm)";
samplingTime = info.samplingTime;

//----- INITIAL DATA -----//
for(k=0; k<MAXDATA; k++)
{
    volt[k] = 0; //Voltage Reading from ADC On PCI-714 Card
    realAngle[k] = 0; //Real Angle Value Converts from Voltage
    realPos[k] = 0; //Real Position Value Reading from LM629
    realDifAngle[k] = 0; //Value Compute On Physics Relation
    realDifPos[k] = 0; //Value Compute On Physics Relation
    realDifDifPos[k] = 0; //Value Compute On Physics Relation
    angle[k] = 0; //Value After Measurement Update
    difAngle[k] = 0; //Value After Measurement Update
    pos[k] = 0; //Value After Measurement Update
    difPos[k] = 0; //Value After Measurement Update
    difDifPos[k] = 0; //Value After Measurement Update
    estimAngle[k] = 0; //Value Before Measurement Update
    estimPos[k] = 0; //Value Before Measurement Update
    estimDifAngle[k] = 0; //Value Before Measurement Update
    estimDifPos[k] = 0; //Value Before Measurement Update
    estimDifDifPos[k] = 0; //Value Before Measurement Update
    controlLaw[k] = 0; //Control Law
    errorPos[k] = 0;
    errorAngle[k] = 0;
    ePos[k] = 0; //realPos-estimatePos
    eAngle[k] = 0; //realAngle-estimateAngle
    filAngle[k] = 0; //Angle Vale That has Filtered from Band Pass Filter
}

// PAGE 141
//----- BAND-PASS FILTER FOR FILTER NOISE FORM ANGLE -----//
a1[1]=1.0000;a1[2]=0.0309;a1[3]= -0.9691;
b1[1]=0.9845;b1[2]=0.0000;b1[3]= -0.9845;

counter=COUNTER_REGISTER(PERIOD*1000000);
//----- INITIAL LM629 -----//
INIT_LM629(); //initial lm629
clrscr();

SET_PID_FILTER(50, 7, 28, 20000);
SET_ACCELERATION_OUT(0x10F); //0X100F);
//----- INITIAL 8253 COUNTER -----//
SET_COUNTER(base, counter); //Load sampling time and start counting

startProcess = clock(); //read start Process clock
startLoop = startProcess; //read start Loop clock
//----- START LOOP CONTROL -----//
clrscr();
gotoxy(1, 3);
printf(" k      errorPos[k]    controlLaw[k]    ang");
k = 0;
clock_t prev_time, pres_time;
prev_time = clock();

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//----- FIND OFFSET VOLTAGE -----//
TRIG(mode,base,ch,counter,k); //select channel and trig ADC to read data
adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time

offset = ((adc.high*256)+adc.low)-8192;
offset = (offset*10)/16384; //converse data to voltage value
char begin_step1 = 1;
char begin_step2 = 0;
char pos_check_stop = 0;
char ang_check_stop = 0;
char set_point_check_stop = 0;
char EXIT = 0;
//-----//

do
{
//----- READ ANGLE FROM ADC -----//
k++;
TRIG(mode,base,ch,counter,k); //select channel and trig ADC to read data
adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time
volt[1] = ((adc.high*256)+adc.low)-8192;
volt[1] = (volt[1]*10)/16384; //converse data to voltage value

volt[1] = volt[1] - offset; //automatic adjust offset
realAngle[1] = ((volt[1]*volt2angle)*2*PI)/360; //from voltage to "radian"

//----- FILTER ANGLE BEFORE COMPUTE -----//
if(begin_step2==1)
{
    filAngle[1] = b1[1]*realAngle[1] + b1[2]*realAngle[2] - a1[2]*filAngle[2];
    begin_step1 = 0;
}
if(begin_step1==1)
{
    filAngle[1] = b1[1]*realAngle[1];
    begin_step2 = 1;
}

if( (begin_step1 = 0) && (begin_step2 = 0) )
{
    filAngle[1]=b1[1]*realAngle[1]+b1[2]*realAngle[2] + b1[3]*realAngle[3]-a1[2]*filAngle[2]-
a1[3]*filAngle[3];
}
//----- READ POSITION FROM LM629 -----//
realPos[1] = READ_REAL_POS(convert.posC2M); //read real position from lm629

//----- STATE ESTIMATOR CALCULUS DifAngle DifPos DifDifPos FROM PHYSICS RELATION FOR LQR -----//
if(begin_step1 == 1)
{
    realDifPos[1] = 0;
    realDifDifPos[0] = 0;
    realDifAngle[0] = 0;
}
else
{
    realDifPos[1]=(realPos[1] - realPos[2])/info.sampling;
    realDifDifPos[1]=(realDifPos[1]-realDifPos[2])/info.sampling;
    realDifAngle[1] = (realAngle[1]-realAngle[2])/info.sampling;
}
errorAngle[1] = angleRef - filAngle[1];
errorPos[1] = posRef - realPos[1];

//----- COMPUTE CONTROL LAW (VELOCITY COMMAND) -----//
//----- ESTIMATE STATE FROM PHYSICS RELATION -----//
f2 = realDifAngle[1]*Kf.difAngle;
f3 = realDifPos[1]*Kf.difPos;
f4 = realDifDifPos[1]*Kf.difDifPos;
P1 = Kf.angle*errorAngle[1];
P2 = Kf.pos*errorPos[1];
P3 = f2+f3+f4;

controllaw[1] = P1 + P2 - P3 ;

//----- SEND VELOCITY COMMAND TO SPEED CONTROL -----//
VELOCITY_OUT(controllaw[1],convert.velM2C); //send command to Speed Control
//*****
samplingTime = samplingTime + info.sampling;//update new time

//----- display to screen -----//
pres time = clock();
if( ( pres_time - prev_time)/CLK_TCK >= 0.05 )
{
    clrscr();
    printf("\n k          errorPos          controllaw          errorRadian ");
    for(j = 0; j<=19; j++)
    {
        gotoxy(1,5+j);
        printf("%3d          %6.4f          %6.4f          %2.4f",
        ",display_k[j],display_errorPos[j],display_controllaw[j],filAngle[1];//errorAngle[k]);
    }
    prev time = clock();
    display_k[0] = display_k[1];
    display_k[1] = display_k[2];
    display_k[2] = display_k[3];
    display_k[3] = display_k[4];
    display_k[4] = display_k[5];
    display_k[5] = display_k[6];
    display_k[6] = display_k[7];
    display_k[7] = display_k[8];
    display_k[8] = display_k[9];
    display_k[9] = display_k[10];
    display_k[10] = display_k[11];
    display_k[11] = display_k[12];
    display_k[12] = display_k[13];
    display_k[13] = display_k[14];
    display_k[14] = display_k[15];
    display_k[15] = display_k[16];
    display_k[16] = display_k[17];
    display_k[17] = display_k[18];
    display_k[18] = display_k[19];
    display_k[19] = k;

    display_errorPos[0] = display_errorPos[1];
    display_errorPos[1] = display_errorPos[2];
    display_errorPos[2] = display_errorPos[3];
    display_errorPos[3] = display_errorPos[4];
    display_errorPos[4] = display_errorPos[5];
    display_errorPos[5] = display_errorPos[6];
    display_errorPos[6] = display_errorPos[7];
    display_errorPos[7] = display_errorPos[8];
    display_errorPos[8] = display_errorPos[9];
    display_errorPos[9] = display_errorPos[10];
    display_errorPos[10] = display_errorPos[11];
    display_errorPos[11] = display_errorPos[12];
    display_errorPos[12] = display_errorPos[13];
    display_errorPos[13] = display_errorPos[14];
    display_errorPos[14] = display_errorPos[15];
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

display_errorPos[15] = display_errorPos[16];
display_errorPos[16] = display_errorPos[17];
display_errorPos[17] = display_errorPos[18];
display_errorPos[18] = display_errorPos[19];
display_errorPos[19] = errorPos[1];

display_controlLaw[0] = display_controlLaw[1];
display_controlLaw[1] = display_controlLaw[2];
display_controlLaw[2] = display_controlLaw[3];
display_controlLaw[3] = display_controlLaw[4];
display_controlLaw[4] = display_controlLaw[5];
display_controlLaw[5] = display_controlLaw[6];
display_controlLaw[6] = display_controlLaw[7];
display_controlLaw[7] = display_controlLaw[8];
display_controlLaw[8] = display_controlLaw[9];
display_controlLaw[9] = display_controlLaw[10];
display_controlLaw[10] = display_controlLaw[11];
display_controlLaw[11] = display_controlLaw[12];
display_controlLaw[12] = display_controlLaw[13];
display_controlLaw[13] = display_controlLaw[14];
display_controlLaw[14] = display_controlLaw[15];
display_controlLaw[15] = display_controlLaw[16];
display_controlLaw[16] = display_controlLaw[17];
display_controlLaw[17] = display_controlLaw[18];
display_controlLaw[18] = display_controlLaw[19];
display_controlLaw[19] = controlLaw[1];
}
//----- ROTATE DATA -----//
realAngle[3] = realAngle[2];
filAngle[3] = filAngle[2];

realAngle[2] = realAngle[1];
filAngle[2] = filAngle[1];
realPos[2] = realPos[1];
realDifPos[2] = realDifPos[1];

//----- CHECK OUT -----//
if( bioskey(1) )
{
    EXIT = 1;
}
else
{
    EXIT = 0;
}

//-----//
}while(EXIT == 0); //END CONTROL LOOP
VELOCITY_OUT(0,convert.velM2C);
INIT_LM629();
clrscr();
end = clock(); //read clock when end routine
time = (long) (((end-startProcess)/CLK_TCK)*1000000);
gotoxy(X-6,Y);
printf("This Process use time = %10.0ld microsec ",time);

//-----//
printf("\n\nPress any key to quit program .");
getch();
//End Of Program

float READ_DATA(void)
{
    struct angle_type
    {
        float angle;
        float volt;
        float ratio;
    }data;

    FILE *potentio_file;
    potentio_file = fopen(POTENTIO_FILE,"rb");
    fread(&data, sizeof(struct angle_Type),1,potentio_file);
    fclose(potentio_file);

    clrscr();
    textcolor(7);
    printf("%4.4f degrees = %4.4f volt \n(ratio = %4.4f / %4.4f = %4.4f)",data.angle, data.volt, data.angle, data.volt,
data.ratio );

    return data.ratio;
}

Conv_type OPEN_CONVERT_DATA(void)
{
    Conv_type convert;
    FILE *data_file;
    data_file = fopen(ConvertFile,"rb");
    fread(&convert, sizeof(struct Conv_Type),1,data_file);
    fclose(data_file);

    return convert;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISTANCE.CPP

```

#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <process.h>
#include <stdlib.h>
#include <bios.h>
#include <math.h>

#include <c_test2.h>

#define ConvertFile "c:\\data\\Convert.dat"

struct Conv_type
{
    float posC2M;
    float velM2C;
}convert;

void SET_DISTANCE(void);
void CALCULATE_CONVERTS_PARAMETER(long count);
void WRITE_CONVERTS_PARAMETER_TO_FILE(Conv_type convert);

void main(void)
{
    char key;
    clrscr();
    printf("This program use for set distance.\n");
    printf("SET DISTANCE (y/n) : ");
    scanf("%c",&key);
    switch(key)
    {
        case 'Y' : SET_DISTANCE();
                    break;
        case 'y' : SET_DISTANCE();
                    break;
    }
}

//end main()

void SET_DISTANCE(void)
{
    long count = 0;
    clrscr();
    printf("\n Enter for set home position .");
    INIT_LM629();
    clrscr();
    printf("\n\n METER = ");
    do
    {
        count = READ_REAL_POS_COUNT();
        gotoxy(11,2);
        printf(" %ld",count);
        delay(20);
    }while(!bioskey(1));
    CALCULATE_CONVERTS_PARAMETER(count);
}

void CALCULATE_CONVERTS_PARAMETER(long count)
{
    Conv_type    convert;
    convert.posC2M = pow(count,-1);
    float a      = count*341.333333E-6;
    convert.velM2C = 65536*a;
    WRITE_CONVERTS_PARAMETER_TO_FILE(convert);
}

void WRITE_CONVERTS_PARAMETER_TO_FILE(Conv_type convert)
{
    FILE *Convert_Parameter_File;
    Convert_Parameter_File = fopen(ConvertFile,"wt+");
    fwrite(&convert, sizeof(struct Conv_type),1,Convert_Parameter_File);
    fclose(Convert_Parameter_File);
    printf("\n\n posC2M = %e          ",convert.posC2M);
    printf("\n\n velM2C = %f          ",convert.velM2C);
    printf("\n\n Write conversion parameter to file c:\\data\\CONVERT.DAT ");
    getch();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

};

//----- INITIAL DATA -----//
for(k=0; k<MAXDATA; k++)
{
    volt[k] = 0; //Voltage Reading from ADC On PCL-714 Card
    realAngle[k] = 0; //Real Angle Value Converse from Voltage
    realDifAngle[k] = 0; //Value Compute On Physics Relation
    angle[k] = 0; //Value After Measurement Update
    difAngle[k] = 0; //Value After Measurement Update
    estimAngle[k] = 0; //Value Before Measurement Update
    estimDifAngle[k] = 0; //Value Before Measurement Update
    errorAngle[k] = 0;
    eAngle[k] = 0; //realAngle-estimateAngle
    filAngle[k] = 0; //Angle Vale That has Filtered from Band Pass Filter
}

//----- BAND-PASS FILTER FOR FILTER NOISE FORM ANGLE -----//
al[1]=1.0000;al[2]=0.0309;al[3]= -0.9691;
bl[1]=0.9845;bl[2]=0.0000;bl[3]= -0.9845;

counter=COUNTER_REGISTER(PERIOD*1000000);
//----- INITIAL LM629 -----//
INIT_LM629(); //initial lm629
clrscr();
//----- INITIAL 8253 COUNTER -----//
SET_COUNTER(base,counter); //load sampling time and start counting

startProcess = clock(); //read start Process clock
startLoop = startProcess; //read start Loop clock
//----- START LOOP CONTROL -----//
clrscr();
gotoxy(1,3);
printf(" k          ang");
k = 0;
clock_t prev_time, pres_time;
prev_time = clock();
//----- FIND OFFSET VOLTAGE -----//
TRIG(mode,base,ch,counter,k); //select channel and trig ADC to read data
adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time

offset = ((adc.high*256+adc.low)-8192);
offset = (offset*10)/16384; //converse data to voltage value
char begin_step1 = 1;
char begin_step2 = 0;
//-----//

do
{
    //----- READ ANGLE FROM ADC -----//
    k++;
    TRIG(mode,base,ch,counter,k); //select channel and trig ADC to read data
    adc = READ_ADC(base); //wait for reading value from ADC when reach sampling time
    volt[1] = (((adc.high*256+adc.low)-8192);
    volt[1] = (volt[1]*10)/16384; //converse data to voltage value
    volt[2] = volt[1] - offset; //automatic adjust offset
    realAngle[1] = ((volt[1]*volt2angle)*2*PI)/360; //from voltage to "radian"
    //----- FILTER ANGLE BEFORE COMPUTE -----//
    if(begin_step2==1)
    {
        filAngle[1] = bl[1]*realAngle[1] + bl[2]*realAngle[2] - al[2]*filAngle[2];
        begin_step1 = 0;
    }
    if(begin_step1==1)
    {
        filAngle[1] = bl[1]*realAngle[1];
        begin_step2 = 1;
    }
    if( (begin_step1 = 0) && (begin_step2 = 0) )
    {
        filAngle[1]=bl[1]*realAngle[1]+bl[2]*realAngle[2] + bl[3]*realAngle[3]-al[2]*filAngle[2]-
al[3]*filAngle[3];
    }
    //----- STATE ESTIMATOR CALCULUS DifAngle DifPos DifDifPos FROM PHYSICS RELATION FOR LQR -----//
    if(begin_step1 == 1)
    {
        realDifAngle[0] = 0;
    }
    else
    {
        realDifAngle[1] = (realAngle[1]-realAngle[2])/info.sampling;
    }
    errorAngle[1] = angleRef - filAngle[1];
    samplingTime = samplingTime + info.sampling;//update new time

    //-----display to screen -----//
    pres_time = clock();
    if( ( pres_time - prev_time)/CLK_TCK ) >= 0.05 )
    {
        clrscr();
        gotoxy(0,0);
        printf("\n\n k          radian          degree          volt*");
        for(j = 0; j<19; j++)
        {
            gotoxy(1,5+j);
            printf("%3d          %6.4f          %6.4f          %4.4f
",display_k[j],display_filangle[j],display_degree[j], display_v[1]);//errorAngle[k]);
        }
        prev_time = clock();

        display_k[ 0] = display_k[1];
        display_k[ 1] = display_k[2];
        display_k[ 2] = display_k[3];
        display_k[ 3] = display_k[4];
        display_k[ 4] = display_k[5];
        display_k[ 5] = display_k[6];
        display_k[ 6] = display_k[7];
        display_k[ 7] = display_k[8];
        display_k[ 8] = display_k[9];
        display_k[ 9] = display_k[10];
        display_k[10] = display_k[11];
        display_k[11] = display_k[12];
        display_k[12] = display_k[13];
        display_k[13] = display_k[14];
        display_k[14] = display_k[15];
        display_k[15] = display_k[16];
        display_k[16] = display_k[17];
        display_k[17] = display_k[18];
        display_k[18] = display_k[19];
        display_k[19] = k;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

display_filangle[ 0] = display_filangle[1];
display_filangle[ 1] = display_filangle[2];
display_filangle[ 2] = display_filangle[3];
display_filangle[ 3] = display_filangle[4];
display_filangle[ 4] = display_filangle[5];
display_filangle[ 5] = display_filangle[6];
display_filangle[ 6] = display_filangle[7];
display_filangle[ 7] = display_filangle[8];
display_filangle[ 8] = display_filangle[9];
display_filangle[ 9] = display_filangle[10];
display_filangle[10] = display_filangle[11];
display_filangle[11] = display_filangle[12];
display_filangle[12] = display_filangle[13];
display_filangle[13] = display_filangle[14];
display_filangle[14] = display_filangle[15];
display_filangle[15] = display_filangle[16];
display_filangle[16] = display_filangle[17];
display_filangle[17] = display_filangle[18];
display_filangle[18] = display_filangle[19];
display_filangle[19] = filAngle[1];

display_degree[ 0] = display_degree[1];
display_degree[ 1] = display_degree[2];
display_degree[ 2] = display_degree[3];
display_degree[ 3] = display_degree[4];
display_degree[ 4] = display_degree[5];
display_degree[ 5] = display_degree[6];
display_degree[ 6] = display_degree[7];
display_degree[ 7] = display_degree[8];
display_degree[ 8] = display_degree[9];
display_degree[ 9] = display_degree[10];
display_degree[10] = display_degree[11];
display_degree[11] = display_degree[12];
display_degree[12] = display_degree[13];
display_degree[13] = display_degree[14];
display_degree[14] = display_degree[15];
display_degree[15] = display_degree[16];
display_degree[16] = display_degree[17];
display_degree[17] = display_degree[18];
display_degree[18] = display_degree[19];
display_degree[19] = (filAngle[1]*180)/PI ;

display_v[ 0] = display_v[1];
display_v[ 1] = display_v[2];
display_v[ 2] = display_v[3];
display_v[ 3] = display_v[4];
display_v[ 4] = display_v[5];
display_v[ 5] = display_v[6];
display_v[ 6] = display_v[7];
display_v[ 7] = display_v[8];
display_v[ 8] = display_v[9];
display_v[ 9] = display_v[10];
display_v[10] = display_v[11];
display_v[11] = display_v[12];
display_v[12] = display_v[13];
display_v[13] = display_v[14];
display_v[14] = display_v[15];
display_v[15] = display_v[16];
display_v[16] = display_v[17];
display_v[17] = display_v[18];
display_v[18] = display_v[19];
display_v[19] = volt[1];
}

//----- ROTATE DATA -----//
realAngle[3] = realAngle[2];
filAngle[3] = filAngle[2];

realAngle[2] = realAngle[1];
filAngle[2] = filAngle[1];

//-----//

}while(!bioskey(1)); //END CONTROL LOOP

end = clock(); //read clock when end routine
time = (long)((end-startProcess)/CLK_TCK)*1000000;
printf("\nThis Process use time = %10.0ld microsec ",time);

//-----//

printf("\n\nPress any key to quit program.");
getch();
//End Of Program

void CHANGE_RATIO(void)
{
float angle,volt,ratio;
clrscr();
gotoxy(5,8);
printf(" Enter ANGLE(degree) and VOLT");
gotoxy(5,10);
printf(" ANGLE : degree = VOLT");
gotoxy(13,10);
scanf("%f",&angle);
gotoxy(36,10);
scanf("%f",&volt);
ratio = angle/volt;

gotoxy(5,12);
printf(" RATIO = %f",ratio);

textattr(0x84); // 1000 0100
gotoxy(5,14);
cprintf(" Are you sure? (Yes/No) : ");
textcolor(7);
char key = getch();
clrscr();
switch(key)
{
case 'Y' : WRITE_DATA(angle,volt,ratio);
break;
case 'y' : WRITE_DATA(angle,volt,ratio);
break;
default : gotoxy(5,14);
printf("OK no any setting. ");
getch();
break;
};
};

void WRITE_DATA(float angle, float volt, float ratio)
{
struct angle_type
{
float angle;
float volt;
float ratio;
}
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}data;
data.angle = angle;
data.volt = volt;
data.ratio = ratio;

FILE *potentio_file;

potentio_file = fopen(POTENTIO_FILE,"wb+");
fwrite(&data,sizeof(struct angle_type),1,potentio_file);
fclose(potentio_file);
}

float READ_DATA(void)
{
    struct angle_type
    {
        float angle;
        float volt;
        float ratio;
    }data;

    FILE *potentio_file;
    potentio_file = fopen(POTENTIO_FILE,"rb");
    fread(&data,sizeof(struct angle_type),1,potentio_file);
    fclose(potentio_file);

    clrscr();
    textcolor(7);
    printf("%4.4f degrees = %4.4f volt \n(ratio = %4.4f / %4.4f = %4.4f)",data.angle, data.volt, data.angle, data.volt,
data.ratio );

    return data.ratio;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SETLM629.CPP

```

#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <bios.h>
#include <process.h>
#include <stdlib.h>

#define pid_head "c:\\b427\\tc\\include\\629pid.h"

void WRITE_TO_FILE(long P,long I,long D,long IL ,long MAX_ACC);
void main(Void)
{
    long P,I,D,IL,MAX_ACC;
    char ch;

LOOP:
    clrscr();
    printf("\nEnter PID in DEC. ");
    gotoxy(5,5);
    textcolor(7);
    cprintf("P = ");
    gotoxy(5,7);
    cprintf("I = ");
    gotoxy(5,9);
    cprintf("D = ");
    gotoxy(5,11);
    cprintf("IL = ");
    gotoxy(5,13);
    cprintf("MAX_ACC = ");

    gotoxy(10,5);
    scanf("%ld",&P);
    gotoxy(10,7);
    scanf("%ld",&I);
    gotoxy(10,9);
    scanf("%ld",&D);
    gotoxy(10,11);
    scanf("%ld",&IL);
    gotoxy(15,13);
    scanf("%ld",&MAX_ACC);

    printf("\n    Set PID with...");
    printf("\n    P = %ld",P);
    printf("\n    I = %ld",I);
    printf("\n    D = %ld",D);
    printf("\n    IL = %ld",IL);
    printf("\n    MAX_ACC = %ld",MAX_ACC);

    printf("\n    Yes/No (y/n) : ");

    ch = bioskey(0);
    switch(ch)
    {
        case 'y': WRITE_TO_FILE(P,I,D,IL,MAX_ACC);
                  break;
        case 'Y': WRITE_TO_FILE(P,I,D,IL,MAX_ACC);
                  break;
        case 'n': exit(0);
                  break;
        case 'N': exit(0);
                  break;
    };
    goto LOOP;
}

void WRITE_TO_FILE(long P, long I, long D, long IL, long MAX_ACC)
{
    FILE *PID_File;
    PID_File = fopen(pid_head,"wt+");
    fprintf(PID_File,"#define P %ld\n#define I %ld\n#define D %ld\n#define IL %ld\n#define MAX_ACC %ld ",P,I,D,IL
,MAX_ACC);
    fclose(PID_File);
    exit(0);
}

```

STEPTEST.CPP

```

#include<dos.h>
#include<conio.h>
#include<stdio.h>
#include<time.h>
#include<math.h>
#include<bios.h>
#include<stdlib.h>
#include<process.h>

#define MAXDATA 18
#define PERIOD 0.1
#define crane_M "c:\\data\\m\\c_goto2.m" //m file for running in matlab

//----- OUR HEADER FILE -----//

#include<c_test2.h>

void main(void)
{
    Conv_type convert;
    convert = OPEN_CONVERT_DATA();
    float m_distance = 0.0;
    long c_distance;
    float velo;
    long c_position;
    float m_position = 0.0;

    clock_t start_T, end_T;
    float m_p1;
    float m_p2;
    float Ts = PERIOD;

    float a,b,c,d,e;

    float m_velo[MAXDATA];
    float step_time[MAXDATA];
    float total_time = 0.0;

    FILE *data_M;

    clrscr();
    //----- RECEIVE DISTANCE IN METER -----//
    INIT_LM629();
    clrscr();
    gotoxy(2,4);
    printf("\nEnter velocity : (m/sec)");
    gotoxy(18,5);
    scanf("%f",&velo);
    //printf("m_distance = %4.6f",m_distance);
    //getch();
    //----- INITIALIZATION -----//
    INIT_LM629();
    //mao++
    SET_PID_FILTER(5,5,285,240);
    SET_ACCELERATION_OUT(0x500);
    step_time[0] = 0.0;
    //----- CONVERT DISTANCE FROM M 2 C -----//
    c_distance = m_distance*convert.posC2M;

    VELOCITY_OUT(velo*1.1); //start motion.
    clrscr();
    //----- DETECT POSITION FOR STOP -----//
    start_T = clock();

    c_position = READ_REAL_POS_COUNT();
    m_position = c_position*convert.posC2M;
    m_p1 = m_position;

    //----- LOOP -----//
    for(int k=0; k<MAXDATA; k++)
    {
        do
        {
            end_T = clock();
        }while( (end_T - start_T)/CLK_TCK < Ts );
        Ts = (end_T - start_T)/CLK_TCK;
        m_p2 = (float)READ_REAL_POS_COUNT()*convert.posC2M;
        m_velo[k] = (m_p2 - m_p1)/Ts;
        total_time = total_time + Ts;
        step_time[k] = total_time;
        Ts = PERIOD;
        c_position = READ_REAL_POS_COUNT();
        m_position = c_position*convert.posC2M;
        m_p1 = m_position;

        start_T = clock();
        printf("\n step_time[%3d] = %3.3f m_position = %4.4f m_velo = %4.4f ",k,step_time[k],m_position,m_velo[k]);
    }
    //end for
    //STOP();
    TURN_OFF_MOTOR();

    //----- WRITE M-FILE -----//
    data_M = fopen(crane_M,"wt+");

    fprintf(data_M,"nx = [");
    for(k = 0; k < MAXDATA; k++)
    {
        fprintf(data_M,"\\n %4.6f",step_time[k]);
    }
    fprintf(data_M,"\\n");
    fprintf(data_M,"ny = [");
    for(k = 0; k < MAXDATA; k++)
    {
        fprintf(data_M,"\\n %4.6f",m_velo[k]);
    }
    fprintf(data_M,"\\n");
    fprintf(data_M,"\\nplot(x,y,'w');");
    fprintf(data_M,"\\nhold;");
    fprintf(data_M,"\\nz = [");
    for(k = 0; k < MAXDATA; k++)
    {
        fprintf(data_M,"\\n %4.6f",1.0);
    }
    fprintf(data_M,"\\n");
    fprintf(data_M,"\\nplot(x,z,'g');");
    fprintf(data_M,"\\ngrid;");
}

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรเผยแพร่ หรือทำซ้ำโดยไม่ได้รับอนุญาต การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(data_M); //close file

printf("\n\nWriting data to file is ready. \n");
printf("\nPress any key to quit program .");
printf("\n total time = %4.4f",total_time);
getch();
)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C TEST2.H

```

#ifdef __c_TEST_H__
#define __c_TEST_H__

// LM629 ADDRESS PORT
#define ADDR_COM 0x300 //address for write command and read status flag
#define ADDR_DATA 0x301 //address for read and write data

#define veloCorrectFact 0.9 //correct factor for convert to //actual velocity

#define kpConvert 256 //mac-> 0.307 //kp=(256*40/33287)*Kp
#define kiConvert 2.8 //mac-> 0.027 //ki=(6553
#define kdConvert 75000 //mac->9901.25
#define kp 3.48 //mac-> 28.5 //old kp=28.5 //proportion gain in metre
#define ki 4.43 //mac-> 345.7 //integration gain in metre
#define kd 0.68 //mac-> 3.6 //old kd=3.6 //derivative gain in metre
#define il 50000 //integration limit
#define filterCommand 0x001F //kp,ki,kd well be loaded

#define InitMaxAccCommand 0x1820 //trajectory control word a 18 hex
// = 0001 1000 0010 0000 //HB program forward direction //velocity mode, acceleration //will be loaded

#define maxVelocity 1000.0 //in m/s
#define maxAcceleration 32760 //in m/s^2
#define posCommand 0x002A //position mode ACC VELO position will be loaded and it 's absolute
#define forwVeloCommand 0x1808 //forward direction velocity will be loaded
#define rewVeloCommand 0x0808 //reward direction velocity will be loaded
#define stopMotorCommand 0x0200 //stop motor abruptly trajectory
#define turn_off_motor 0x0100 // turn off Motor (Output Zero Drive) command
#define resetBreakCommand 0x0000 //reset all interrupt

//command code for lm629 define
#define RESET 0x00 //reset the lm629 data n
#define DFH 0x02 //define home data n
#define SIP 0x03 //set index position data n
#define LPEI 0x1B //load position error for interrupt data 2
#define MSKI 0x1C //mask interrupts data 2
#define RSTI 0x1D //reset interrupts data 2
#define LFIL 0x1E //funtion load parameter of filter to lm629
#define UDF 0x04 //update filter parameter
#define LTRJ 0x1F //load trajectory
#define STT 0x01 //start trajectory
#define SBPA 0x20 //set break point absolute
#define RDSIGS 0x0C //read signal register
#define RDDP 0x08 //read desired position
#define RDRP 0x0A //read real position
#define RDDV 0x07 //read desired velocity
#define RDRV 0x0B //read real velocity

#define ConvertFile "c:\\data\\Convert.dat"

/* PROTOTYPE */
/--->Big function
void INIT_LM629(void); //initial lm629
void VELOCITY_OUT(float,float); //send velocity command to lm629
float READ_REAL_POS(float); //read real position

/--->Sub function
void WAITBUSY(void); //wait if lm629 busies
void WRITE_COM(unsigned char); //write command to lm629
void WRITE_DATA_BYTE(unsigned char); //write data 1 byte to lm629
void WRITE_DATA_2BYTES(int); //write data 2 bytes to lm629
void WRITE_DATA_4BYTES(unsigned long); //write data 4 bytes to lm629
void WRITE_DATA_WORD(int); //write word data ONLY FILTER DATA
void WRITE_DATA_2WORDS(unsigned long); //write 2 words data ONLY TRAJECTORY DATA
unsigned char READ_DATA_BYTE(void); //read data from lm629
int READ_DATA_2BYTES(void); //read 2 BYTES data from lm629
long READ_DATA_4BYTES(void); //read 4 BYTES data from lm629
/---> NEW FUNCTION
void EXIT_PROGRAM(void);
void STOP(void);
void TURN_OFF_MOTOR(void);
void SET_FID_FILTER(unsigned int, unsigned int, unsigned int, unsigned int);
void SET_ACCELERATION_OUT(unsigned long);
long READ_REAL_POS_COUNT(void);

/-----*/ FUNCTION */-----//
long READ_REAL_POS_COUNT(void) //read read position
{
    WRITE_COM(RDRP); //COMMAND CODE
    return READ_DATA_4BYTES(); //read real position
}

void WAITBUSY{//1
{
    while(0x01 & inp(ADDR_COM)); //wait util busy flag is zero
}

void WRITE_COM(unsigned char comm){//2
{
    WAITBUSY();
    outp(ADDR_COM, comm);
}

void WRITE_DATA_BYTE(unsigned char data){//3
{
    WAITBUSY();
    outp(ADDR_DATA, data);
}

void WRITE_DATA_2BYTES(int data){//4
{
    char *temp;
    temp=(char *)&data;
    WAITBUSY();
    outp(ADDR_DATA, *(temp+1));
    WAITBUSY();
    outp(ADDR_DATA, *(temp));
}

void WRITE_DATA_4BYTES(unsigned long data){//5
{
    char *temp;
    temp=(char *)&data;
    WAITBUSY();
    outp(ADDR_DATA, *(temp+3));
    WAITBUSY();
    outp(ADDR_DATA, *(temp+2));
    WAITBUSY();
    outp(ADDR_DATA, *(temp+1));
    WAITBUSY();
    outp(ADDR_DATA, *(temp));
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outp(ADDR_DATA,*temp);
}

void WRITE_DATA_WORD(int data)//6
{
    char *temp;
    temp=(char *)&data;
    WAITBUSY();
    outp(ADDR_DATA,* (temp+1));
    outp(ADDR_DATA,* (temp));
}

void WRITE_DATA_2WORDS(unsigned long data)//7
{
    char *temp;
    temp=(char *)&data;
    WAITBUSY();
    outp(ADDR_DATA,* (temp+3));
    outp(ADDR_DATA,* (temp+2));
    WAITBUSY();
    outp(ADDR_DATA,* (temp+1));
    outp(ADDR_DATA,*temp);
}

unsigned char READ_DATA_BYTE(void)//8
{
    WAITBUSY();
    return inp(ADDR_DATA);
}

int READ_DATA_2BYTES(void)//9
{
    char temp1[1];
    int *temp2;
    WAITBUSY();
    temp1[1]=inp(ADDR_DATA);
    WAITBUSY();
    temp1[0]=inp(ADDR_DATA);
    temp2=(int *)temp1;
    return *temp2;
}

long READ_DATA_4BYTES(void)// 10
{
    char temp1[3];
    long *temp2;
    WAITBUSY();
    temp1[3]=inp(ADDR_DATA);
    WAITBUSY();
    temp1[2]=inp(ADDR_DATA);
    WAITBUSY();
    temp1[1]=inp(ADDR_DATA);
    WAITBUSY();
    temp1[0]=inp(ADDR_DATA);
    temp2=(long *)temp1;
    return *temp2;
}

void INIT_LM629(void)//12
{
    int status;

    printf("\n>>CONTROL CRANE PROGRAM NOW ACTIVE ");
    WRITE_COM(RESET); //reset command code

    printf("\n>>INITIAL LM629 NO.1 NOW PASS ");
    WRITE_COM(DFH); //define home command
    WRITE_COM(SIP); //set index position command
    printf("\n>>INITIAL LM629 NO.2 NOW PASS ");
}

void VELOCITY_OUT(float velocity, float velM2C)
{
    long velo;

    /* if (velocity>maxVelocity)
        velocity=maxVelocity;

    if (velocity<-maxVelocity)
        velocity=-maxVelocity;
    */

    velo=(long) (velocity*veloCorrectFact*velM2C);
    if (velo>=0) //forward direction
    {
        WRITE_COM(LTRJ); //load trajectory command code
        WRITE_DATA_WORD(forwVeloCommand);
        WRITE_DATA_2WORDS((unsigned long)velo);
        WRITE_COM(STT);
    }
    if (velo<0) //reward direction
    {
        velo=velo*(-1);
        WRITE_COM(LTRJ); //load trajectory command code
        WRITE_DATA_WORD(rewVeloCommand);
        WRITE_DATA_2WORDS((unsigned long)velo);
        WRITE_COM(STT);
    }
}

float READ_REAL_POS(float posC2M) //read read position
{
    WRITE_COM(RDRP); //COMMAND CODE
    return (float)READ_DATA_4BYTES()*posC2M; //read real position
}
//-----
void EXIT_PROGRAM(void)
{
    STOP();
    INIT_LM629();
    exit(1);
}

void STOP(void)
{
    WRITE_COM(LTRJ);
    WRITE_DATA_WORD(stopMotorCommand); // 0000 0111 0000 0000
    WRITE_COM(STT);
    WRITE_COM(RST);
    WRITE_DATA_2BYTES(resetBreakCommand);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE_COM(LTRJ);
WRITE_DATA_WORD(turn_off_motor); // 0000 0111 0000 0000
WRITE_COM(STT);
WRITE_COM(RSTI);
WRITE_DATA_2BYTES(resetBreakCommand);
}

void SET_PID_FILTER(unsigned int Kp, unsigned int Ki, unsigned int Kd, unsigned int Il)
{
    WRITE_COM(LFIL); //send command code.
    WRITE_DATA_WORD(filterCommand); //Kp, Ki, Kd will be loaded.
    WRITE_DATA_WORD(Kp); //Kp parameter
    WRITE_DATA_WORD(Ki); //Ki parameter
    WRITE_DATA_WORD(Kd); //Kd parameter
    WRITE_DATA_WORD(Il); //Il parameter
    WRITE_COM(UDF); //update filter parameter
}

void SET_ACCELERATION_OUT(unsigned long Acc)
{
    WRITE_COM(LTRJ); //COMMAND CODE
    WRITE_DATA_WORD(InitMaxAccCommand); //load trajectory control ward //velocity mode acceleration //will be loaded

    WRITE_DATA_2WORDS(Acc);
    WRITE_COM(STT);
}

//-----//

#endif

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC.H

```

//-----> 14 bits Analog to Digital Conversion (for PCL 714 card)
#if !defined( _ADC_H )
#define _ADC_H

#include <conio.h>
#include <math.h>
#include <time.h>

//Define Constant Value

#define BASE 0X200
#define CH 12 //select ADC channel
#define TRIGGER 1
#define PACER 2
typedef struct
{
    char high;
    char low;
}word;
typedef struct
{
    //word c0 //counter 0 not use
    word c1; //counter 1
    word c2; //counter 2
}counterRegister;

//-----Prototype of Function-----//
counterRegister COUNTER_REGISTER(unsigned long period);
void SET_COUNTER(int base,counterRegister counter);
void TRIG(int mode, int base, int channel, counterRegister period, unsigned int k);
word READ_ADC(int base);
counterRegister READ_COUNTER(int base);

//----- Function -----//
counterRegister COUNTER_REGISTER(unsigned long period)
{
    unsigned int high = 0;
    unsigned int low = 2;
    counterRegister counter;

    low = (int)sqrt(period);
    high= (period*2)/low;
    counter.c1.high = high/256;
    counter.c1.low = high%256;
    counter.c2.high = low/256;
    counter.c2.low = low%256;

    return counter;
}
void SET_COUNTER(int base, counterRegister counter)
{
    outp(base+3, 0x74);
    outp(base+1,counter.c1.low);
    outp(base+1,counter.c1.high);
    outp(base+3,0x84);
    outp(base+2,counter.c2.low);
    outp(base+2,counter.c2.high);
}
counterRegister READ_COUNTER(int base)
{
    counterRegister count;
    //Read data from counter
    outp(base+3,64);
    count.c1.low = inp(base+1);
    count.c1.high = inp(base+1);
    outp(base+3,192);
    count.c2.low = inp(base+2);
    count.c2.high = inp(base+2);
    //time=(count.c1.high*256+count.c1.low)*(count.c2.high*256+count.c2.low);
    //write counter data to screen
    gotoxy(20,15);
    printf("COUNTER1 = %d ",count.c1.low); //test
    gotoxy(20,17);
    printf("COUNTER1 = %d ",count.c1.high); //test
    gotoxy(20,19);
    printf("COUNTER2 = %d ",count.c2.low); //test
    gotoxy(20,21);
    printf("COUNTER2 = %d ",count.c2.high); //test
    return count;
}
word READ_ADC(int base)
{
    word data;
    do
    {
        //wait for complete conversion
        data.high = inp(base+5);
    }while(data.high >= 64);
    data.low = inp(base+4);

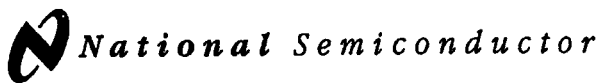
    //----->>>MRO
    // printf("\n adc = %X %X ",data.high, data.low);
    return data;
}
void TRIG(int mode, int base, int channel, counterRegister period, unsigned int k)
{
    outp(base+10,channel);
    outp(base+11,mode);
}

#endif

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



February 1996

DS26LS31C/DS26LS31M Quad High Speed Differential Line Driver

General Description

The DS26LS31 is a quad differential line driver designed for digital data transmission over balanced lines. The DS26LS31 meets all the requirements of EIA Standard RS-422 and Federal Standard 1020. It is designed to provide unipolar differential drive to twisted-pair or parallel-wire transmission lines.

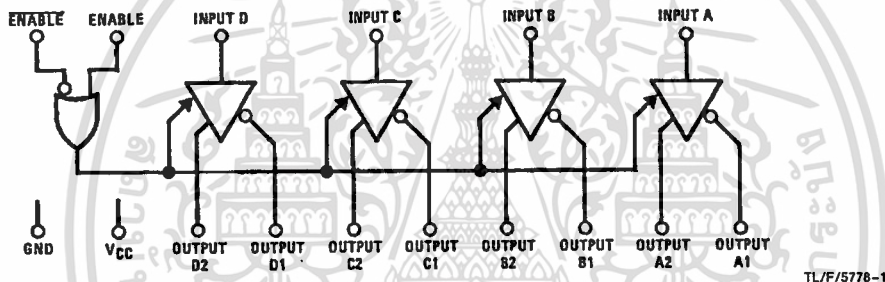
The circuit provides an enable and disable function common to all four drivers. The DS26LS31 features TRI-STATE® outputs and logically ANDed complementary outputs. The inputs are all LS compatible and are all one unit load.

Features

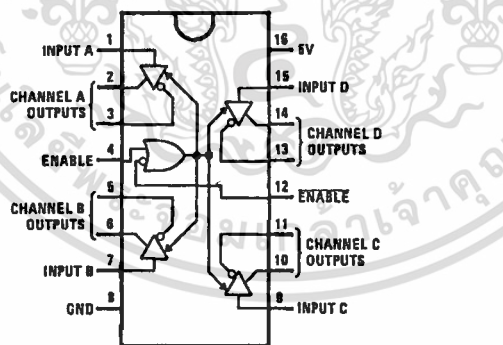
- Output skew—2.0 ns typical
- Input to output delay—10 ns typical
- Operation from single 5V supply
- Outputs won't load line when $V_{CC} = 0V$
- Four line drivers in one package for maximum package density
- Output short-circuit protection
- Complementary outputs
- Meets the requirements of EIA Standard RS-422
- Pin compatible with AM26LS31
- Available in military and commercial temperature range

DS26LS31C/DS26LS31M Quad High Speed Differential Line Driver

Logic and Connection Diagrams



Dual-In-Line Package



Top View

Order Number DS26LS31CJ, DS26LS31CM,
DS26LS31CN or DS26LS31MJ
See NS Package Number J16A, M16A or N16A

For Complete Military 883 Specifications, see RETS Data Sheet.
Order Number DS26LS31MJ/883 or DS26LS31MW/883
See NS Package J16A or W16A

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Output Voltage	5.5V
Output Voltage (Power OFF)	-0.25 to 6V
Maximum Power Dissipation* at 25°C	
Cavity Package	1509 mW
Molded DIP Package	1476 mW
SO Package	1051 mW

*Derate cavity package 10.1 mW/°C above 25°C; derate molded DIP package 11.9 mW/°C above 25°C; derate SO package 8.41 mW/°C above 25°C.

Operating Conditions

	Min	Max	Units
Supply Voltage, V_{CC}			
DS26LS31M	4.5	5.5	V
DS26LS31	4.75	5.25	V
Temperature, T_A			
DS26LS31M	-55	+125	°C
DS26LS31	0	+70	°C

Electrical Characteristics (Notes 2, 3 and 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{OH}	Output High Voltage	$I_{OH} = -20$ mA	2.5			V
V_{OL}	Output Low Voltage	$I_{OL} = 20$ mA			0.5	V
V_{IH}	Input High Voltage		2.0			V
V_{IL}	Input Low Voltage				0.8	V
I_{IL}	Input Low Current	$V_{IN} = 0.4$ V		-40	-200	• A
I_{IH}	Input High Current	$V_{IN} = 2.7$ V			20	• A
I_I	Input Reverse Current	$V_{IN} = 7$ V			0.1	mA
I_O	TRI-STATE Output Current	$V_O = 2.5$ V			20	• A
		$V_O = 0.5$ V			-20	• A
V_{CL}	Input Clamp Voltage	$I_{IN} = -18$ mA			-1.5	V
I_{SC}	Output Short-Circuit Current		-30		-150	mA
I_{CC}	Power Supply Current	All Outputs Disabled or Active		35	60	mA

Switching Characteristics $V_{CC} = 5$ V, $T_A = 25^\circ$ C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{PLH}	Input to Output	$C_L = 30$ pF		10	15	ns
t_{PHL}	Input to Output	$C_L = 30$ pF		10	15	ns
Skew	Output to Output	$C_L = 30$ pF		2.0	6.0	ns
t_{LZ}	Enable to Output	$C_L = 10$ pF, S2 Open		15	35	ns
t_{HZ}	Enable to Output	$C_L = 10$ pF, S1 Open		15	25	ns
t_{ZL}	Enable to Output	$C_L = 30$ pF, S2 Open		20	30	ns
t_{ZH}	Enable to Output	$C_L = 30$ pF, S1 Open		20	30	ns

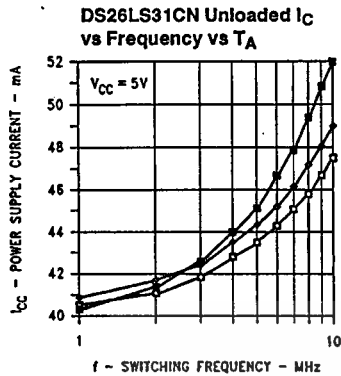
Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the devices should be operated at these limits. The tables of "Electrical Characteristics" provide conditions for actual device operation.

Note 2: Unless otherwise specified min/max limits apply across the -55°C to +125°C temperature range for the DS726LS31M and across the 0°C to +70°C range for the DS26LS31. All typicals are given for $V_{CC} = 5$ V and $T_A = 25^\circ$ C.

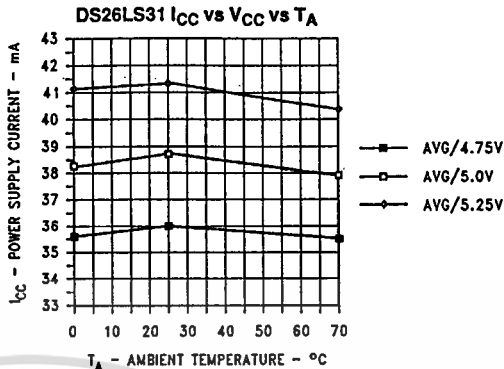
Note 3: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to ground unless otherwise specified.

Note 4: Only one output at a time should be shorted.

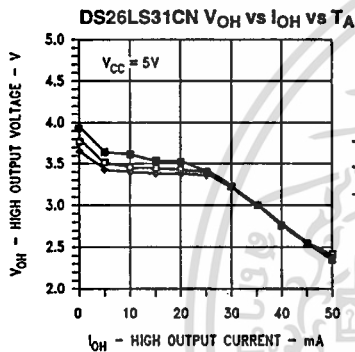
Typical Performance Characteristics



TL/F/5778-7



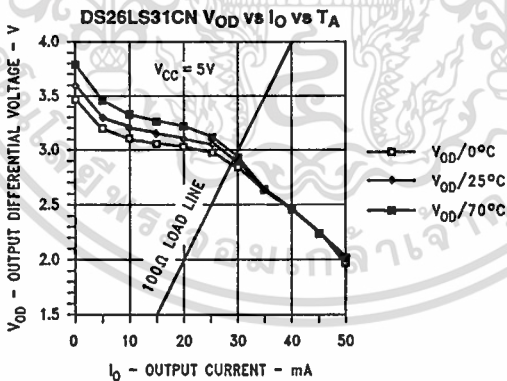
TL/F/5778-8



TL/F/5778-9



TL/F/5778-10



TL/F/5778-11

AC Test Circuit and Switching Time Waveforms

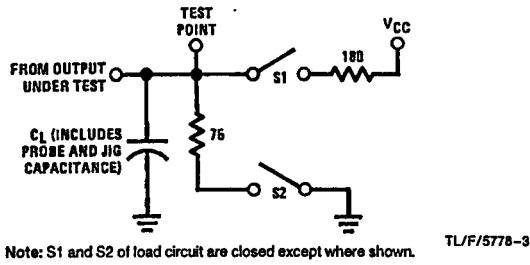


FIGURE 1. AC Test Circuit

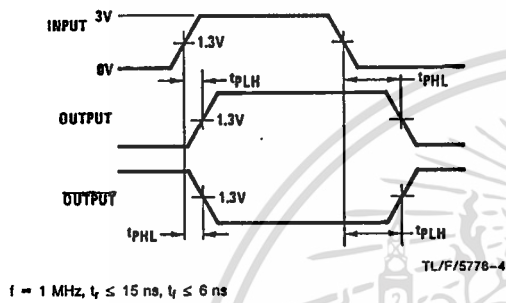


FIGURE 2. Propagation Delays

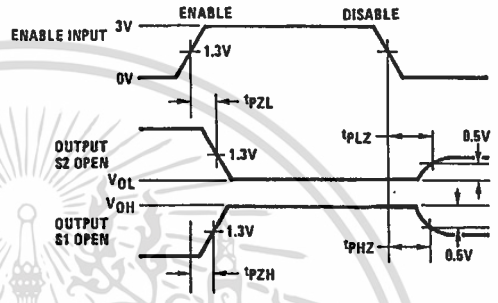


FIGURE 3. Enable and Disable Times

Typical Applications

Two-Wire Balanced System, RS-422



*RT is optional although highly recommended to reduce reflection.

TL/F/5778-8

DS26LS32C/DS26LS32M/DS26LS32AC/DS26LS33C/DS26LS33M/DS26LS33AC Quad Differential Line Receivers

General Description

The DS26LS32 and DS26LS32A are quad differential line receivers designed to meet the RS-422, RS-423 and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission.

The DS26LS32 and DS26LS32A have an input sensitivity of 200 mV over the input voltage range of $\pm 7V$ and the DS26LS33 and DS26LS33A have an input sensitivity of 500 mV over the input voltage range of $\pm 15V$.

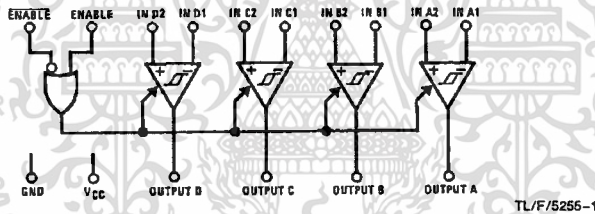
Both the DS26LS32A and DS26LS33A differ in function from the popular DS26LS32 and DS26LS33 in that input pull-up and pull-down resistors are included which prevent output oscillation on unused channels.

Each version provides an enable and disable function common to all four receivers and features TRI-STATE® outputs with 8 mA sink capability. Constructed using low power Schottky processing, these devices are available over the full military and commercial operating temperature ranges.

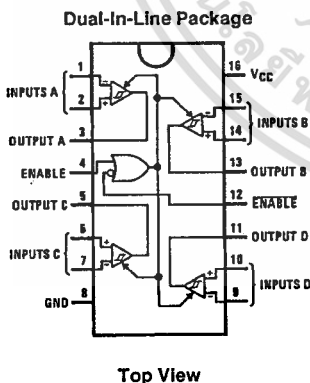
Features

- High differential or common-mode input voltage ranges of $\pm 7V$ on the DS26LS32 and DS26LS32A and $\pm 15V$ on the DS26LS33 and DS26LS33A
- $\pm 0.2V$ sensitivity over the input voltage range on the DS26LS32 and DS26LS32A, $\pm 0.5V$ sensitivity on the DS26LS33 and DS26LS33A
- DS26LS32 and DS26LS32A meet all requirements of RS-422 and RS-423
- 6k minimum input impedance
- 100 mV input hysteresis on the DS26LS32 and DS26LS32A, 200 mV on the DS26LS33 and DS26LS33A
- Operation from a single 5V supply
- TRI-STATE outputs, with choice of complementary output enables for receiving directly onto a data bus

Logic Diagram



Connection Diagram



Truth Table

ENABLE	ENAB \bar{L}	Input	Output
0	1	X	Hi-Z
See Note Below		$V_{ID} \geq V_{TH} (Max)$	1
		$V_{ID} \leq V_{TH} (Min)$	0

Hi-Z = TRI-STATE

Note: Input conditions may be any combination not defined for ENABLE and ENAB \bar{L} .

Order Number DS26LS32CM, DS26LS32CN, DS26LS32MJ, DS26LS32ACM, DS26LS32ACN, DS26LS33CN, DS26LS33MJ or DS26LS33ACN See NS Package Number J16A, M16A or N16A

For Complete Military 883 Specifications, See RETS Data Sheet.

Order Number DS26LS32MJ/883, DS26LS32MW/883, DS26LS33MJ/883, DS26LS33MW/883 See NS Package Number J16A or W16A

TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Common-Mode Range	±25V
Differential Input Voltage	±25V
Enable Voltage	7V
Output Sink Current	50 mA
Maximum Power Dissipation* at 25°C	
Cavity Package	1433 mW
Molded Dip Package	1362 mW
SO Package DS26LS32	1002 mW
DS26LS32A	1051 mW

*Derate cavity package 9.6 mW/°C above 25°C; derate molded DIP package 10.9 mW/°C above 25°C.

Derate SO Package 8.01 mW/°C for DS26LS32
8.41 mW/°C for DS26LS32A

Storage Temperature Range	-65°C to +165°C
Lead Temperature (Soldering, 4 seconds)	260°C

Operating Conditions

	Min	Max	Units
Supply Voltage, (V _{CC})			
DS26LS32M, DS26LS33M (MIL)	4.5	5.5	V
DS26LS32C, DS26LS33C (COML)	4.75	5.25	V
Temperature, (T _A)			
DS26LS32M, DS26LS33M (MIL)	-55	+125	°C
DS26LS32C, DS26LS33C (COML)	0	+70	°C

Electrical Characteristics over the operating temperature range unless otherwise specified (Notes 2, 3 and 4)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V _{TH}	Differential Input Voltage	V _{OUT} = V _{OH} or V _{OL} DS26LS32, DS26LS32A, -7V ≤ V _{CM} ≤ +7V	-0.2	±0.07	0.2	V
		DS26LS33, DS26LS33A, -15V ≤ V _{CM} ≤ +15V	-0.5	±0.14	0.5	V
R _{IN}	Input Resistance	-15V ≤ V _{CM} ≤ +15V (One Input AC GND)	6.0	8.5		kΩ
I _{IN}	Input Current (Under Test)	V _{IN} = 15V, Other Input -15V ≤ V _{IN} ≤ +15V			2.3	mA
		V _{IN} = -15V, Other Input -15V ≤ V _{IN} ≤ +15V			-2.8	mA
V _{OH}	Output High Voltage	V _{CC} = Min, ΔV _{IN} = 1V, V _{ENABLE} = 0.8V, I _{OH} = -440 μA				
		Commercial	2.7	4.2		V
V _{OL}	Output Low Voltage	V _{CC} = Min, ΔV _{IN} = -1V, V _{ENABLE} = 0.8V				
		I _{OL} = 4 mA			0.4	V
V _{IL}	Enable Low Voltage	I _{OL} = 8 mA			0.45	V
					0.8	V
V _{IH}	Enable High Voltage		2.0			V
V _I	Enable Clamp Voltage	V _{CC} = Min, I _{IN} = -18 mA			-1.5	V
I _O	OFF-State (High Impedance) Output Current	V _{CC} = Max				
		V _O = 2.4V			20	μA
I _{IL}	Enable Low Current	V _{IN} = 0.4V			-0.36	mA
		V _{IN} = 2.7V			20	μA
I _{IH}	Enable High Current	V _{IN} = 2.7V			20	μA
I _{SC}	Output Short-Circuit Current	V _O = 0V, V _{CC} = Max, ΔV _{IN} = 1V	-15		-85	mA
I _{CC}	Power Supply Current	V _{CC} = Max, All V _{IN} = GND, Outputs Disabled				
		DS26LS32, DS26LS32A		52	70	mA
I _I	Input High Current	V _{IN} = 5.5V			100	μA
					100	μA
V _{HYST}	Input Hysteresis	T _A = 25°C, V _{CC} = 5V, V _{CM} = 0V				
		DS26LS32, DS26LS32A		100		mV
V _{HYST}	Input Hysteresis					
		DS26LS33, DS26LS33A		200		mV

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All currents into device pins are shown as positive, all currents out of device pins are shown as negative, all voltages are referenced to ground, unless otherwise specified. All values shown as max or min are so classified on absolute value basis.

Note 3: All typical values are V_{CC} = 5V, T_A = 25°C.

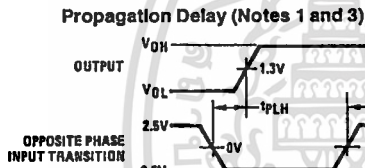
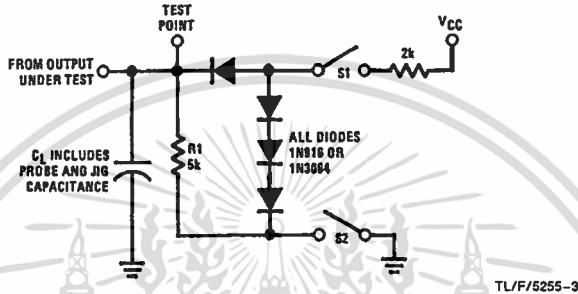
Note 4: Only one output at a time should be shorted.

Switching Characteristics $V_{CC} = 5V, T_A = 25^{\circ}C$

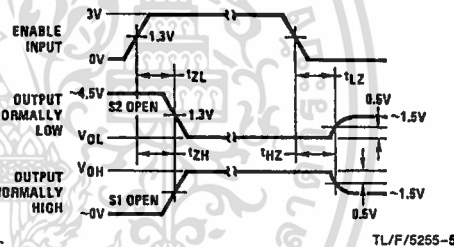
Symbol	Parameter	Conditions	DS26LS32/DS26LS33			DS26LS32A/DS26LS33A			Units
			Min	Typ	Max	Min	Typ	Max	
t_{PLH}	Input to Output	$C_L = 15\text{ pF}$		17	25		23	35	ns
t_{PHL}				17	25		23	35	ns
t_{LZ}	ENABLE to Output	$C_L = 5\text{ pF}$		20	30		15	30	ns
t_{HZ}				15	22		20	25	ns
t_{ZL}	ENABLE to Output	$C_L = 15\text{ pF}$		15	22		14	22	ns
t_{ZH}				15	22		15	22	ns

AC Test Circuit and Switching Time Waveforms

Load Test Circuit for TRI-STATE Outputs



Enable and Disable Times (Notes 2 and 3)



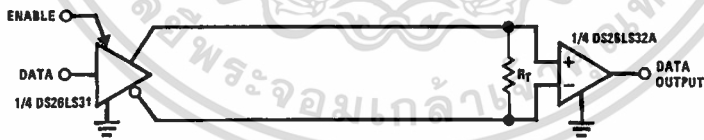
Note 1: Diagram shown for ENABLE low.

Note 2: S1 and S2 of load circuit are closed except where shown.

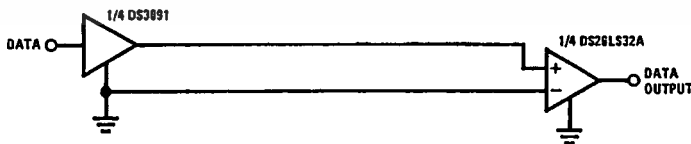
Note 3: Pulse generator for all pulses: Rate = 1.0 MHz; $Z_0 = 50\Omega$; $t_r \leq 6\text{ ns}$; $t_f \leq 6.0\text{ ns}$.

Typical Applications

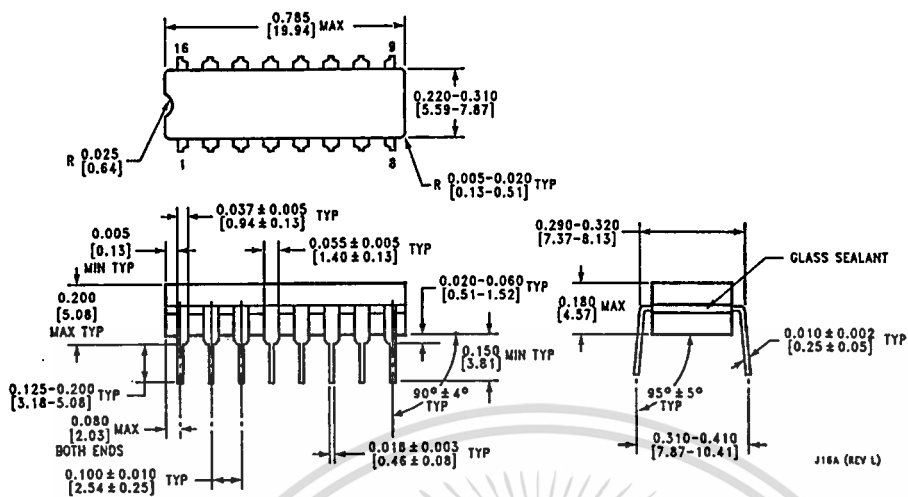
Two-Wire Balanced Interface—RS-422



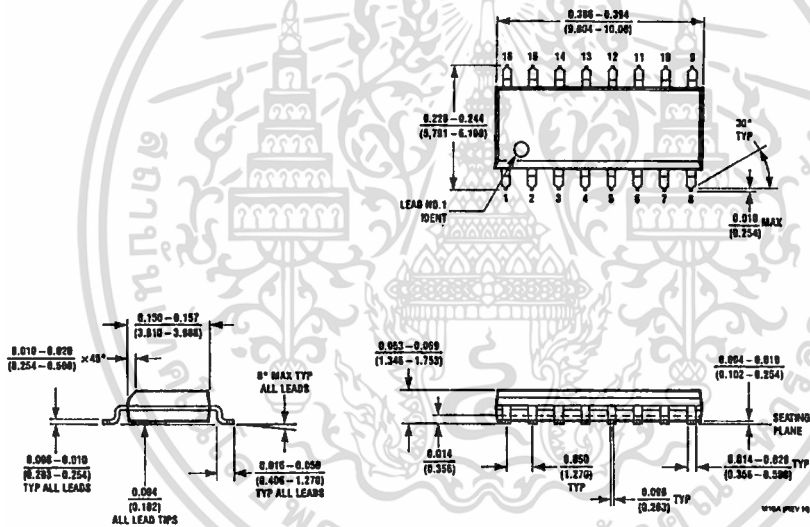
Single Wire with Driver Ground Reference—RS-423



Physical Dimensions inches (millimeters)



Ceramic Dual-In-Line Package (J)
 Order Number DS26LS32MJ, DS26LS33MJ, DS26LS32MJ/883 or DS26LS33MJ/883
 NS Package Number J16A

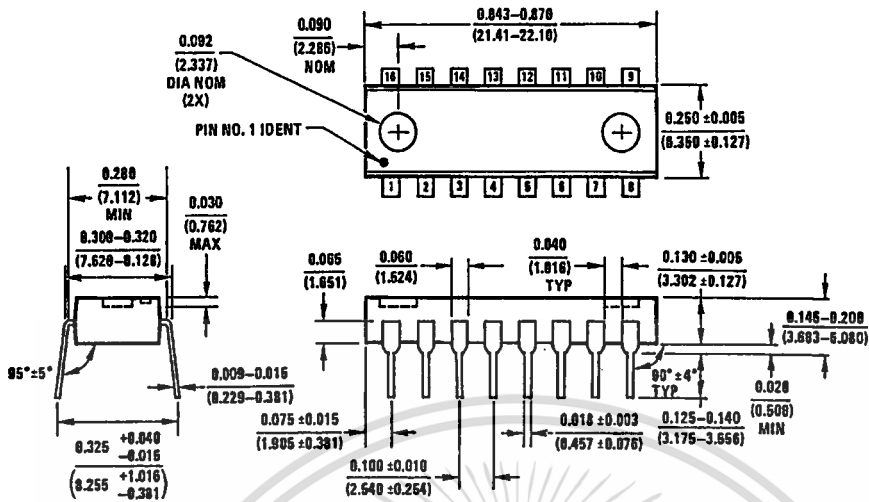


Small Outline Package (M)
 Order Number DS26LS32CM or DS26LS32ACM
 NS Package Number M16A

<http://www.national.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
 Order Number DS26LS32CN, DS26LS32ACN, DS26LS33CN or DS26LS33ACN
 NS Package Number N16A

N16A (REV. E)

