

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การใช้ MRAC เพื่อควบคุมตำแหน่งมอเตอร์กระแสตรง

POSITION CONTROL OF DC MOTORS BY MRAC



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

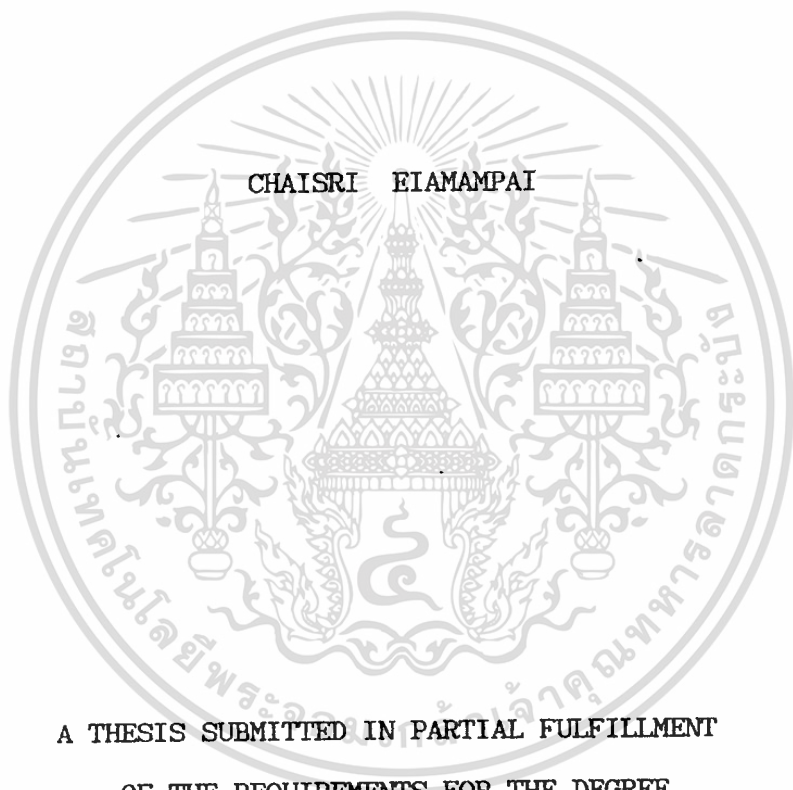
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2534

ISBN 974-8157-38-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POSITION CONTROL OF DC MOTORS BY MRAC



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEER IN ELECTRICAL ENGINEERING
GRADUATE SCHOOL

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

1991

ISBN 974-8157-38-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การใช้ MRAC เพื่อควบคุมตำแหน่งมอเตอร์กระแสตรง POSITION CONTROL OF DC MOTORS BY MRAC
นักศึกษา	นายชัยศรี เอี่ยมอำไพ
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. โยธิน เปรมปราณีรัชต์
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
พ.ศ.	2534

บทคัดย่อ

วิทยานิพนธ์นี้เป็นการนำเอาตัวควบคุมแอดาปทีฟ (adaptive control) แบบ MRAC (Model Reference Adaptive Control) มาใช้ควบคุมตำแหน่งมอเตอร์กระแสตรงเพื่อให้ได้ผลตอบสนองดีที่สุด MRAC สามารถควบคุมตำแหน่งของระบบไว้ได้อย่างถูกต้องเที่ยงตรง แม้ว่าโหลดจะเปลี่ยนแปลงอย่างทันทีทันใดก็ตามซึ่งเป็นข้อดีกว่าตัวควบคุมแบบดั้งเดิม แอดาปทีฟคอนโทรลอัลกอริทึม (adaptive control algorithm) ของ MRAC ถูกออกแบบโดยวิธีที่ 2 ของไลฟฟู่นอวจึงทำให้แน่ใจได้ว่าระบบจะมีเสถียรภาพที่ดี MRAC ที่วิจัยในวิทยานิพนธ์นี้เป็นชนิดไม่ต่อเนื่อง (discrete) จึงจำเป็นต้องแปลงทรานส์เฟอร์ฟังก์ชันของระบบควบคุมตำแหน่งจากชนิดต่อเนื่อง (continuous) ไปเป็นชนิดไม่ต่อเนื่องเสียก่อน รวมทั้งแอดาปทีฟคอนโทรลอัลกอริทึมของ MRAC ต้องเป็นชนิดไม่ต่อเนื่องด้วย นอกจากนี้ยังต้องใช้ตัวสังเกตสถานะ (state observer) เพื่อสร้างตัวแปรสถานะ (state variable) ของระบบควบคุมตำแหน่งเพื่อใช้ในคอนโทรลอัลกอริทึม วงจรขับมอเตอร์กระแสตรงใช้แบบ PWM คอนเวอร์เตอร์ ซึ่งจะมีกำลังสูญเสียน้อยกว่าวงจรขับแบบอื่นๆ ในส่วนท้ายของวิทยานิพนธ์นี้ได้เสนอผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ และผลการทดลองในระบบจริง ทั้งในขณะ ไม่มีโหลดและมีโหลด นอกจากนี้ยังได้ทดลองเปรียบเทียบกับตัวควบคุม PI ด้วย

THESIS TITLE POSITION CONTROL OF DC MOTORS BY MRAC
 STUDY MR. CHAISRI EIAMAMPAI
 THESIS ADVISOR ASSOC. PROF. DR. YOTHIN PREMPRANEERACH
 LEVEL OF STUDY MASTER OF ENGINEER IN ELECTRICAL ENGINEERING
 DEPARTMENT ENGINEER KING MONGKUT'S INSTITUTE OF TECHNOLOGY
 LADKRABANG
 ACADEMIC YEAR 1991

ABSTRACT

This research paper presents a development of adaptive controller based on MRAC (Model Reference Adaptive Control) for use in DC motor position control system in order to obtain optimum time response of the system, and the MRAC can keep steady the high precision of a set point even though the step load has been presented. This is the main advantage of MRAC which is better than the classical method. Adaptive control algorithm of MRAC is designed on Liapunov's second method to ensure that system has a good stability. The MRAC using in this paper is based on a discrete time system so that the transfer function of position control system must be transformed from the continuous system to the discrete system and the control algorithm of MRAC is necessary in the discrete form. In addition, the state observer must be used to produce the state variable of position control system for using in control algorithm. The driver of this system is in form of PWM converter which has a less power loss when compares to the another type. The last part of this research paper has shown the experimental results of computer simulation and actual system. The experiment has been tested in no load and load condition, and also compared with PI controller.

สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	IX
1. บทนำ	1
2. การนำเอาแบบฝึกคอนโทรลมาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรง	3
2.1. บทนำ	3
2.2. การนำ MRAC มาควบคุมตำแหน่งมอเตอร์กระแสตรง	4
2.2.1. โครงสร้างและทรานส์เฟอร์ฟังก์ชันของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง	4
2.2.2. ทฤษฎีพื้นฐานของ MRAC	6
2.3. MRAC ชนิดไม่ต่อเนื่อง	11
2.3.1. การแปลงจากระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง	11
2.3.2. ทฤษฎีของ MRAC ชนิดไม่ต่อเนื่อง	17
2.3.3. การใช้ตัวสังเกตสถานะสร้างตัวแปรสถานะของระบบควบคุมตำแหน่ง	20
2.4. สรุป	25
3. วงจรอิเล็กทรอนิกส์กำลังและวงจรควบคุมตำแหน่ง	26
3.1. บทนำ	26
3.2. โครงสร้างของระบบ PWM คอนเวอร์เตอร์	26
3.2.1. วงจรสร้างความถี่	26
3.2.2. วงจรเปรียบเทียบ	27
3.2.3. วงจรขับ	29
3.3. การนำ MRAC มาใช้ในระบบควบคุมตำแหน่งจริง	31
3.3.1. วงจรตีโค้ดเดอร์	32
3.3.2. วงจรเอาท์พุท	33

3.3.3. วงจรอินพุท	33
3.4. สรุป	39
4. ผลการทดลอง	40
4.1. บทนำ	40
4.2. ผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์	40
4.2.1. ผลการทดลองในขณะ ไม่มี โหลด	40
4.2.2. ผลการทดลองในขณะมี โหลด	42
4.2.3. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะ ไม่มี โหลด	45
4.2.4. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะมี โหลด	45
4.3. ผลการทดลองในระบบจริง	47
4.3.1. ผลการทดลองในขณะ ไม่มี โหลด	47
4.3.2. ผลการทดลองในขณะมี โหลด	51
4.3.3. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะ ไม่มี โหลด	58
4.3.4. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะมี โหลด	60
4.4. สรุป	68
5. บทสรุป	73
กิตติกรรมประกาศ	75
เอกสารอ้างอิง	76
ภาคผนวก ก. การวิเคราะห์เสถียรภาพของ ไลฟูนอฟ	79
ภาคผนวก ข. กฎเกณฑ์ของซิลเวสเตอร์สำหรับเมตริกซ์สมมาตรของเลขจำนวนจริง	85
ภาคผนวก ค. ทฤษฎีการล่มสลาย	87
ภาคผนวก ง. การสามารถสังเกตได้	90
ภาคผนวก จ. การออกแบบตัวควบคุม PI	93
ภาคผนวก ฉ. โปรแกรมจำลองผลตอบสนอง และควบคุมตำแหน่งโดย MRAC	96
ภาคผนวก ช. พิกัดของมอเตอร์	132
ภาคผนวก ซ. ผลงานวิจัยที่ได้รับการเผยแพร่ในระหว่างศึกษาปริญญาโท	133

สารบัญรูป

รูปที่ 2.1	แสดงโครงสร้างพื้นฐานการใช้ MRAC เพื่อควบคุมตำแหน่งมอเตอร์ กระแสตรง	3
รูปที่ 2.2	แสดงโครงสร้างของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง	4
รูปที่ 2.3	แสดงผลตอบสนองของโมเดลอ้างอิงต่อคำสั่งอินพุท	7
รูปที่ 2.4	แสดงการประมาณค่าโดยวิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน	12
รูปที่ 3.1	แสดงโครงสร้างของ PWM คอนเวอร์เตอร์	26
รูปที่ 3.2	แสดงวงจรสร้างสัญญาณสามเหลี่ยม	27
รูปที่ 3.3	แสดงวงจรเปรียบเทียบ	27
รูปที่ 3.4	แสดงเอาต์พุทของวงจรเปรียบเทียบ	28
รูปที่ 3.5	แสดงวงจรฮับเบส	29
รูปที่ 3.6	แสดงการต่อวงจรกำลังและมอเตอร์กระแสตรง	29
รูปที่ 3.7	แสดงวงจรจัดเฟส	30
รูปที่ 3.8	แสดงทิศทางของกระแสเมื่อมอเตอร์หมุนเดินหน้า	31
รูปที่ 3.9	แสดงทิศทางของกระแสเมื่อมอเตอร์หมุนถอยหลัง	31
รูปที่ 3.10	แสดงวงจรตีโค้ดเตอร์	32
รูปที่ 3.11	แสดงวงจรเอาต์พุท	33
รูปที่ 3.12	แสดงความต่างเฟสของพัลส์จากพัลส์เอ็นโค้ดเตอร์	34
รูปที่ 3.13	แสดงวงจรอินพุท	35
รูปที่ 3.14	แสดงการต่ออุปกรณ์ร่วมกัน	36
รูปที่ 3.15	รูปถ่ายแสดงวงจรกำลัง	37
รูปที่ 3.16	รูปถ่ายแสดงวงจรตีโค้ดเตอร์	37
รูปที่ 3.17	รูปถ่ายแสดงวงจรเอาต์พุทและวงจรอินพุท	38
รูปที่ 3.18	รูปถ่ายแสดงอุปกรณ์การทดลอง	38
รูปที่ 4.1	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ	
	3 เรเดียน/วินาที	41

รูปที่ 4.2	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที	41
รูปที่ 4.3	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และมีสแตปโหลดขนาด 25 % ของคำสั่งอินพุต	42
รูปที่ 4.4	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที และมีสแตปโหลดขนาด 25 % ของคำสั่งอินพุต	43
รูปที่ 4.5	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และมีโหลดขนาด 25 % ของผลตอบสนอง	43
รูปที่ 4.6	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที และมีโหลดขนาด 25 % ของผลตอบสนอง	44
รูปที่ 4.7	แสดงผลตอบสนองเปรียบเทียบระหว่างตัวควบคุม MRAC และตัวควบคุม PI	45
รูปที่ 4.8	แสดงผลตอบสนองเปรียบเทียบเมื่อมีสแตปโหลดขนาด 25 % ของคำสั่งอินพุต	46
รูปที่ 4.9	แสดงผลตอบสนองเปรียบเทียบเมื่อมีโหลดขนาด 25 % ของผลตอบสนอง	46
รูปที่ 4.10	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	48
รูปที่ 4.11	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	48
รูปที่ 4.12	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	49
รูปที่ 4.13	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	49
รูปที่ 4.14	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	50
รูปที่ 4.15	แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	50
รูปที่ 4.16	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	51

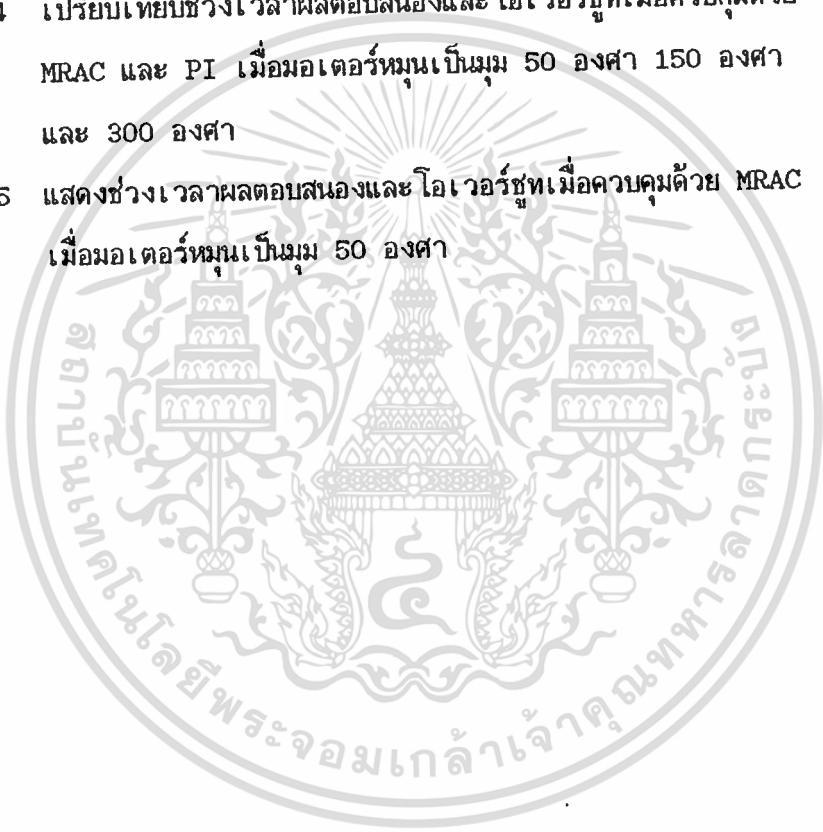
VII

รูปที่ 4.17	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	52
รูปที่ 4.18	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	52
รูปที่ 4.19	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	53
รูปที่ 4.20	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	53
รูปที่ 4.21	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	54
รูปที่ 4.22	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	54
รูปที่ 4.23	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	55
รูปที่ 4.24	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	55
รูปที่ 4.25	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา	56
รูปที่ 4.26	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	56
รูปที่ 4.27	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีขั้วรวมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา	57
รูปที่ 4.28	แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 50 องศา	59
รูปที่ 4.29	แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 150 องศา	59
รูปที่ 4.30	แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 300 องศา	60
รูปที่ 4.31	แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 50 องศา	60

รูปที่ 4.32	แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 150 องศา	61
รูปที่ 4.33	แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 300 องศา	61
รูปที่ 4.34	แสดงผลตอบสนองเปรียบเทียบเมื่อปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 50 องศา	62
รูปที่ 4.35	แสดงผลตอบสนองเปรียบเทียบเมื่อปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 150 องศา	62
รูปที่ 4.36	แสดงผลตอบสนองเปรียบเทียบเมื่อปิดขั้วเครื่องกำเนิดไฟฟ้า มอเตอร์หมุนเป็นมุม 300 องศา	63
รูปที่ 4.37	แสดงผลการทดลองเมื่อ โมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	65
รูปที่ 4.38	แสดงผลการทดลองเมื่อ โมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	65
รูปที่ 4.39	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีธรรมชาติ 10 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	66
รูปที่ 4.40	แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีธรรมชาติ 17 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	66
รูปที่ 4.41	แสดงผลตอบสนองเมื่อปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีธรรมชาติ 10 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	67
รูปที่ 4.42	แสดงผลตอบสนองเมื่อปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดล อ้างอิงมีธรรมชาติ 17 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา	67
รูปที่ ค.1	แสดงการล่มสัญญาณ	87
รูปที่ ค.2	แสดงสเปกตรัมของสัญญาณต่อเนื่อง	87
รูปที่ ค.3	แสดงสเปกตรัมของสัญญาณที่ถูกล่ม	89
รูปที่ จ.1	แผนภาพแสดงลำดับการทำงานของโปรแกรมจำลองผลตอบสนอง ด้วยคอมพิวเตอร์	96
รูปที่ จ.2	แผนภาพแสดงลำดับการทำงานในระบบจริง	124

สารบัญตาราง

ตารางที่ 4.1	แสดงช่วงเวลาผลตอบแทนและโอเวอร์ชู้ทเมื่อควบคุมด้วย MRAC	44
ตารางที่ 4.2	เปรียบเทียบช่วงเวลาผลตอบแทนและโอเวอร์ชู้ทเมื่อควบคุมด้วย MRAC และ PI	47
ตารางที่ 4.3	แสดงช่วงเวลาผลตอบแทนและโอเวอร์ชู้ทเมื่อควบคุมด้วย MRAC เมื่อมอเตอร์หมุนเป็นมุม 50 องศา 150 องศา และ 300 องศา	58
ตารางที่ 4.4	เปรียบเทียบช่วงเวลาผลตอบแทนและโอเวอร์ชู้ทเมื่อควบคุมด้วย MRAC และ PI เมื่อมอเตอร์หมุนเป็นมุม 50 องศา 150 องศา และ 300 องศา	64
ตารางที่ 4.5	แสดงช่วงเวลาผลตอบแทนและโอเวอร์ชู้ทเมื่อควบคุมด้วย MRAC เมื่อมอเตอร์หมุนเป็นมุม 50 องศา	68



บทที่ 1.

บทนำ

ปัจจุบันมอเตอร์กระแสตรงถูกนำมาใช้งานทางอุตสาหกรรมอย่างกว้างขวาง โดยเฉพาะในการควบคุมความเร็วและการควบคุมตำแหน่ง ตัวอย่างการใช้งานทางอุตสาหกรรม เช่น แขนหุ่นยนต์ในโรงงานอุตสาหกรรมขนาดใหญ่ ระบบ CNC (Computer Numerical Control) หรือระบบติดตามของเรดาร์ เป็นต้น ในอดีตที่ผ่านมาการควบคุมตำแหน่งมอเตอร์กระแสตรงไม่ต้องการความเที่ยงตรงมากนัก ดังนั้นตัวควบคุมแบบดั้งเดิม เช่น ตัวควบคุม PI หรือ PID [1-6] สามารถตอบสนองความต้องการในทางอุตสาหกรรมได้ แต่ในปัจจุบันระบบควบคุมตำแหน่งต้องการความเที่ยงตรงมากยิ่งขึ้น นอกจากนี้ตัวควบคุมแบบดั้งเดิมไวต่อการเปลี่ยนแปลงพารามิเตอร์และการเปลี่ยนแปลงโหลด ทำให้พฤติกรรมการตอบสนองของระบบเปลี่ยนแปลงไปตามสภาพของการทำงานยากต่อการควบคุม ตัวควบคุมแบบดั้งเดิมจึงไม่สามารถตอบสนองความต้องการในทางอุตสาหกรรมได้ จากเหตุจูงใจนี้ทำให้มีความพยายามที่จะนำเอาทฤษฎีควบคุมสมัยใหม่ (modern control theory) มาใช้สำหรับควบคุมตำแหน่งมอเตอร์กระแสตรงมากยิ่งขึ้น

จากความพยายามที่จะนำเอาทฤษฎีควบคุมสมัยใหม่ มาใช้ควบคุมตำแหน่งมอเตอร์กระแสตรง ทำให้ได้ตัวควบคุมที่สามารถปรับพารามิเตอร์ของตัวเองเพื่อบังคับให้ผลตอบสนองของระบบควบคุมตำแหน่งถูกต้องเที่ยงตรง เรียกตัวควบคุมแบบนี้ว่าตัวควบคุมแอดaptive (adaptive controller) [7-18] ซึ่งมีหลายแบบ เช่น เซลฟ์จูนนิ่ง (self-tuning) [7] วาเรียเบิลสตรักเจอร์ (Variable structure) [8] และ MRAC (Model Reference Adaptive Control) [1,2,7,9-18]

วิทยานิพนธ์นี้เลือกใช้ตัวควบคุมแบบ MRAC หลักการของ MRAC คือสร้างโมเดลอ้างอิง (reference model) ซึ่งมีผลตอบสนองต่อคำสั่งอินพุตตามต้องการ และจะพยายามบังคับให้ผลตอบสนองของระบบควบคุมตำแหน่งเหมือนกับผลตอบสนองของโมเดลอ้างอิง โดยอาศัยความแตกต่างระหว่างผลตอบสนองทั้งสองสำหรับปรับค่าพารามิเตอร์ของตัวควบคุม ดังนั้นผลตอบสนองของระบบควบคุมตำแหน่ง จะสามารถติดตามผลตอบสนองของโมเดลอ้างอิงได้อย่างถูกต้องเที่ยงตรงตลอดเวลาโดยไม่สนใจการเปลี่ยนแปลงพารามิเตอร์ หรือการเปลี่ยนแปลงของโหลด

วิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 5 บท ดังต่อไปนี้

บทที่ 1 เป็นบทนำจะอธิบายถึงวัตถุประสงค์ของวิทยานิพนธ์นี้ และเนื้อหาคร่าวๆ ของวิทยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิพนธ์ฉบับนี้

บทที่ 2 อธิบายถึงการนำอแดปทีฟคอนโทรลมาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรง ทฤษฎีพื้นฐานของ MRAC และทฤษฎีของ MRAC ชนิดไม่ต่อเนื่อง (Discrete MRAC) การแปลงระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง การสร้างอแดปทีฟคอนโทรลอัลกอริทึม (adaptive control algorithm) และการใช้ตัวสังเกตสถานะ (state observer) เพื่อสร้างตัวแปรสถานะ (state variable) ของระบบควบคุมตำแหน่ง

บทที่ 3 อธิบายถึงโครงสร้างของวงจรรันแบบ PWM (Pulse Width Modulation) [19-22] และการนำ MRAC มาใช้ในระบบควบคุมตำแหน่งจริงๆ

บทที่ 4 เป็นการทดลองในสภาพต่างๆ กัน เช่น ในขณะไม่มีโหลด และในขณะมีโหลด โดยผลการทดลองมีทั้งการจำลองผลตอบสนองด้วยคอมพิวเตอร์ (computer simulation) และการทดลองในระบบจริงๆ นอกจากนี้ยังได้ทำการทดลองเปรียบเทียบกับตัวควบคุม PI ด้วย

บทที่ 5 เป็นบทสรุป

ส่วนสุดท้ายของวิทยานิพนธ์นี้เป็นภาคผนวก ประกอบด้วยทฤษฎีการวิเคราะห์เสถียรภาพของไลฟูนอฟ (Liapunov's stability analysis) [1,2] กฎเกณฑ์ของซิลเวสเตอร์สำหรับเมตริกซ์สมมาตรของเลขจำนวนจริง (Sylvester's criterion for real symmetrical matrix) [1,2] เงื่อนไขการสามารถสังเกตได้ (observability condition) [1,2] ทฤษฎีการสุ่มสัญญาณ (sampling theorem) [2] การออกแบบตัวควบคุม PI โดยวิธีซิมเมตริกคอลลอปติวมัม [4] โปรแกรมที่ใช้ในวิทยานิพนธ์ ขนาดพิกัดของมอเตอร์ และผลงานที่ได้รับการเผยแพร่ในระหว่างการศึกษาปริญญาโท

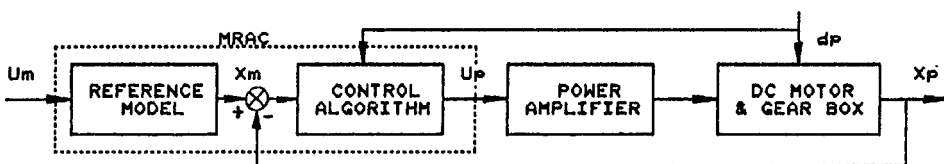
บทที่ 2.

การนำอแดปทีฟคอนโทรลมาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรง

2.1. บทนำ

จากความพยายามที่จะนำเอาทฤษฎีควบคุมสมัยใหม่มาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรง ทำให้ได้ตัวควบคุมอแดปทีฟหลายชนิด เช่น เซลฟ์จูนนิ่ง วาริเอเบิลสตรักเจอร์ และ MRAC เป็นต้น หลักการพื้นฐานของตัวควบคุมเหล่านี้คือ การปรับพารามิเตอร์หรือโครงสร้างได้อย่างอัตโนมัติเพื่อนำระบบไปสู่จุดมุ่งหมายที่ต้องการ โดยไม่สนใจการเปลี่ยนแปลงพารามิเตอร์หรือการเปลี่ยนแปลงโหลดของระบบ ในอดีตแม้ว่าความพยายามนี้จะประสบผลสำเร็จในทางทฤษฎี แต่ไม่สามารถนำมาใช้งานได้จริง เนื่องจากทฤษฎีควบคุมสมัยใหม่มีความซับซ้อน และต้องการการคำนวณมากมาย การสร้างฮาร์ดแวร์จึงเป็นเรื่องยุ่งยาก แต่ในปัจจุบันไมโครโปรเซสเซอร์สามารถคำนวณได้เร็วขึ้น และอุปกรณ์อิเล็กทรอนิกส์มีความเร็วสูงขึ้นทำให้สามารถนำเอาทฤษฎีควบคุมสมัยใหม่มาใช้งานได้อย่างจริงจัง

โครงสร้างพื้นฐานของการนำ MRAC มาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรงแสดงดังรูปที่ 2.1 โดยโมเดลอ้างอิงทำหน้าที่สร้างผลตอบสนองต่อคำสั่งอินพุตตามต้องการ ผลตอบสนองของระบบควบคุมตำแหน่งจะถูกนำมาเปรียบเทียบกับผลตอบสนองของโมเดลอ้างอิง ค่าผิดพลาดที่ได้จะถูกนำมาใช้ปรับอแดปทีฟคอนโทรลอัลกอริทึม เพื่อสร้างสัญญาณควบคุม (u_p) สำหรับบังคับผลตอบสนองของระบบควบคุมตำแหน่งให้เหมือนกับผลตอบสนองของโมเดลอ้างอิง ดังนั้นการเปลี่ยนแปลงพารามิเตอร์ หรือการเปลี่ยนแปลงโหลดจะไม่มีผลกระทบต่อผลตอบสนองของระบบ โมเดลอ้างอิงและอแดปทีฟคอนโทรลอัลกอริทึมอาจจะเป็นฮาร์ดแวร์ หรือเป็นเพียงโปรแกรมในคอมพิวเตอร์ก็ได้ สำหรับการทําวิจัยในวิทยานิพนธ์นี้ ทั้งโมเดลอ้างอิง และอแดปทีฟคอนโทรลอัลกอริทึมถูกเขียนเป็นโปรแกรมไว้ในไมโครคอมพิวเตอร์ อแดปทีฟคอนโทรลอัลกอริทึมได้ถูกออกแบบโดยวิธีที่ 2 ของไลฟเนอร์ [ภาคผนวก ก.] จึงทำให้เชื่อได้ว่าระบบที่ได้จะมีเสถียรภาพดี



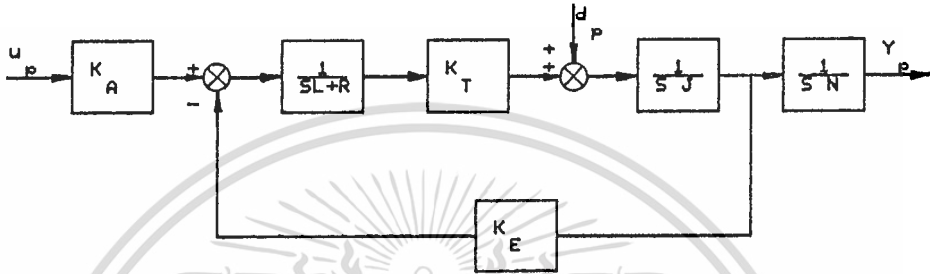
รูปที่ 2.1 แสดงโครงสร้างพื้นฐานการใช้ MRAC เพื่อควบคุมตำแหน่งมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2. การนำ MRAC มาควบคุมตำแหน่งมอเตอร์กระแสตรง

2.2.1. โครงสร้างและทรานส์เฟอร์ฟังก์ชันของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง

โครงสร้างของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง[1] เฉพาะในแง่ของวงจรรยาย มอเตอร์กระแสตรง และเฟืองทดรอบ แสดงดังรูปที่ 2.2 มอเตอร์กระแสตรงที่ใช้ในงานวิจัย เป็นของบริษัท Sanyo Denki Co., Ltd รุ่น U 718(T)



รูปที่ 2.2 แสดงโครงสร้างของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง

- โดยที่ K_A คือ อัตราขยายของวงจรรยาย (amplifier gain)
 L คือ ค่าความเหนี่ยวนำขดลวดของมอเตอร์ (armature inductance)
 R คือ ค่าความต้านทานขดลวดของมอเตอร์ (armature resistance)
 K_T คือ ค่าคงที่ทอร์ก (torque constant)
 J คือ ค่าความเฉื่อย (inertia)
 N คือ อัตราการทดรอบของเฟืองทด (gear ratio)
 K_E คือ ค่าโวลต์เตจคงที่ (voltage constant)
 u_p คือ สัญญาณควบคุมที่ได้มาจากแอมป์ที่ฟคอน ไทโรลอัลกอริทึม
 y_p คือ องศาที่แกนของเฟืองทดรอบหมุนไป
 d_p คือ โหลดจากภายนอกที่กระทำต่อแกนของเฟืองทดรอบ

จากรูปที่ 2.2 ทำให้สามารถเขียนทรานส์เฟอร์ฟังก์ชันแบบรูปเปิดได้เป็น

$$\frac{y_p(s)}{u_p(s)} = \frac{a_4}{s^3 + s^2 a_3 + s a_2} \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

$$\frac{y_p(s)}{d_p(s)} = \frac{sa_5 + a_6}{s^3 + s^2a_3 + sa_2} \quad (2.2)$$

โดยที่

$$\begin{aligned} a_2 &= K_T K_E / LJ \\ a_3 &= R/L \\ a_4 &= K_A K_T / NLJ \\ a_5 &= 1/NJ \\ a_6 &= R/NLJ \end{aligned} \quad (2.3)$$

จากสมการที่ (2.1) และ (2.2) สามารถนำมาเขียนสมการสถานะได้เป็น

$$\begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{p3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -a_2 & -a_3 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & a_5 \\ a_4 & 0 \end{bmatrix} \begin{bmatrix} u_p \\ d_p \end{bmatrix} \quad (2.4)$$

$$y_p = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{bmatrix} \quad (2.5)$$

โดยที่ x_{p1} คือ ตำแหน่ง
 x_{p2} คือ ความเร็ว
 x_{p3} คือ ความเร่ง

สมการที่ (2.4) และ (2.5) อาจเขียนเป็น

$$\dot{X}_p = A_p X_p + B_p U_p \quad (2.6)$$

$$Y_p = C_p X_p \quad (2.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ทฤษฎีพื้นฐานของ MRAC

ดังที่ได้กล่าวมาแล้วข้างต้น แนวความคิดพื้นฐานของการควบคุมมอเตอร์กระแสตรงโดยวิธี MRAC คือ การบังคับให้ผลตอบสนองของระบบควบคุมตำแหน่งมีผลตอบสนองเหมือนกับผลตอบสนองของโมเดลอ้างอิง ซึ่งออกแบบที่คอนโทรลเลอร์จะเป็นตัวสร้างสัญญาณควบคุม โดยอาศัยความแตกต่างระหว่างผลตอบสนองทั้งสอง [1,7-13] ดังนั้นจุดสำคัญของการควบคุมโดยวิธีนี้คือการสร้างโมเดลอ้างอิงให้มีผลตอบสนองต่อคำสั่งอินพุตตามต้องการ และการสร้างออกแบบที่คอนโทรลเลอร์ที่สามารถบังคับผลตอบสนองของระบบควบคุมตำแหน่งได้

โมเดลอ้างอิงจำเป็นต้องมีคุณสมบัติที่สำคัญคือ มีผลตอบสนองต่อคำสั่งอินพุต คือ มีผลตอบสนองเร็ว โอเวอร์ชูตต่ำ และไม่มีค่าผิดพลาดในสภาวะสงบนิ่ง จากความต้องการนี้จึงเลือกออกแบบโมเดลอ้างอิงโดยใช้กฎเกณฑ์ ITAE (Integral Time Multiplied Absolute Error Criterion) [1] หลักการออกแบบโดยวิธีนี้ คือ การทำให้ตัวชี้พฤติกรรม J (performance index) มีค่าต่ำที่สุด โดย

$$J = \int_0^{\infty} t |e(t)| dt \quad (2.8)$$

กฎเกณฑ์ ITAE ได้ถูกสรุปเพื่อความสะดวกสำหรับการออกแบบระบบอันดับใดๆ เป็น

$$\frac{C(s)}{R(s)} = \frac{a_n}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (2.9)$$

โดย $C(s)$ คือ ผลตอบสนองต่อคำสั่งอินพุต

$R(s)$ คือ คำสั่งอินพุต

เนื่องจากระบบควบคุมตำแหน่งมีอันดับ 3 ดังนั้นโมเดลอ้างอิงจึงมีอันดับ 3 ด้วย จะได้

$$\frac{y_m(s)}{u_m(s)} = \frac{m_4}{s^3 + s^2 m_3 + s m_2 + m_1} \quad (2.10)$$

$$\text{ซึ่ง } \frac{y_m(s)}{u_m(s)} = \frac{W_n^3}{s^3 + s^2(1.75W_n) + s(2.15W_n^2) + W_n^3} \quad (2.11)$$

โดย $y_m(s)$ คือ ผลตอบสนองต่อคำสั่งอินพุท

$u_m(s)$ คือ คำสั่งอินพุท

W_n คือ ความถี่ธรรมชาติ (natural frequency) ของระบบ



รูปที่ 2.3 แสดงผลตอบสนองของโมเดลอ้างอิงต่อคำสั่งอินพุท

จากสมการที่ (2.10) สามารถนำมาเขียนในรูปของสมการสถานะเป็น

$$\begin{bmatrix} \dot{X}_{m1} \\ \dot{X}_{m2} \\ \dot{X}_{m3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -m_1 & -m_2 & -m_3 \end{bmatrix} \begin{bmatrix} X_{m1} \\ X_{m2} \\ X_{m3} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ m_4 \end{bmatrix} [u_m] \quad (2.12)$$

$$y_m = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{m1} \\ X_{m2} \\ X_{m3} \end{bmatrix} \quad (2.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ

$$\dot{X}_m = A_m X_m + B_m U_m \quad (2.14)$$

$$Y_m = C_m X_m \quad (2.15)$$

หลังจากที่ออกแบบโมเดลอ้างอิงแล้วจึงออกแบบอแดปทีฟคอนโทรลอัลกอริทึม กำหนดให้ e เป็น เวกเตอร์ผิดพลาด (error vector) โดย

$$e = X_m - X_p = \begin{bmatrix} X_{m1} - X_{p1} \\ X_{m2} - X_{p2} \\ X_{m3} - X_{p3} \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (2.16)$$

หลักการพื้นฐานในการออกแบบอแดปทีฟคอนโทรลอัลกอริทึมคือ ทำให้เวกเตอร์ผิดพลาดเป็น ศูนย์ นั่นคือ $X_m = X_p$, $\dot{X}_m = \dot{X}_p$ และ $e = \dot{e} = 0$ โดยเลือกค่า u_p ที่เหมาะสม

$$\begin{aligned} \dot{e} &= \dot{X}_m - \dot{X}_p \\ &= A_m X_m + B_m U_m - A_p X_p - B_p U_p \end{aligned} \quad (2.17)$$

การออกแบบอแดปทีฟคอนโทรลอัลกอริทึม จะออกแบบโดยวิธีที่ 2 ของไลทูนอว [ภาคผนวก ก.] และกำหนดให้ $V(e)$ คือสมการของไลทูนอว

$$V(e) = e^T P e \quad (2.18)$$

โดยที่ P เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงซึ่งมีความหมายเป็นบวก (positive - definiteness real symmetrical matrix) [ภาคผนวก ข.]

$$P = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_2 & P_4 & P_5 \\ P_3 & P_5 & P_6 \end{bmatrix} \quad (2.19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเงื่อนไขของไลฟท์นอกล่าวไว้ สมการสเกลาร์ใดๆ จะเป็นสมการของไลฟท์นอถ้า
เงื่อนไขทั้งสองข้อเป็นจริง

เงื่อนไขที่ 1. สมการนี้มีความหมายเป็นบวก (positive definiteness)

เงื่อนไขที่ 2. ค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลาของสมการนั้น มีความหมายเป็นลบ
(negative definiteness) หรือมีความหมายกึ่งเป็นลบ (negative semidefiniteness)

เนื่องจาก P เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงซึ่งมีความหมายเป็นบวก ดังนั้น

เงื่อนไขที่ 1. เป็นจริง และค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลาคือ

$$\dot{V}(e) = e^T \dot{P}e + e^T P \dot{e} \quad (2.20)$$

เพราะว่า

$$\dot{e} = e^T A_m^T + X_p^T A_m^T + U_m^T B_m^T - X_p^T A_p^T - U_p^T B_p^T \quad (2.21)$$

ดังนั้น

$$\begin{aligned} \dot{V}(e) &= e^T A_m^T P e + [X_p^T A_m^T + U_m^T B_m^T - X_p^T A_p^T - U_p^T B_p^T] P e + e^T P A_m e + e^T P [A_m X_p + B_m U_m - A_p X_p - B_p U_p] \\ &= e^T [-Q]e + 2M \end{aligned} \quad (2.22)$$

$$\text{โดยที่ } Q = A_m^T P + P A_m \quad (2.23)$$

$$M = e^T P (A_m X_p + B_m U_m - A_p X_p - B_p U_p) \quad (2.24)$$

M เป็นปริมาณสเกลาร์ เงื่อนไขที่ 2. จะเป็นจริงถ้า Q มีความหมายเป็นบวก ทำให้ได้ P ตามสมการที่ (2.23) มีความหมายเป็นบวก และ M มีความหมายกึ่งเป็นลบ ($M \leq 0$) ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$V(e)$ มีความหมายเป็นลบ ซึ่งทำได้โดยการเลือกค่า u_p ที่เหมาะสม วิธีที่สะดวกที่สุดคือกำหนด Q เป็นเมตริกซ์ทแยง (diagonal matrix)

$$Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix} \quad (2.25)$$

โดยใช้กฎเกณฑ์ของซิลเวสเตอร์สำหรับเมตริกซ์สมมาตรที่มีความหมายเป็นบวก [ภาคผนวก ข.]

$$q_1 > 0, \quad \begin{vmatrix} q_1 & 0 \\ 0 & q_2 \end{vmatrix} > 0, \quad \begin{vmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{vmatrix} > 0 \quad (2.26)$$

ทำให้สามารถพิสูจน์ได้ว่าเมตริกซ์ Q เป็นมีความหมายเป็นบวก จากสมการที่ (2.23) และ (2.25) ทำให้สามารถหาเมตริกซ์ P ได้ และใช้กฎเกณฑ์ซิลเวสเตอร์พิสูจน์ว่าเมตริกซ์ P มีความหมายเป็นบวกหรือไม่ จากสมการที่ (2.4), (2.12), (2.16), (2.19), (2.24) จะได้

$$M = \begin{bmatrix} E_1 & E_2 & E_3 \\ Z_1 \\ Z_2 \\ Z_3 - a_4 u_p \end{bmatrix} \quad (2.27)$$

$$\begin{aligned} \text{โดยที่ } E_1 &= e_1 P_1 + e_2 P_2 + e_3 P_3 \\ E_2 &= e_1 P_2 + e_2 P_4 + e_3 P_5 \\ E_3 &= e_1 P_3 + e_2 P_5 + e_3 P_6 \\ Z_1 &= 0 \\ Z_2 &= -a_5 d_p \\ Z_3 &= -m_1 x_{P1} + (-m_2 + a_2) x_{P2} + (-m_3 + a_3) x_{P3} + m_4 u_m \end{aligned} \quad (2.28)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$M = E_2 Z_2 + E_3 (Z_3 - a_4 u_p) \quad (2.29)$$

กำหนดให้ $M = 0$ จะได้ u_p ดังสมการที่ (2.30) ซึ่งจะถูกนำไปใช้ควบคุมระบบควบคุมตำแหน่ง

$$u_p = (E_2 Z_2 + E_3 Z_3) / a_4 E_3 \quad (2.30)$$

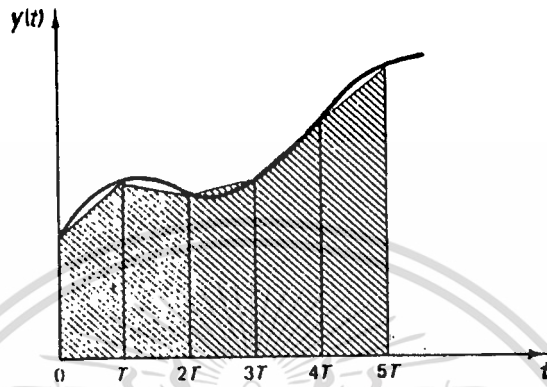
2.3. MRAC ชนิดไม่ต่อเนื่อง

ในอดีตการนำเอาทฤษฎีควบคุมสมัยใหม่มาใช้ในระบบควบคุมยังไม่ประสบความสำเร็จ เนื่องจากทฤษฎีการควบคุมสมัยใหม่ต้องการการคำนวณมากมายทำให้ฮาร์ดแวร์มีความยุ่งยาก แต่ในปัจจุบันไมโครโปรเซสเซอร์มีความสามารถในการคำนวณสูงและรวดเร็ว ทำให้สามารถลดความยุ่งยากของฮาร์ดแวร์ลง เหลือเป็นเพียงโปรแกรมเก็บไว้ในหน่วยความจำเท่านั้น ดังเช่นการวิจัยในวิทยานิพนธ์นี้ ทั้งไมโครอั้งอิงและขอแคบที่ฟคอนโทรลอัลกอริทึมถูกสร้างไว้เป็นโปรแกรม แต่การคำนวณของไมโครโปรเซสเซอร์เป็นชนิดไม่ต่อเนื่อง (discrete) ดังนั้นจึงจำเป็นต้องแปลงสมการของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง ไมโครอั้งอิง และขอแคบที่ฟคอนโทรลอัลกอริทึมเป็นชนิดไม่ต่อเนื่องด้วย

2.3.1. การแปลงระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง

มีหลายวิธีที่จะแปลงระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง เช่น วิธีแบ็คเวิร์ดดิฟเฟอเรนซ์ (Backward difference method) วิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน (Bilinear transformation method) และวิธีโพล-ซีโรแมปปิง (Pole - zero mapping method) เป็นต้น หลังจากแปลงแล้วระบบไม่ต่อเนื่องที่ได้จะต้องมีคุณสมบัติทางไดนามิก (dynamic characteristic) คือ ผลตอบสนองชั่วขณะ (transient response) และผลตอบสนองทางความถี่ (frequency response) ใกล้เคียงกับระบบต่อเนื่องเดิมมากที่สุด ในการวิจัยนี้เลือกการแปลงโดยวิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน [2] ซึ่งจะทำให้ระบบไม่ต่อเนื่องที่ได้มีเสถียรภาพถ้าระบบต่อเนื่องเดิมมีเสถียรภาพ นอกจากเลือกวิธีแปลงแล้วช่วงเวลาสุ่มสัญญาณ T (sampling time) จะต้องมีค่าสูงพอตามทฤษฎีการสุ่มสัญญาณ [ภาคผนวก ค.] ด้วย

หลักการแปลงระบบต่อเนื่องไปเป็นระบบไม่ต่อเนื่องโดยวิธี ไบไลเนียร์ทรานส์ฟอร์มเมชัน คือ การประมาณค่าระหว่างจุดสุ่มสัญญาณเป็นเส้นตรง ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 แสดงการประมาณค่าโดยวิธี ไบไลเนียร์ทรานส์ฟอร์มเมชัน

สมมติระบบต่อเนื่องเดิมคือ

$$Y(s)/X(s) = a/(s + a) \quad (2.31)$$

หรือ $dy/dt = -ay + ax \quad (2.32)$

อินทิเกรตทั้งสองด้านตั้งแต่เวลา 0 ถึง t

$$\int_0^t (dy(t)/dt) \cdot dt = -a \int_0^t y(t) \cdot dt + a \int_0^t x(t) \cdot dt \quad (2.33)$$

สมมติว่าหาค่าของ $y(t)$ ทุกๆ เวลาสุ่มสัญญาณ T โดยแทนค่า $t = kT$ ดังนั้นสมการที่ (2.33) จะมีค่าเป็น

$$\int_0^{kT} (dy(t)/dt) \cdot dt = -a \int_0^{kT} y(t) \cdot dt + a \int_0^{kT} x(t) \cdot dt \quad (2.34)$$

หรือ
$$y(kT) - y(0) = -a \int_0^{kT} y(t) \cdot dt + a \int_0^{kT} x(t) \cdot dt \quad (2.35)$$

และเมื่อเปลี่ยน kT เป็น $(k-1)T$ สมการที่ (2.34) มีค่าเป็น

$$y((k-1)T) - y(0) = -a \int_0^{(k-1)T} y(t) \cdot dt + a \int_0^{(k-1)T} x(t) \cdot dt \quad (2.36)$$

ลบสมการที่ (2.35) โดยสมการที่ (2.36) จะได้

$$y(kT) = y((k-1)T) - a \int_{(k-1)T}^{kT} y(t) \cdot dt + a \int_{(k-1)T}^{kT} x(t) \cdot dt \quad (2.37)$$

โดยวิธีไบลีเนียร์ทรานส์ฟอร์มเมชัน

$$\int_{(k-1)T}^{kT} y(t) \cdot dt = [y(kT) + y((k-1)T)]T/2 \quad (2.38)$$

และ

$$\int_{(k-1)T}^{kT} x(t) \cdot dt = [x(kT) + x((k-1)T)]T/2 \quad (2.39)$$

ดังนั้นสมการที่ (2.37) สามารถเขียนเป็น

$$y(kT) = y((k-1)T) - [y(kT) + y((k-1)T)]aT/2 + [x(k) + x((k-1)T)]aT/2 \quad (2.40)$$

โดยใช้ Z transform กับสมการที่ (2.40)

$$Y(z) = z^{-1}Y(z) - [Y(z) + z^{-1}Y(z)]aT/2 + [X(z) + z^{-1}X(z)]aT/2$$

$$\frac{Y(z)}{X(z)} = \frac{(1+z^{-1})aT/2}{(1-z^{-1}) + (1+z^{-1})aT/2} = \frac{a}{2(1-z^{-1})/T(1+z^{-1}) + a} \quad (2.41)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเปรียบเทียบสมการที่ (2.31) และ (2.41) จะได้

$$s = 2(1 - z^{-1})/T(1 + z^{-1}) \quad (2.42)$$

การแปลงโดยวิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน จะทำให้ได้จำนวนโพลและซีโรของระบบ ไม่ต่อเนื่องเท่ากัน นอกจากนี้อันดับของระบบไม่ต่อเนื่องจะเท่ากับอันดับของระบบต่อเนื่องเดิมด้วย เมื่อแทนค่า s ตามสมการที่ (2.42) จะทำให้ทางด้านซ้ายของ S -plane ($\text{Re}(S) < 0$) เป็น

$$\text{Re}[2(1-z^{-1})/T(1+z^{-1})] = \text{Re}[2(z-1)/T(z+1)] < 0 \quad (2.43)$$

เนื่องจาก $T > 0$ ดังนั้น

$$\text{Re}[(z-1)/(z+1)] < 0 \quad (2.44)$$

ถ้ากำหนดให้ $z = r + jw$ ดังนั้นสมการที่ (2.44) คือ

$$\begin{aligned} \text{Re}[(z-1)/(z+1)] &= \text{Re}[(r+jw-1)/(r+jw+1)] \\ &= \text{Re}\{[(r+jw-1)(r-jw+1)/(r+jw+1)(r-jw+1)]\} \\ &= \text{Re}[(r^2-1+w^2+j2w)/((r+1)^2+w^2)] < 0 \quad (2.45) \end{aligned}$$

$$\text{ซึ่งหมายความว่า} \quad r^2 - 1 + w^2 < 0$$

$$\text{หรือ} \quad r^2 + w^2 < 1 \quad (2.46)$$

นั่นคือ การแปลงโดยวิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน จะทำให้ทางซ้ายของ S -plane ถูกย้ายเข้าไปในวงกลมรัศมี 1 จุดศูนย์กลางอยู่ที่จุดกำเนิดใน Z -plane ซึ่งเป็นช่วงเสถียรภาพของ Z -plane นั่นหมายความว่าถ้าระบบต่อเนื่องเดิมมีเสถียรภาพ จะทำให้ระบบไม่ต่อเนื่องที่ได้มีเสถียรภาพด้วย

โดยแทนสมการที่ (2.42) ลงในสมการที่ (2.1) จะได้

$$\frac{y_p(z)}{u_p(z)} = \frac{z^3 L_0 + z^2 L_1 + z L_2 + L_3}{z^3 + z^2 R_1 + z R_2 + R_3} \quad (2.47)$$

$$\left. \begin{aligned} \text{โดยที่ } R_0 &= 8 + 4a_3 T + 2a_2 T^2 \\ R_1 &= (-24 - 4a_3 T + 2a_2 T^2)/R_0 \\ R_2 &= (24 - 4a_3 T - 2a_2 T^2)/R_0 \\ R_3 &= (-8 + 4a_3 T - 2a_2 T^2)/R_0 \\ L_0 &= L_3 = (a_4 T^3)/R_0 \\ L_1 &= L_2 = (3a_4 T^3)/R_0 \end{aligned} \right\} \quad (2.48)$$

และแทนสมการที่ (2.42) ลงในสมการที่ (2.2) จะได้

$$\frac{y_p(z)}{d_p(z)} = \frac{z^3 F_0 + z^2 F_1 + z F_2 + F_3}{z^3 + z^2 R_1 + z R_2 + R_3} \quad (2.49)$$

$$\left. \begin{aligned} \text{โดยที่ } F_0 &= (2a_5 T^2 + a_6 T^3)/R_0 \\ F_1 &= (2a_5 T^2 + 3a_6 T^3)/R_0 \\ F_2 &= (-2a_5 T^2 + 3a_6 T^3)/R_0 \\ F_3 &= (-2a_5 T^2 + a_6 T^3)/R_0 \end{aligned} \right\} \quad (2.50)$$

จากสมการที่ (2.47) และ (2.49) นำมาเขียนเป็นสมการสถานะเป็น

$$\begin{bmatrix} x_{p1}(k+1) \\ x_{p2}(k+1) \\ x_{p3}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -R_3 & -R_2 & -R_1 \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \\ x_{p3}(k) \end{bmatrix} + \begin{bmatrix} h_1 & g_1 \\ h_2 & g_2 \\ h_3 & g_3 \end{bmatrix} \begin{bmatrix} u_p(k) \\ d_p(k) \end{bmatrix} \quad (2.51)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$y_p(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1}(k) \\ x_{p2}(k) \\ x_{p3}(k) \end{bmatrix} + \begin{bmatrix} h_o & g_o \end{bmatrix} \begin{bmatrix} u_p(k) \\ d_p(k) \end{bmatrix} \quad (2.52)$$

$$\begin{aligned} \text{โดยที่ } h_o &= L_o \\ h_1 &= L_1 - R_1 h_o \\ h_2 &= L_2 - R_1 h_1 - R_2 h_o \\ h_3 &= L_3 - R_1 h_2 - R_2 h_1 - R_3 h_o \\ g_o &= F_o \\ g_1 &= F_1 - R_1 g_o \\ g_2 &= F_2 - R_1 g_1 - R_2 g_o \\ g_3 &= F_3 - R_1 g_2 - R_2 g_1 - R_3 g_o \end{aligned} \quad (2.53)$$

หรือสมการที่ (2.51) และ (2.52) อาจเขียนเป็น

$$X_p(k+1) = A_p X_p(k) + B_p U_p(k) \quad (2.54)$$

$$Y_p(k) = C_p X_p(k) + D_p U_p(k) \quad (2.55)$$

และในทำนองเดียวกัน แพนสมการที่ (2.10) จะได้โมเดลอ้างอิงชนิดไม่ต่อเนื่อง

$$\frac{y_m(z)}{u_m(z)} = \frac{z^3 B_o + z^2 B_1 + z B_2 + B_3}{z^3 + z^2 A_1 + z A_2 + A_3} \quad (2.56)$$

$$\begin{aligned} \text{โดยที่ } A_o &= 8 + 4m_3 T + 2m_2 T^2 + m_1 T^3 \\ A_1 &= (-24 - 4m_3 T + 2m_2 T^2 + 3m_1 T^3)/A_o \\ A_2 &= (24 - 4m_3 T - 2m_2 T^2 + 3m_1 T^3)/A_o \\ A_3 &= (-8 + 4m_3 T - 2m_2 T^2 + m_1 T^3)/A_o \\ B_o &= B_3 = m_1 T^3/A_o \\ B_1 &= B_2 = 3m_1 T^3/A_o \end{aligned} \quad (2.57)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (2.56) นำมาเขียนเป็นสมการสถานะได้เป็น

$$\begin{bmatrix} x_{m1}(k+1) \\ x_{m2}(k+1) \\ x_{m3}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -A_3 & -A_2 & -A_1 \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ x_{m3}(k) \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \begin{bmatrix} u_m(k) \end{bmatrix} \quad (2.58)$$

$$y_m(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{m1}(k) \\ x_{m2}(k) \\ x_{m3}(k) \end{bmatrix} + \begin{bmatrix} c_o \end{bmatrix} \begin{bmatrix} u_m(k) \end{bmatrix} \quad (2.59)$$

โดยที่ $c_o = B_o$
 $c_1 = B_1 - A_1 c_o$
 $c_2 = B_2 - A_1 c_1 - A_2 c_o$
 $c_3 = B_3 - A_1 c_2 - A_2 c_1 - A_3 c_o$ (2.60)

หรือสมการที่ (2.58) และ (2.59) อาจเขียนเป็น

$$X_m(k+1) = A_m X_m(k) + B_m U_m(k) \quad (2.61)$$

$$Y_m(k) = C_m X_m(k) + D_m U_m(k) \quad (2.62)$$

2.3.2. ทฤษฎีของ MRAC ชนิดไม่ต่อเนื่อง [2,14-18]

สมการของไลฟนอวสำหรับระบบไม่ต่อเนื่อง คือ

$$V(e(k)) = e(k)^T P e(k) \quad (2.63)$$

เมื่อ $e(k)$ คือ เวกเตอร์ผิดพลาด และ

$$\mathbf{e}(k) = \mathbf{X}_m(k) - \mathbf{X}_p(k) = \begin{bmatrix} x_{m1}(k) - x_{p1}(k) \\ x_{m2}(k) - x_{p2}(k) \\ x_{m3}(k) - x_{p3}(k) \end{bmatrix} = \begin{bmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \end{bmatrix} \quad (2.64)$$

สำหรับระบบไม่ต่อเนื่องเราสามารถหาค่า $\Delta V(\mathbf{e}(k))$ โดยใช้ฟอเวิร์ดดิฟเฟอเรนซ์ (forward difference)

$$\begin{aligned} \Delta V(\mathbf{e}(k)) &= V(\mathbf{e}(k+1)) - V(\mathbf{e}(k)) \\ &= [\mathbf{e}(k+1)]^T \mathbf{P} \mathbf{e}(k+1) - \mathbf{e}(k)^T \mathbf{P} \mathbf{e}(k) \end{aligned} \quad (2.65)$$

จากสมการที่ (2.54), (2.61) และ (2.65) จะได้

$$\begin{aligned} \Delta V(\mathbf{e}(k)) &= [\mathbf{A}_m \mathbf{X}_m(k) + \mathbf{B}_m \mathbf{U}_m(k) - \mathbf{A}_p \mathbf{X}_p(k) - \mathbf{B}_p \mathbf{U}_p(k)]^T \mathbf{P} [\mathbf{A}_m \mathbf{X}_m(k) \\ &\quad + \mathbf{B}_m \mathbf{U}_m(k) - \mathbf{A}_p \mathbf{X}_p(k) - \mathbf{B}_p \mathbf{U}_p(k)] - \mathbf{e}(k)^T \mathbf{P} \mathbf{e}(k) \\ &= \mathbf{e}(k)^T [\mathbf{A}_p^T \mathbf{P} \mathbf{A}_p - \mathbf{P}] \mathbf{e}(k) + \mathbf{e}(k)^T \mathbf{A}_p^T \mathbf{P} \mathbf{M}_2 + \mathbf{M}_2^T \mathbf{P} \mathbf{A}_p \mathbf{e}(k) \\ &\quad + \mathbf{M}_2^T \mathbf{P} \mathbf{M}_2 \end{aligned} \quad (2.66)$$

$$\text{โดยที่} \quad \mathbf{M}_2 = \mathbf{A}_m \mathbf{X}_m(k) + \mathbf{B}_m \mathbf{U}_m(k) - \mathbf{A}_p \mathbf{X}_m(k) - \mathbf{B}_p \mathbf{U}_p(k) \quad (2.67)$$

$$\text{จะได้} \quad \Delta V(\mathbf{e}(k)) = \mathbf{e}(k)^T [-\mathbf{Q}] \mathbf{e}(k) + \mathbf{M} \quad (2.68)$$

$$\text{โดยที่} \quad \mathbf{Q} = \mathbf{A}_p^T \mathbf{P} \mathbf{A}_p - \mathbf{P} \quad (2.69)$$

$$\mathbf{M} = 2\mathbf{e}(k)^T \mathbf{A}_p^T \mathbf{P} \mathbf{M}_2 + \mathbf{M}_2^T \mathbf{P} \mathbf{M}_2 \quad (2.70)$$

การพิจารณาจะกระทำแบบเดียวกับ MRAC ชนิดต่อเนื่อง สมการใดๆ จะเป็นสมการไลทูนอวชนิดไม่ต่อเนื่อง ถ้าเงื่อนไขทั้ง 2 ข้อเป็นจริง
เงื่อนไขที่ 1. สมการนั้นมีความหมายเป็นบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขที่ 2. ค่าพหุวัตติเฟอเรนซ์ของสมการนั้นมีความหมายเป็นลบ

ในสมการที่ (2.63) กำหนดให้ P เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงซึ่งมีความหมายเป็นบวกเช่นเดียวกับในชนิดต่อเนื่อง ดังนั้นเงื่อนไขที่ 1. เป็นจริง จากสมการที่ (2.69) และ (2.70) เลือก Q มีความหมายเป็นบวก และเลือก M มีความหมายกึ่งเป็นลบ จะทำให้สมการ (2.68) มีความเป็นลบ ดังนั้นเงื่อนไขที่ 2. เป็นจริง เช่นเดียวกับ MRAC ชนิดต่อเนื่อง เพื่อความสะดวกจะกำหนด Q ให้มีความหมายเป็นบวกก่อน แล้วจึงหาค่า P ตามสมการที่ (2.69) และพิสูจน์ว่ามีความหมายเป็นบวกหรือไม่ การทำให้ M มีความหมายกึ่งเป็นลบ ทำได้โดยการเลือกค่า $u_p(k)$ ที่เหมาะสม จากสมการที่ (2.67) และ (2.70)

$$\begin{aligned}
 M &= t_1(c_4+2e_2(k)) + t_2(c_5+2e_3(k)) + t_3(x_5-2e_4(k)) \\
 &\quad - u_p(k)[2(e_2(k)h_4 + e_3(k)h_5 - e_4(k)h_6 + h_1t_1 + h_2t_2 \\
 &\quad + h_3t_3)] - [u_p(k)]^2(h_1h_4 + h_2h_5 + h_3h_6) \\
 &= a_0 - a_1u_p(k) + a_2[u_p(k)]^2
 \end{aligned} \tag{2.71}$$

$$\begin{aligned}
 \text{โดยที่ } c_4 &= c_1u_m(k) - g_1d_p(k) \\
 c_5 &= c_2u_m(k) - g_2d_p(k) \\
 c_6 &= c_3u_m(k) - g_3d_p(k) \\
 x_4 &= (-A_3+R_3)x_{m1}(k) + (-A_2+R_2)x_{m2}(k) + (-A_1+R_1)x_{m3}(k) \\
 x_5 &= x_4 + c_6 \\
 e_4(k) &= R_3e_1(k) + R_2e_2(k) + R_1e_3(k) \\
 h_4 &= p_1h_1 + p_2h_2 + p_3h_3 \\
 h_5 &= p_2h_1 + p_4h_2 + p_5h_3 \\
 h_6 &= p_3h_1 + p_5h_2 + p_6h_3 \\
 t_1 &= p_1c_4 + p_2c_5 + p_3x_5 \\
 t_2 &= p_2c_4 + p_4c_5 + p_5x_5 \\
 t_3 &= p_3c_4 + p_5c_5 + p_6x_5 \\
 a_0 &= t_1(c_4+2e_2(k)) + t_2(c_5+2e_3(k)) + t_3(x_5-2e_4(k))
 \end{aligned} \tag{2.72}$$

$$\left. \begin{aligned} a_1 &= 2(e_2(k)h_4 + e_3(k)h_5 - e_4(k)h_6 + h_1t_1 + h_2t_2 + h_3t_3) \\ a_2 &= h_1h_4 + h_2h_5 + h_3h_6 \end{aligned} \right\}$$

จากสมการที่ (2.71) กำหนดให้ $M = 0$

$$0 = a_0 - a_1u_p(k) + a_2[u_p(k)]^2 \quad (2.73)$$

$$\text{ดังนั้น } u_p(k) = (a_1 + \sqrt{(a_1)^2 - 4a_0a_2})/2a_2 \quad (2.74)$$

สัญญาณควบคุม $u_p(k)$ ที่ได้จะถูกนำมาใช้ควบคุมระบบควบคุมตำแหน่ง เพื่อให้มีผลตอบสนองตามต้องการ

2.3.3. การใช้ตัวสังเกตสถานะสร้างตัวแปรสถานะของระบบควบคุมตำแหน่ง

เมื่อนำ MRAC มาเพื่อใช้ควบคุมตำแหน่งมอเตอร์กระแสตรงจริงๆ สิ่งที่ได้คือ จำนวนองศาที่เกินของเฟืองทดรอบหมุนไป ($y_p(k)$) แต่สิ่งที่ต้องใช้ในการคำนวณคือตัวแปรสถานะ (state variable, $X_p(k)$) ซึ่งไม่สามารถวัดได้โดยตรงจากระบบควบคุมตำแหน่ง ดังนั้นจึงจำเป็นต้องใช้ตัวสังเกตสถานะ (state observer) [2] สร้างตัวแปรสถานะขึ้นมาจากจำนวนองศาที่หมุนไป และสัญญาณควบคุม ตัวแปรสถานะจะถูกสร้างขึ้นได้ถ้าระบบควบคุมตำแหน่งเป็นไปตามเงื่อนไขสามารถสังเกตได้ (observability condition) [ภาคผนวก ง.] ระบบใดๆ จะสามารถสังเกตได้ ถ้าเรกซ์ (rank) ของเมตริกซ์ตามสมการที่ (2.75) มีค่าเท่ากับ n เมื่อ n คือ อันดับของระบบนั้น

$$\begin{bmatrix} C_p^T & ; A_p^T C_p^T & ; (A_p^T)^2 C_p^T & ; \dots & ; (A_p^T)^{n-1} C_p^T \end{bmatrix} \quad (2.75)$$

โดยอ้างอิงถึงระบบควบคุมตำแหน่งตามสมการที่ (2.51), (2.52), (2.54) และ (2.55) เมตริกซ์ตามสมการที่ (2.75) จะมีค่าเป็น

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.76)$$

และดีเทอร์มิแนนต์ (determinant) ของเมตริกซ์ในสมการที่ (2.76) จะมีค่าไม่เท่ากับ 0 ดังนั้นจึงเป็นนอหิงกูลาร์ (nonsingular) และมีแรงค์เท่ากับ 3 เพราะฉะนั้นระบบควบคุมตำแหน่งนี้สามารถสังเกตได้ จึงสามารถสร้างตัวแปรสถานะจาก $Y_p(k)$ และ $u_p(k)$ ได้ โดยอ้างอิงสมการที่ (2.54) และ (2.55) อีกครั้งหนึ่ง

$$X_p(k+1) = A_p X_p(k) + B_p U_p(k) \quad (2.77)$$

$$Y_p(k) = C_p X_p(k) + D_p U_p(k) \quad (2.78)$$

คูณสมการที่ (2.77) ด้วย A_p^{-1}

$$A_p^{-1} X_p(k+1) = X_p(k) + A_p^{-1} B_p U_p(k)$$

$$\text{หรือ } X_p(k) = A_p^{-1} X_p(k+1) - A_p^{-1} B_p U_p(k) \quad (2.79)$$

เลื่อนค่า k ไป 1 ตำแหน่ง

$$X_p(k-1) = A_p^{-1} X_p(k) - A_p^{-1} B_p U_p(k-1) \quad (2.80)$$

แทนสมการที่ (2.79) ลงใน (2.80)

$$\begin{aligned} X_p(k-1) &= A_p^{-1} [A_p^{-1} X_p(k+1) - A_p^{-1} B_p U_p(k)] \\ &\quad - A_p^{-1} B_p U_p(k-1) \\ &= A_p^{-2} X_p(k+1) - A_p^{-2} B_p U_p(k) - A_p^{-1} B_p U_p(k-1) \end{aligned} \quad (2.81)$$

และในทำนองเดียวกัน

$$\begin{aligned}
 X_p(k-2) &= A_p^{-3} X_p(k+1) - A_p^{-3} B_p U_p(k) - A_p^{-2} B_p U_p(k-1) \\
 &\quad - A_p^{-1} B_p U_p(k-2) \\
 &\quad \vdots \\
 &\quad \vdots \\
 X_p(k-n+1) &= A_p^{-n} X_p(k+1) - A_p^{-n} B_p U_p(k) - A_p^{-n+1} B_p U_p(k-1) \\
 &\quad - \dots - A_p^{-1} B_p U_p(k-n+1) \quad (2.82)
 \end{aligned}$$

โดยแทนสมการที่ (2.79) ลงใน (2.78)

$$Y_p(k) = C_p A_p^{-1} X_p(k+1) - C_p A_p^{-1} B_p U_p(k) + D_p U_p(k) \quad (2.83)$$

และในทำนองเดียวกัน

$$\begin{aligned}
 Y_p(k-1) &= C_p A_p^{-2} X_p(k+1) - C_p A_p^{-2} B_p U_p(k) \\
 &\quad - C_p A_p^{-1} B_p U_p(k-1) + D_p U_p(k-1) \\
 &\quad \vdots \\
 &\quad \vdots \\
 Y_p(k-n+1) &= C_p A_p^{-n} X_p(k+1) - C_p A_p^{-n} B_p U_p(k) \\
 &\quad - C_p A_p^{-n+1} B_p U_p(k-1) - \dots - C_p A_p^{-1} B_p U_p(k-n+1) \\
 &\quad + D_p U_p(k-n+1) \quad (2.84)
 \end{aligned}$$

รวมสมการที่ (2.83) และ (2.84) เข้าเป็นเมตริกซ์จะได้

$$\begin{bmatrix} Y_p(k) \\ Y_p(k-1) \\ \vdots \\ Y_p(k-n+1) \end{bmatrix} = \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ \vdots \\ C_p A_p^{-n} \end{bmatrix} X_p(k+1) - \begin{bmatrix} C_p A_p^{-1} B_p & 0 & \vdots & \dots & \vdots & 0 \\ C_p A_p^{-2} B_p & C_p A_p^{-1} B_p & \vdots & \dots & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_p A_p^{-n} B_p & C_p A_p^{-n+1} B_p & \dots & C_p A_p^{-1} B_p \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} U_p(k) \\ U_p(k-1) \\ \vdots \\ U_p(k-n+1) \end{bmatrix} + \begin{bmatrix} D_p & 0 & \cdots & 0 \\ 0 & D_p & \cdots & \\ & & \ddots & \\ 0 & 0 & \cdots & D_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ \vdots \\ U_p(k-n+1) \end{bmatrix} \quad (2.85)$$

หรือ

$$\begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ \vdots \\ C_p A_p^{-n} \end{bmatrix} X_p(k+1) = \begin{bmatrix} Y_p(k) \\ Y_p(k-1) \\ \vdots \\ Y_p(k-n+1) \end{bmatrix} + \begin{bmatrix} C_p A_p^{-1} B_p & 0 & \cdots & 0 \\ C_p A_p^{-2} B_p & C_p A_p^{-1} B_p & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C_p A_p^{-n} B_p & C_p A_p^{-n+1} B_p & \cdots & C_p A_p^{-1} B_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ \vdots \\ U_p(k-n+1) \end{bmatrix} \quad (2.86)$$

จะสังเกตเห็นได้ว่าทางด้านขวาของสมการที่ (2.86) จะทราบค่าทั้งหมด เพราะได้จากการวัดโดยตรงจากระบบควบคุมตำแหน่ง และระบบควบคุมมีอันดับ 3 ดังนั้นจะสามารถหาค่าของ $X_p(k+1)$ ได้ถ้าและเท่านั้นถ้า

$$\text{rank} \begin{bmatrix} C_p \\ C_p A_p^{-1} \\ C_p A_p^{-2} \end{bmatrix} = 3 \quad (2.87)$$

ซึ่งได้พิสูจน์มาแล้วในสมการที่ (2.76)

โดยคุณทั้งสองด้านของสมการที่ (2.86) ด้วยค่าอินเวอร์สเมทริกซ์ของสมการที่ (2.87) ดังนั้นสมการที่ (2.86) จะถูกตัดแปลงเป็น

$$\begin{aligned}
 X_p(k+1) &= \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \begin{bmatrix} Y_p(k) \\ Y_p(k-1) \\ Y_p(k-2) \end{bmatrix} + \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \\
 &\quad \begin{bmatrix} C_p A_p^{-1} B_p & ; & 0 & ; & 0 \\ C_p A_p^{-2} B_p & ; & C_p A_p^{-1} B_p & ; & 0 \\ C_p A_p^{-3} B_p & ; & C_p A_p^{-2} B_p & ; & C_p A_p^{-1} B_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ U_p(k-2) \end{bmatrix} \\
 &\quad - \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \begin{bmatrix} D_p & 0 & 0 \\ 0 & D_p & 0 \\ 0 & 0 & D_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ U_p(k-2) \end{bmatrix} \quad (2.88)
 \end{aligned}$$

หรือ

$$\begin{aligned}
 \begin{bmatrix} X_{p1}(k+1) \\ X_{p2}(k+1) \\ X_{p3}(k+1) \end{bmatrix} &= \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \begin{bmatrix} Y_p(k) \\ Y_p(k-1) \\ Y_p(k-2) \end{bmatrix} + \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \\
 &\quad \begin{bmatrix} C_p A_p^{-1} B_p & ; & 0 & ; & 0 \\ C_p A_p^{-2} B_p & ; & C_p A_p^{-1} B_p & ; & 0 \\ C_p A_p^{-3} B_p & ; & C_p A_p^{-2} B_p & ; & C_p A_p^{-1} B_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ U_p(k-2) \end{bmatrix} \\
 &\quad - \begin{bmatrix} C_p A_p^{-1} \\ C_p A_p^{-2} \\ C_p A_p^{-3} \end{bmatrix}^{-1} \begin{bmatrix} D_p & 0 & 0 \\ 0 & D_p & 0 \\ 0 & 0 & D_p \end{bmatrix} \begin{bmatrix} U_p(k) \\ U_p(k-1) \\ U_p(k-2) \end{bmatrix} \quad (2.89)
 \end{aligned}$$

จากสมการที่ (2.89) จะเห็นว่าสามารถสร้างตัวแปรสถานะของระบบควบคุมตำแหน่ง $X_p(k+1)$

ได้จาก $Y_p(k-2), \dots, Y_p(k)$ และ $U_p(k-2), \dots, U_p(k)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4. สรุป

ในบทที่ 2 ได้กล่าวถึงการนำทฤษฎีควบคุมสมัยใหม่มาควบคุมตำแหน่งมอเตอร์กระแสตรง จนได้ตัวควบคุมอแดปทีฟแบบ MRAC การสร้างอแดปทีฟคอนโทรลอัลกอริทึมของ MRAC ทั้งชนิดต่อเนื่องและไม่ต่อเนื่อง โดยวิธีที่ 2 ของไลฟนูอว การแปลงระบบต่อเนื่องเป็นระบบไม่ต่อเนื่องโดยวิธีไบลิเนียร์ทรานส์ฟอร์มเมชัน นอกจากนี้ยังได้กล่าวถึงการสร้างตัวแปรสถานะของระบบควบคุมตำแหน่งชนิดไม่ต่อเนื่องจากสัญญาณควบคุมและเอาต์พุทของระบบควบคุมตำแหน่ง



บทที่ 3.

วงจรรีเล็กทรอนิกส์กำลังและวงจรควบคุมตำแหน่ง

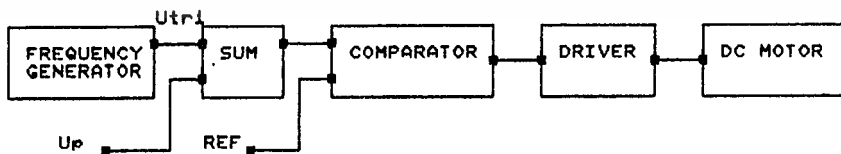
3.1. บทนำ

โดยทั่วไปวงจรรีเล็กทรอนิกส์สำหรับขับเคลื่อนมอเตอร์กระแสตรงมี 2 ชนิด คือ วงจรขับชนิดลีนีเยร์ และวงจรขับชนิด PWM คอนเวอร์เตอร์ (Pulse Width Modulation Converter) วงจรขับชนิดลีนีเยร์จะบังคับให้ทรานส์ซิสเตอร์ทำงานในช่วงแควคัพ มีข้อดีคือสร้างง่ายและไม่ทำให้เกิดเสียงฮัม แต่ข้อเสียของวงจรชนิดนี้คือ จะเกิดกำลังสูญเสียบนตัวทรานส์ซิสเตอร์สูง ทำให้ต้องติดตั้งระบบระบายความร้อนขนาดใหญ่ เพื่อป้องกันทรานส์ซิสเตอร์เสียหายเนื่องจากร้อนเกินไป เพื่อหลีกเลี่ยงการสูญเสียบนตัวทรานส์ซิสเตอร์จึงเลือกใช้วงจรรีเล็กทรอนิกส์ PWM คอนเวอร์เตอร์ ซึ่งทรานส์ซิสเตอร์ถูกบังคับให้ทำงานอยู่ใน 2 ช่วง คือ ช่วงอิ่มตัว และช่วงคัทออฟเท่านั้น ในขณะที่ทรานส์ซิสเตอร์ทำงานในช่วงอิ่มตัวจะมีโวลต์ตกคร่อมประมาณ 1 โวลต์หรือมากกว่าเล็กน้อย ทำให้เกิดกำลังสูญเสียเพียงเล็กน้อยเท่านั้น และในขณะที่ทรานส์ซิสเตอร์ทำงานในช่วงคัทออฟจะมีกระแสเพียงเล็กน้อยที่ไหลผ่านทรานส์ซิสเตอร์ ทำให้เกิดกำลังสูญเสียเล็กน้อยเช่นเดียวกัน

แม้ว่าวงจรรีเล็กทรอนิกส์ PWM คอนเวอร์เตอร์จะมีความยุ่งยากในการสร้างมากกว่า แต่ข้อดีเรื่องกำลังสูญเสียต่ำ จึงเลือกใช้วงจรรีเล็กทรอนิกส์

3.2. โครงสร้างของ PWM คอนเวอร์เตอร์

โครงสร้างของ PWM คอนเวอร์เตอร์ประกอบด้วย วงจรสร้างความถี่ วงจรเปรียบเทียบ และวงจรขับ [19-22] ดังรูปที่ 3.1.

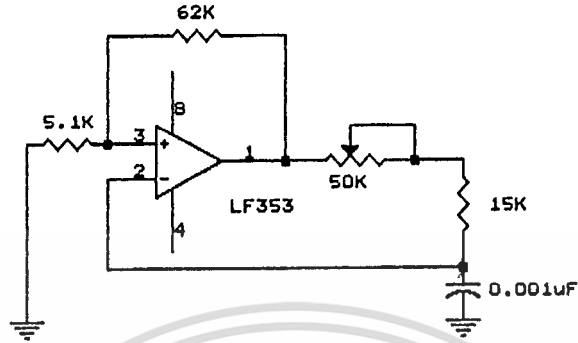


รูปที่ 3.1 แสดงโครงสร้างของ PWM คอนเวอร์เตอร์

3.2.1. วงจรสร้างความถี่

วงจรสร้างความถี่ทำหน้าที่สร้างสัญญาณสามเหลี่ยมเพื่อใช้ร่วมกับสัญญาณควบคุม (u_p) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

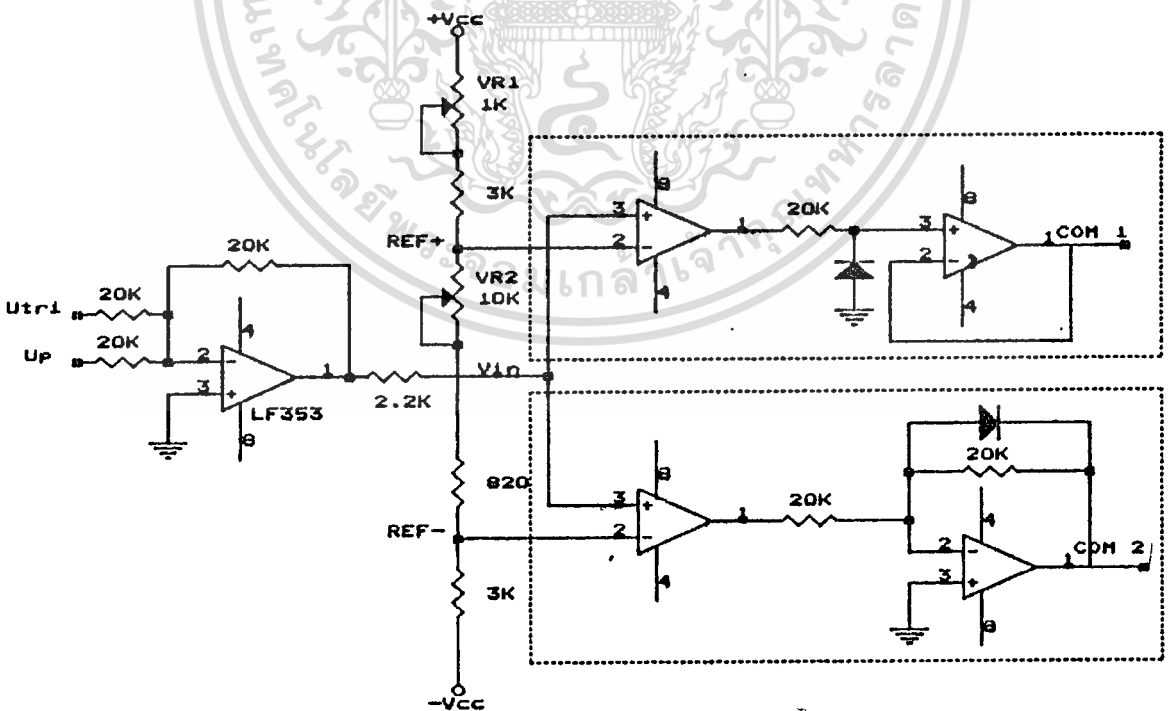
และป้อนต่อไปยังวงจรเปรียบเทียบ ความถี่ที่ได้จากวงจรนี้จะเป็นความถี่สวิทช์ของวงจรขับ ความถี่สูงนั้นจะทำให้กระแสมอเตอร์เร็วขึ้น สามารถปรับความถี่โดยปรับความต้านทานปรับค่าได้ในที่นี้เลือกใช้ความถี่ 10 กิโลเฮิร์ตซ์



รูปที่ 3.2 แสดงวงจรสร้างควมถี่

3.2.2. วงจรเปรียบเทียบ

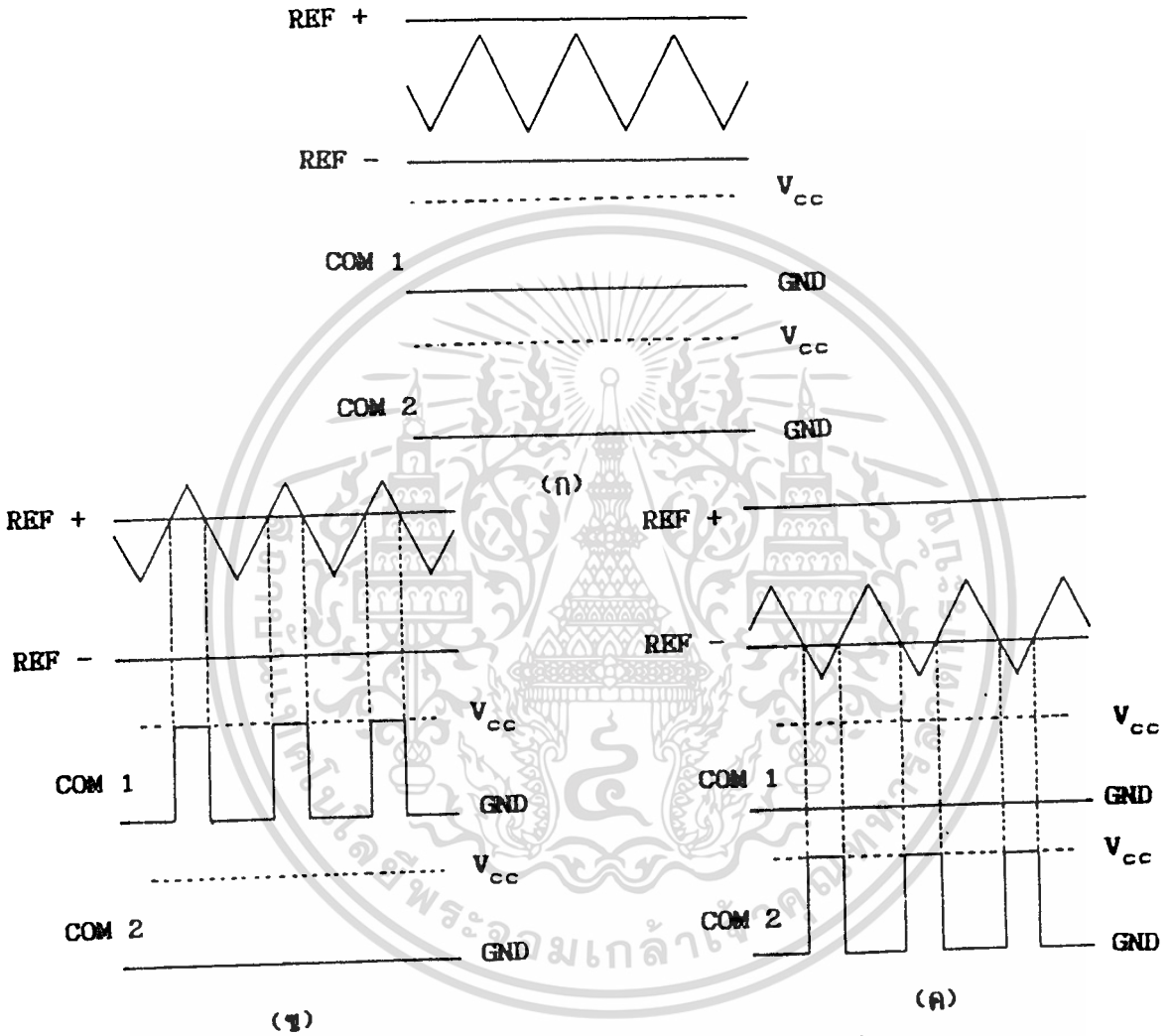
วงจรเปรียบเทียบทำหน้าที่เปรียบเทียบผลรวมของสัญญาณสามเหลี่ยมและสัญญาณควมกับสัญญาณอ้างอิง ลักษณะของวงจรแสดงดังรูปที่ 3.3 ขนาดของสัญญาณอ้างอิงบวกและลบ (REF+ และ REF-) สามารถปรับได้โดยปรับความต้านทานปรับค่าได้ VR1 และ VR2



รูปที่ 3.3 แสดงวงจรเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอ้างอิงบวกและลบจะต้องถูกปรับให้แตกต่างกันมากเพียงพอที่จะทำให้ COM 1 และ COM 2 ไม่มีค่าเป็น $+V_{cc}$ พร้อมกัน เพราะจะทำให้เกิดการช็อตแหล่งจ่ายไฟในวงจรกำลัง และสัญญาณอ้างอิงจะต้องไม่แตกต่างกันมากเกินไป เพราะจะทำให้ทรานส์ซิสเตอร์ในวงจรกำลัง นำกระแสเข้าเกินไป

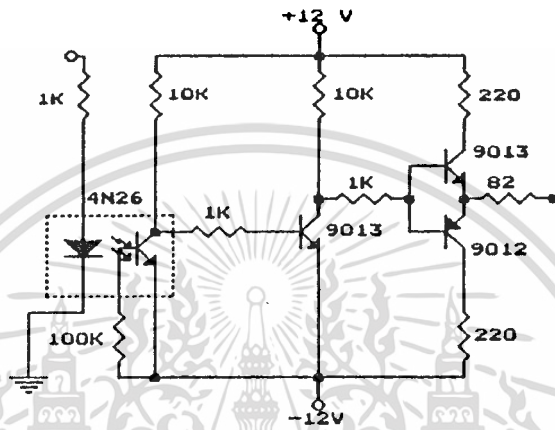


รูปที่ 3.4 แสดงเอาต์พุตของวงจรเปรียบเทียบ

- รูปที่ 3.4 ก. เมื่อผลรวมของสัญญาณควบคุมและสัญญาณสามเหลี่ยมมีค่าอยู่ระหว่างสัญญาณอ้างอิงบวกและลบ
- รูปที่ 3.4 ข. เมื่อผลรวมของสัญญาณควบคุมและสัญญาณสามเหลี่ยมบางส่วนมีค่ามากกว่าสัญญาณอ้างอิงบวก
- รูปที่ 3.4 ค. เมื่อผลรวมของสัญญาณควบคุมและสัญญาณสามเหลี่ยมบางส่วนมีค่าน้อยกว่าสัญญาณอ้างอิงลบ

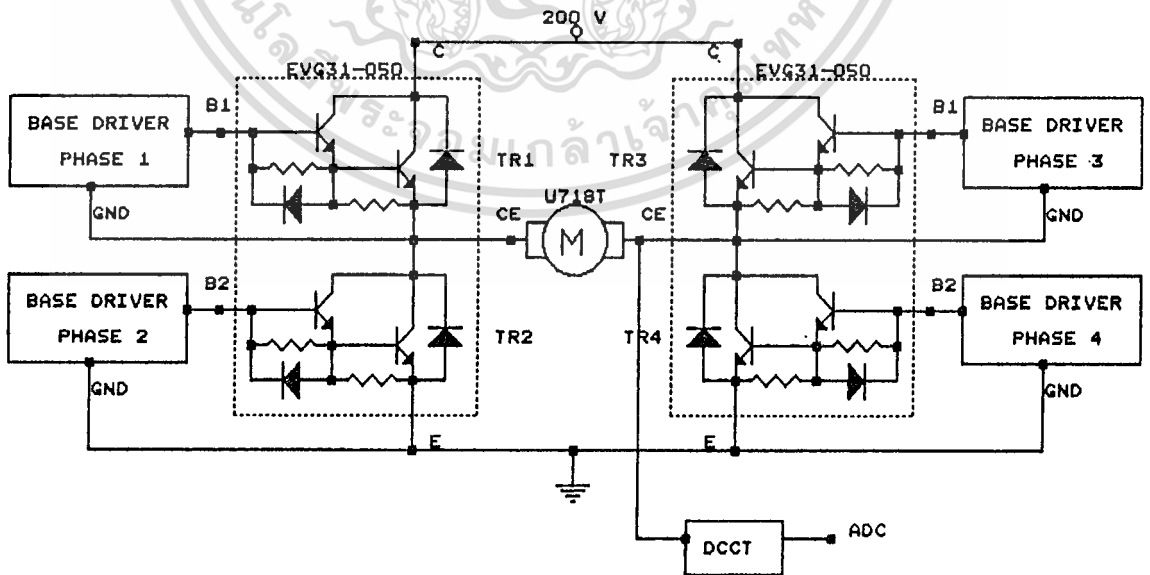
3.2.3. วงจรขับ

วงจรขับประกอบด้วย 2 ส่วนคือ วงจรขับเบส และวงจรกำลัง วงจรขับเบสทำหน้าที่ขยายสัญญาณจากวงจรเปรียบเทียบให้ใหญ่เพียงพอจะทำให้วงจรกำลังทำงานได้ วงจรขับเบสถูกแยกออกจากวงจรเปรียบเทียบโดยวิธีการเชื่อมโยงทางแสง (opto coupling) เพื่อป้องกันสัญญาณรบกวนที่เกิดขึ้นในวงจรกำลัง ไปรบกวนการทำงานของวงจรอื่นๆ วงจรขับเบสแต่ละชุดจะมีแหล่งจ่ายไฟแยกกัน



รูปที่ 3.5 แสดงวงจรขับเบส

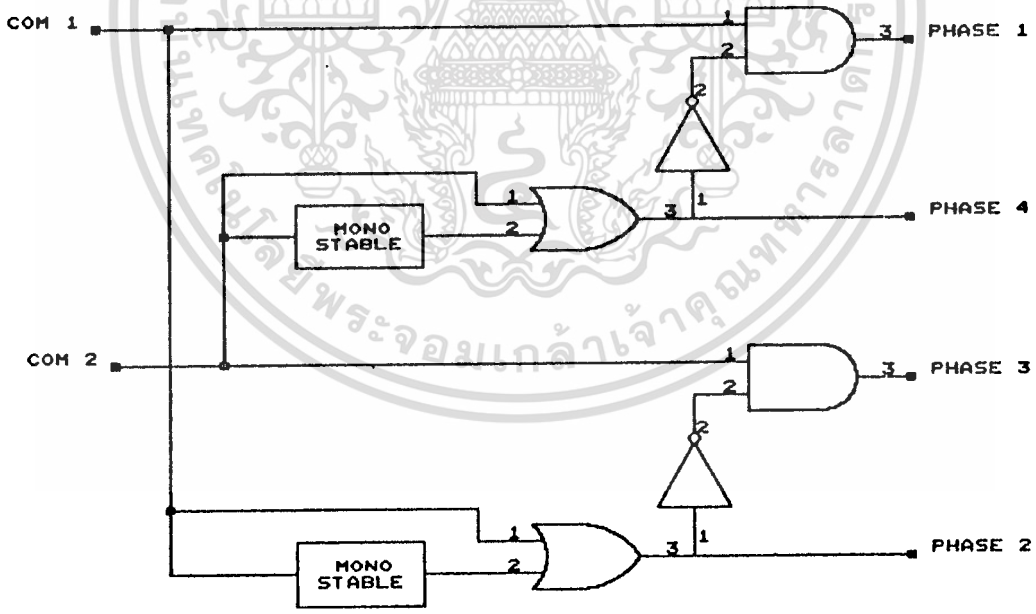
วงจรกำลังใช้เพาเวอร์โมดูลทรานซิสเตอร์ (Power module transistor) เบอร์ EVG31-050 ของฟูจิอิเล็กทิก 2 ตัวต่อกันแบบ H



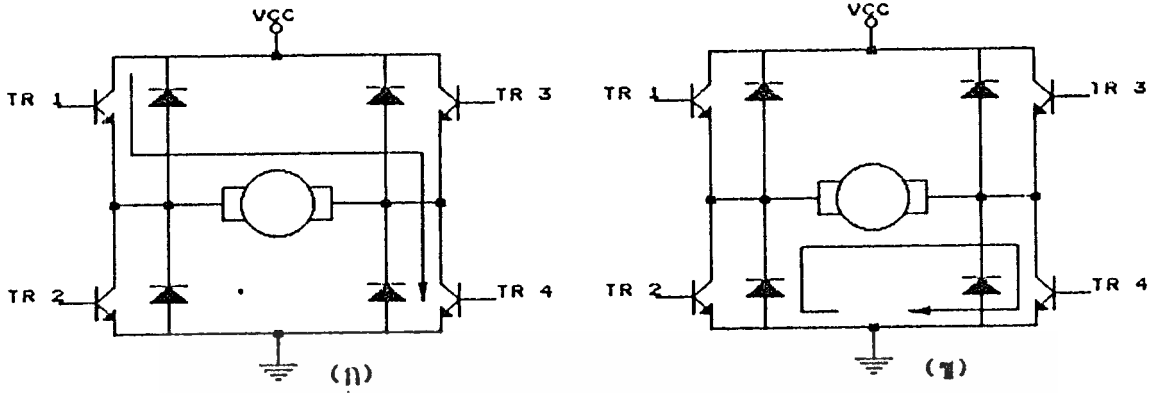
รูปที่ 3.6 แสดงการต่อวงจรกำลังและมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการให้มอเตอร์หมุนเดินหน้าจะบังคับให้ทรานส์ซิสเตอร์ชุดที่ 1 และทรานส์ซิสเตอร์ชุดที่ 4 นำกระแส และให้ทรานส์ซิสเตอร์ชุดที่ 2 และทรานส์ซิสเตอร์ชุดที่ 3 นำกระแส เมื่อต้องการให้มอเตอร์หมุนถอยหลัง นอกจากนี้ในช่วงที่ทรานส์ซิสเตอร์แต่ละชุดหยุดนำกระแสจำเป็นต้องให้ทรานส์ซิสเตอร์ชุดที่ 4 นำกระแสต่อไปเมื่อมอเตอร์หมุนเดินหน้า หรือให้ทรานส์ซิสเตอร์ชุดที่ 2 นำกระแสต่อไปเมื่อมอเตอร์หมุนถอยหลัง เพื่อใช้เป็นทางคายพลังงานที่สะสมอยู่ในขดลวดของมอเตอร์ [26] ดังนั้นสัญญาณจาก COM 1 และ COM 2 จะต้องผ่านวงจรถัดเฟสก่อนป้อนไปยังทรานส์ซิสเตอร์ในวงจรกำลัง ลักษณะของวงจรถัดเฟสแสดงดังรูปที่ 3.7 โดยวงจรมอนอสเตเบิล (monostable) ทำหน้าที่เพิ่มช่วงเวลากการนำกระแสของทรานส์ซิสเตอร์ชุดที่ 2 และทรานส์ซิสเตอร์ชุดที่ 4 วงจรลอจิกทำหน้าที่ป้องกันไม่ให้ทรานส์ซิสเตอร์ชุดที่ 1 นำกระแสในขณะที่ทรานส์ซิสเตอร์ชุดที่ 2 กำลังนำกระแส และไม่ให้ทรานส์ซิสเตอร์ชุดที่ 3 นำกระแสในขณะที่ทรานส์ซิสเตอร์ชุดที่ 4 กำลังนำกระแส



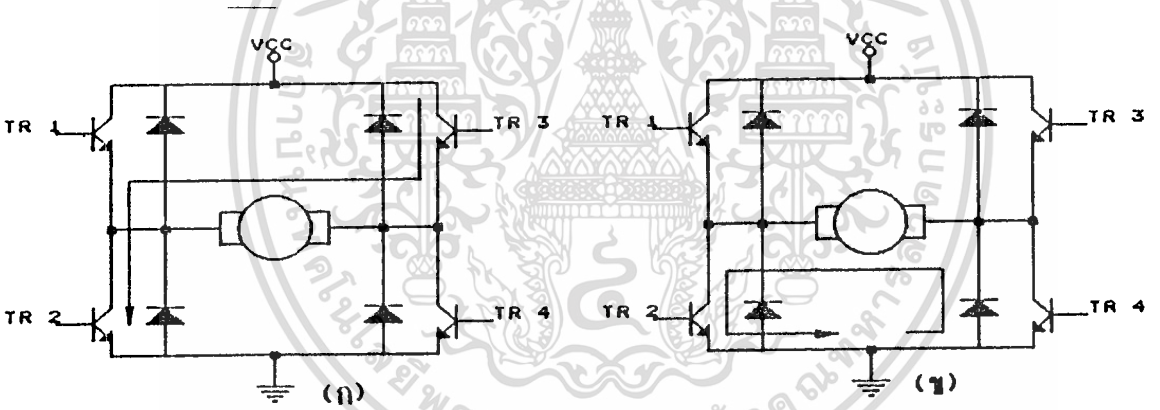
รูปที่ 3.7 แสดงวงจรถัดเฟส



รูปที่ 3.8 แสดงทิศทางของกระแสเมื่อมอเตอร์หมุนเดินหน้า

รูปที่ 3.8 ก. เมื่อทรานส์ซิสเตอร์ชุดที่ 1 และทรานส์ซิสเตอร์ชุดที่ 4 นำกระแส

รูปที่ 3.8 ข. เมื่อทรานส์ซิสเตอร์ชุดที่ 4 นำกระแส



รูปที่ 3.9 แสดงทิศทางของกระแสเมื่อมอเตอร์หมุนถอยหลัง

รูปที่ 3.9 ก. เมื่อทรานส์ซิสเตอร์ชุดที่ 2 และทรานส์ซิสเตอร์ชุดที่ 3 นำกระแส

รูปที่ 3.9 ข. เมื่อทรานส์ซิสเตอร์ชุดที่ 2 นำกระแส

3.3. การนำ MRAC มาใช้ในระบบควบคุมตำแหน่งจริง

เนื่องจากอแดปทีฟคอนโทรลอัลกอริทึมเป็นโปรแกรมไมโครคอมพิวเตอร์ ดังนั้นการจะนำมาควบคุมตำแหน่งมอเตอร์ต้องผ่านวงจรรีนาเทอร์เฟส ซึ่งประกอบด้วย 3 ส่วนคือ วงจรวัดได้เตอร์ (Decoder) วงจรเอาต์พุท และวงจรรีนาพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1. วงจรตีโค้ดเดอร์

วงจรตีโค้ดเดอร์ทำหน้าที่เลือกพอร์ตที่ต้องการใช้ติดต่อกับอุปกรณ์ภายนอก และกำหนดทิศทางของข้อมูล โดยเลือกใช้พอร์ตตำแหน่ง 2F0 ถึง 2F7 และกำหนดให้

ตำแหน่ง 2F0 เป็นเอาต์พุต (PJOUT) ใช้ส่งสัญญาณควบคุม (n_p) ไปยังวงจรกำลังผ่าน DAC

ตำแหน่ง 2F1 เป็นอินพุต (PJIN1) ใช้เป็น 8 บิตล่างของวงจรรับความถี่

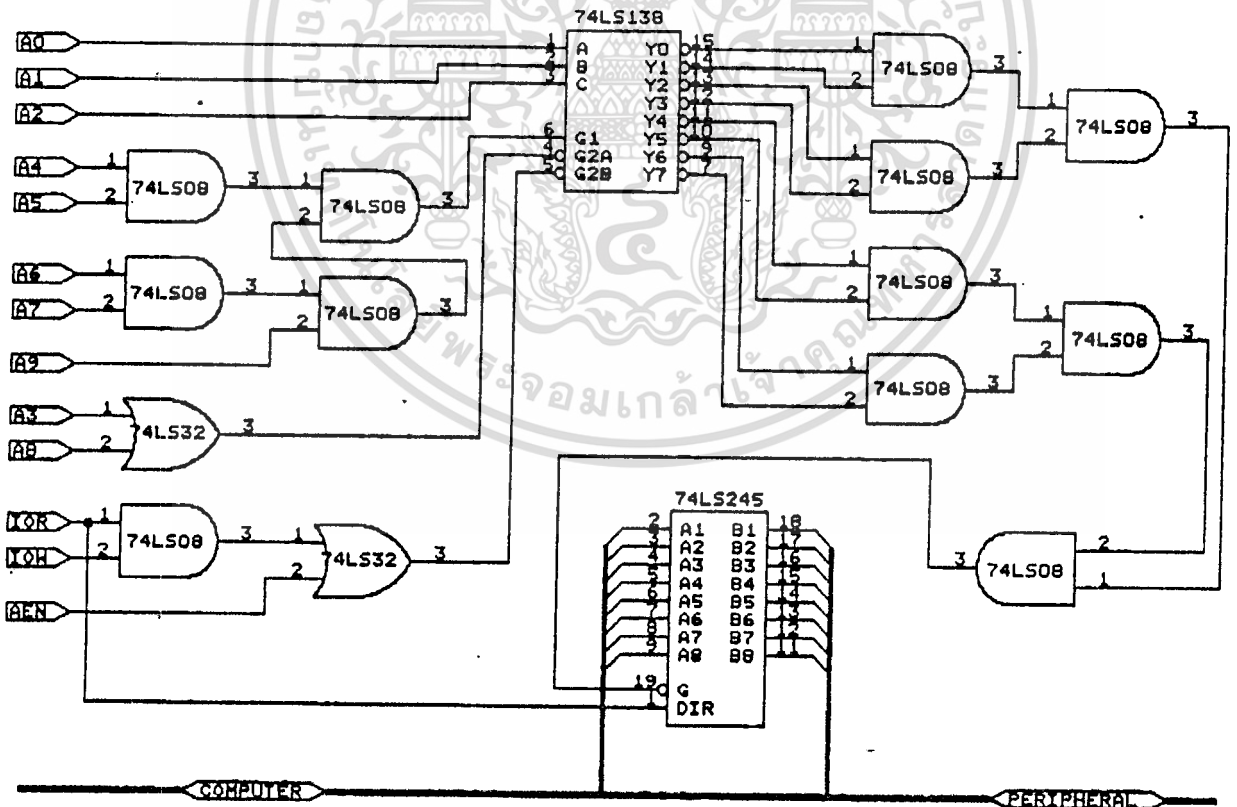
ตำแหน่ง 2F2 เป็นอินพุต (PJIN2) ใช้เป็น 8 บิตบนของวงจรรับความถี่

ตำแหน่ง 2F3 เป็นเอาต์พุต (PJCLR) ใช้เคลียร์วงจรรับความถี่

ตำแหน่ง 2F4 เป็นอินพุต (PJADC) ใช้อ่านค่ากระแสแอมเพอเตอร์ผ่าน ADC

ตำแหน่ง 2F5 ถึง 2F7 ใช้เป็นพอร์ตสำรอง

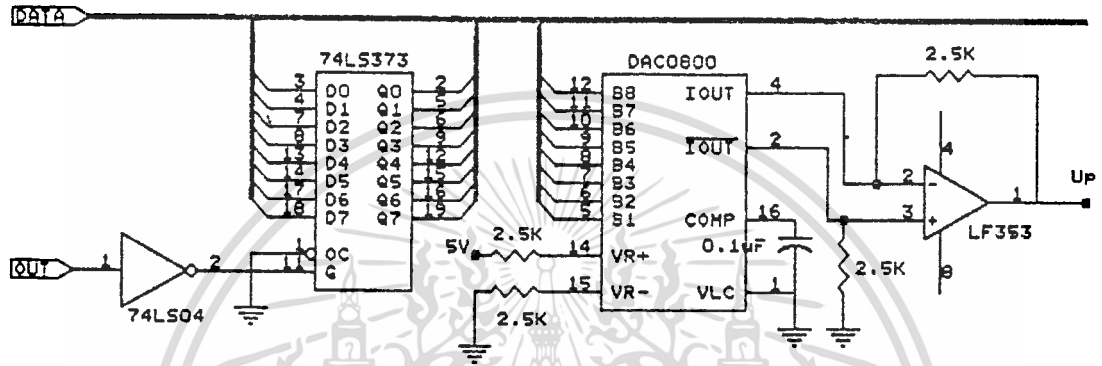
สัญญาณ IOR (IO READ) จากคอมพิวเตอร์จะถูกใช้ควบคุมการทำงาน 74LS245 (Bi-direction buffer) เพื่อกำหนดทิศทางข้อมูล



รูปที่ 3.10 แสดงวงจรถีโค้ดเดอร์

3.3.2. วงจรเอาท์พุท

วงจรเอาท์พุททำหน้าที่แปลงสัญญาณควบคุมจากอแดปทีฟคอนโทรลอัลกอริทึม ซึ่งเป็นสัญญาณดิจิทัลให้เป็นสัญญาณอนาลอกโดยตัวแปลงสัญญาณดิจิทัลเป็นอนาลอก แล้วป้อนสัญญาณไปยังวงจรถ่ายเทียบ เนื่องจากมอเตอร์สามารถรับสัญญาณสูงสุด 75 โวลต์และต่ำสุด -75 โวลต์ ดังนั้นโปรแกรมจึงจำกัดให้สัญญาณควบคุมมีขนาดสูงสุด 75 และต่ำสุด -75 ด้วย



รูปที่ 3.11 แสดงวงจรเอาท์พุท

3.3.3. วงจรอินพุท

วงจรอินพุทประกอบด้วย 2 ส่วนคือ วงจรนับความถี่ และวงจรวัดกระแสมอเตอร์ วงจรนับความถี่จะนับสัญญาณจากพัลส์เอ็นโคดีเดอร์ (Pulse encoder) เพื่อใช้บอกทิศทางและตำแหน่งของมอเตอร์ โดยใช้ความต่างเฟสของพัลส์แชนเนล A และแชนเนล B จากพัลส์เอ็นโคดีเดอร์เพื่อบอกทิศทางการหมุนของมอเตอร์ และใช้ 74LS74 เป็นตัวจัดเงื่อนไขการนับขึ้นหรือนับลงของวงจรถ่ายเทียบ นับความถี่ที่มีขนาด 16 บิตทำให้สามารถนับพัลส์ได้ถึง 65536 พัลส์ และสามารถบอกองศาที่แกนของเฟืองทดรอบหมุนไปโดยคำนวณจาก

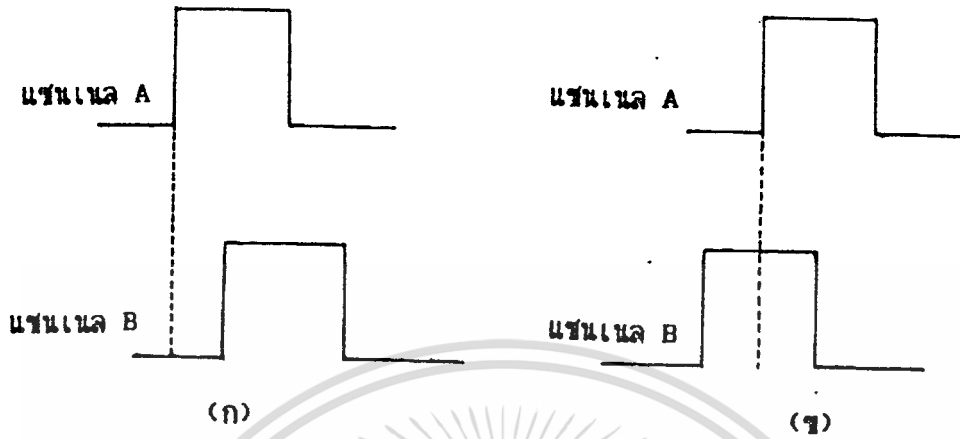
$$\text{deg} = 1000 * N / 360 \tag{3.1}$$

โดยที่ deg คือ จำนวนองศาที่แกนของเฟืองทดรอบหมุน

1000 คือ จำนวนพัลส์ที่เกิดขึ้นจากพัลส์เอ็นโคดีเดอร์เมื่อมอเตอร์หมุน 1 รอบ

N คือ อัตราทดของเฟืองทดรอบ

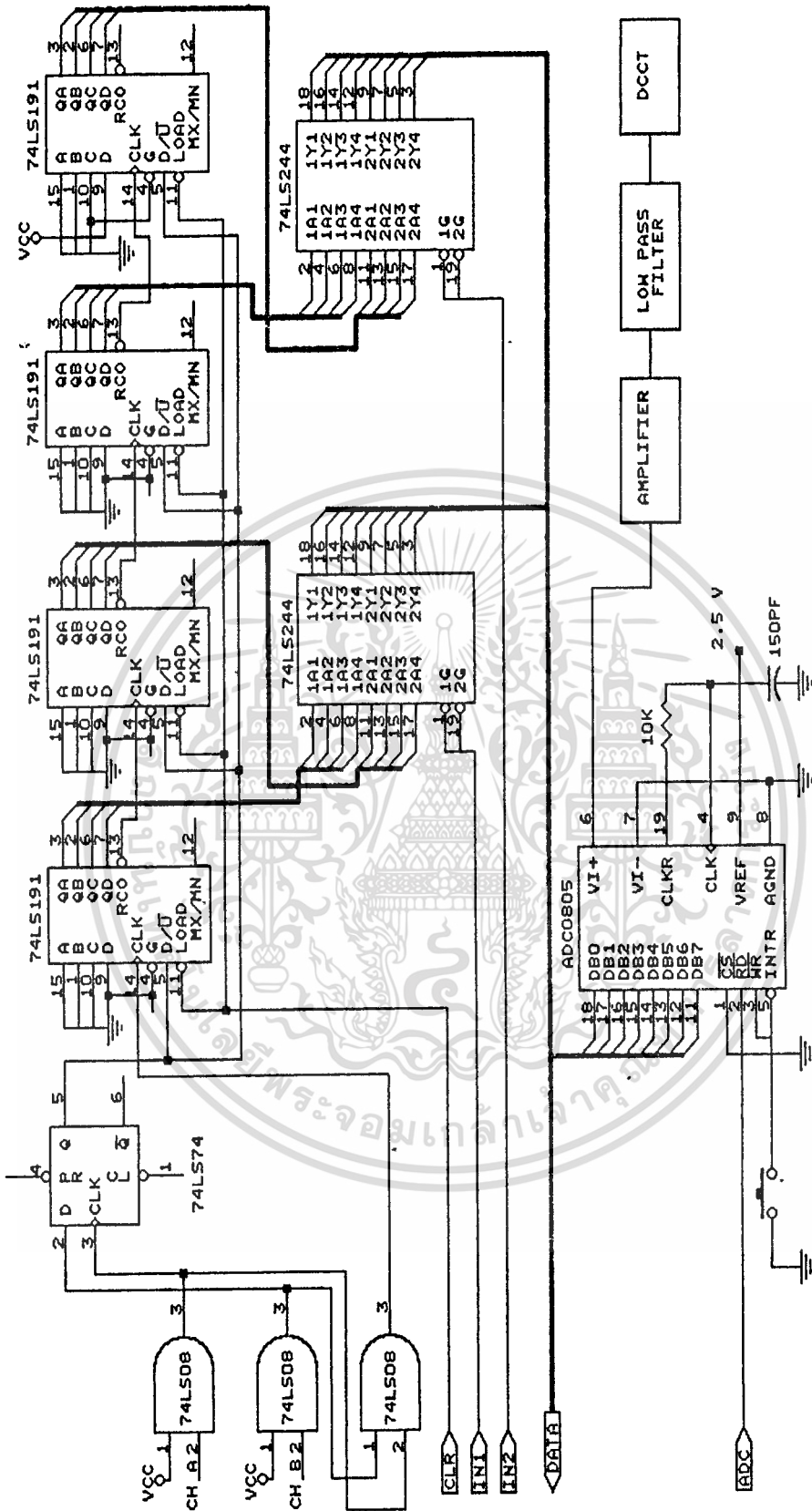
360 คือ จำนวนองศาต่อรอบ



รูปที่ 3.12 แสดงความต่างเฟสของพัลส์จากพัลส์เอ็นดีดีเตอร์

รูปที่ 3.12 ก. เมื่อมอเตอร์หมุนเดินหน้า รูปที่ 3.12 ข. เมื่อมอเตอร์หมุนถอยหลัง

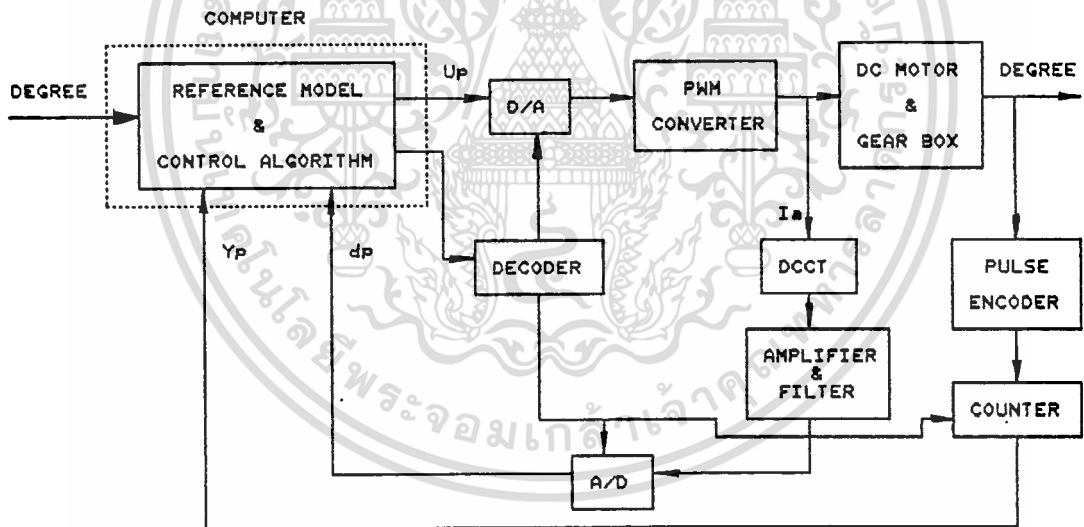
วงจรวัดกระแสมอเตอร์จะรับสัญญาณจาก DCCT (DC Current Transformer) ซึ่งวัดกระแสที่ไหลผ่านมอเตอร์และแปลงเป็นโวลต์เตจด้วยอัตรา 3 แอมป์แปร์ ต่อ 1 โวลต์ สัญญาณจาก DCCT จะถูกกรองโดยวงจรกรองความถี่แบบต่ำผ่าน และปรับให้มีขนาดเหมาะสมโดยวงจรขยาย แล้วจึงป้อนไปยังตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล



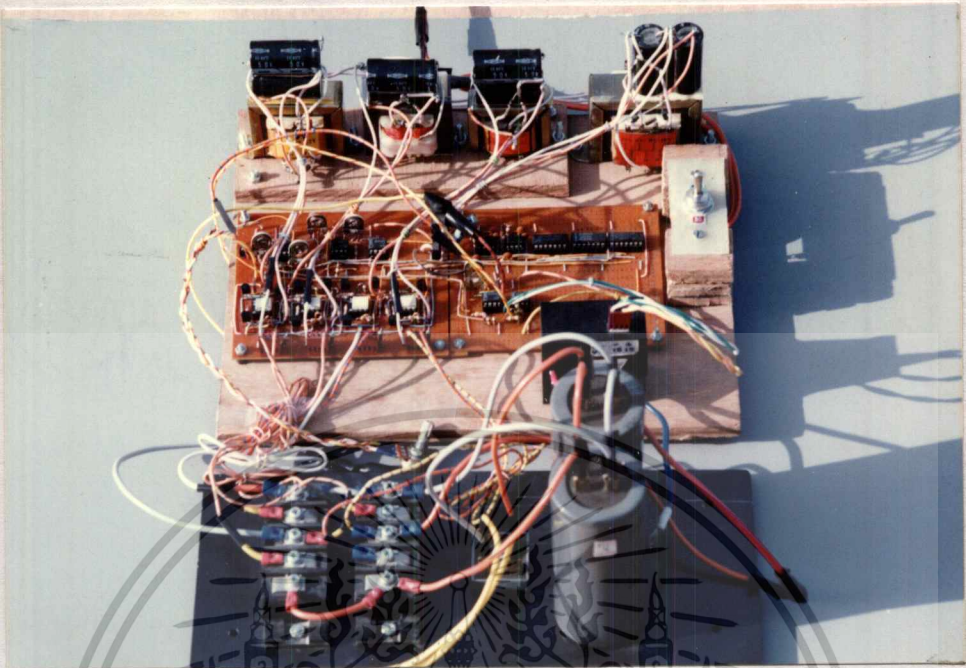
รูปที่ 3.13 แสดงวงจรอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

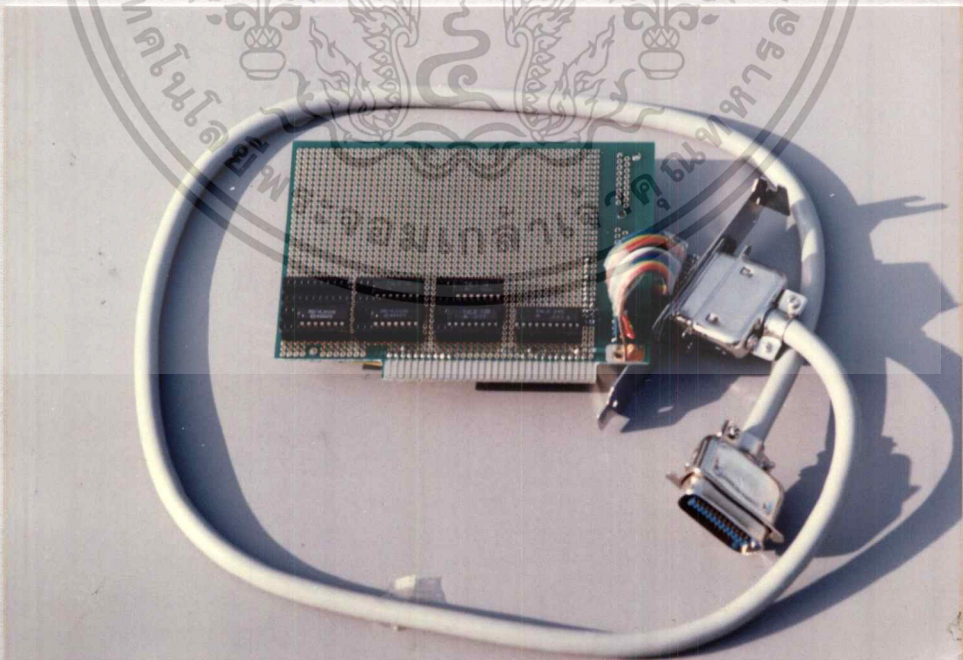
คอนโทรลอัลกอริทึมจะเปรียบเทียบผลตอบสนองของ โมเดลอ้างอิงกับผลตอบสนองของระบบจริงและสร้างสัญญาณควบคุม (u_p) ตามสมการ (2.74) แล้วจึงส่งสัญญาณควบคุมผ่านตัวแปลงดิจิทัลเป็นอนาลอกในวงจรเอาต์พุตไปยังวงจร PWM คอนเวอร์เตอร์ วงจรนับความถี่จะนับพัลส์จากพัลส์เอ็นโค้ดเดอร์และป้อนผ่านวงจรอินพุตไปยังคอนโทรลอัลกอริทึม จำนวนพัลส์ที่นับได้จะถูกแปลงเป็นจำนวนองศาที่มอเตอร์หมุน (y_p) ตามสมการที่ (3.1) DCCT จะแปลงค่ากระแสที่มอเตอร์ใช้เป็นโวลต์เตจ ซึ่งจะถูกรับให้เหมาะสมโดยวงจรขยายและวงจรกรองความถี่แบบต่ำผ่าน หลังจากนั้นจึงส่งค่ากระแส (d_p) ผ่านตัวแปลงอนาลอกเป็นดิจิทัลในวงจรอินพุตไปยังคอนโทรลอัลกอริทึม จำนวนองศาที่มอเตอร์หมุนและกระแสที่มอเตอร์ใช้จะถูกนำมาสร้างเป็นตัวแปรสถานะของระบบควบคุมตำแหน่งโดยตัวสังเกตสถานะตามสมการ (2.89) ตัวแปรสถานะที่ได้จะถูกนำมาใช้ในคอนโทรลอัลกอริทึมเพื่อใช้สร้างสัญญาณควบคุมในครั้งต่อไป



รูปที่ 3.14 แสดงการต่ออุปกรณ์ร่วมกัน

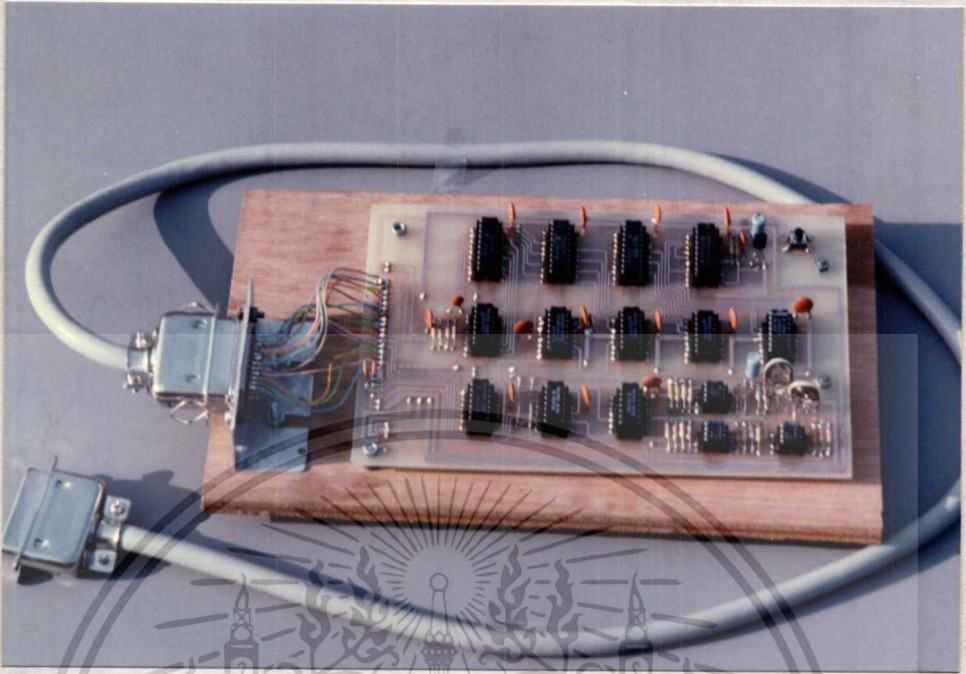


รูปที่ 3.15 รูปถ่ายแสดงวงจรวางจาวอิเล็กทรอนิกส์กำลัง

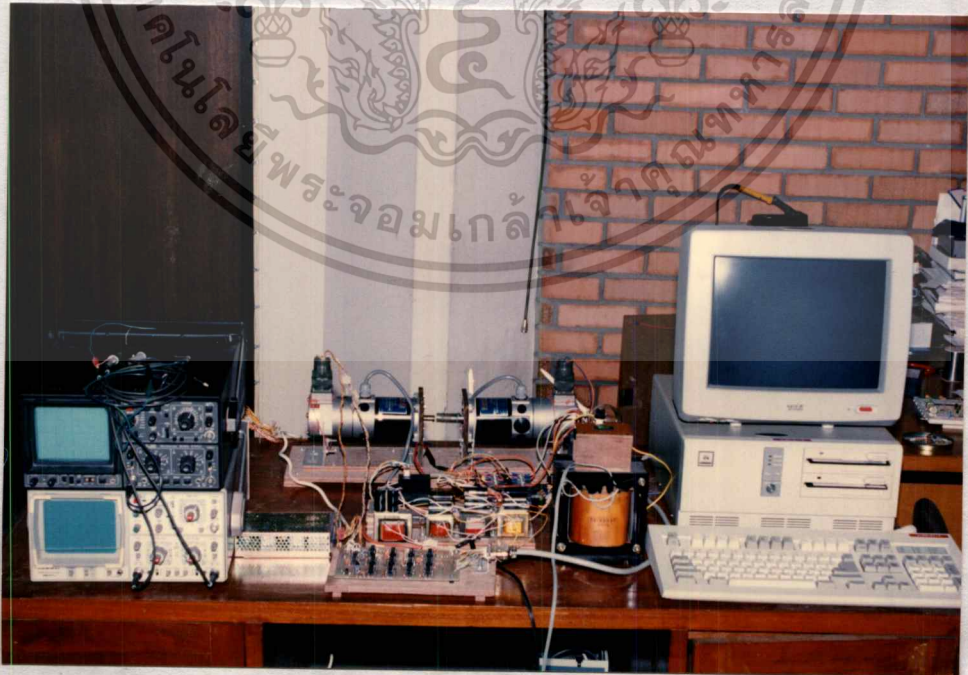


รูปที่ 3.16 รูปถ่ายแสดงวงจรวางจาวดีโคตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 รูปถ่ายแสดงวงจรเอาต์พุตและวงจรอินพุต



รูปที่ 3.18 รูปถ่ายแสดงอุปกรณ์การทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4. สรุป

ในบทที่ 3 ได้กล่าวถึงโครงสร้างของวงจร PWM คอนเวอร์เตอร์ ซึ่งประกอบด้วยวงจรสร้างความถี่ วงจรเปรียบเทียบ วงจรขับเบสและวงจรถูกำลัง นอกจากนี้ยังได้เพิ่มวงจรจัดเฟสเพื่อใช้จัดช่วงเวลาการทำงานของทรานส์ซิสเตอร์ในวงจรถูกำลัง ในส่วนสุดท้ายได้กล่าวถึง การนำ MRAC มาใช้งานในระบบควบคุมตำแหน่งจริง เนื่องจากคอนโทรลอัลกอริทึมเป็นโปรแกรมในคอมพิวเตอร์จึงจำเป็นต้องมีวงจรอินเทอร์เฟสทำหน้าที่เชื่อมโยงระหว่างคอมพิวเตอร์ และระบบควบคุมตำแหน่ง วงจรอินเทอร์เฟสประกอบด้วย วงจรดีโค๊ดเดอร์ วงจรเอาต์พุท และ วงจรอินพุท



บทที่ 4.

ผลการทดลอง

4.1. บทนำ

การทดลองแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ การทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ ซึ่งโปรแกรมจำลองผลตอบสนองได้ถูกแสดงไว้ในภาคผนวก ฉ. และการทดลองในระบบจริง ซึ่งระบบได้ถูกแสดงไว้ในรูปที่ 3.14 การทดลองทั้งสองแบบกระทำทั้งในขณะ ไม่มีโหลด และมีโหลด โหลดที่ใช้ในโปรแกรมจำลองผลตอบสนองมี 2 แบบ คือ สเตปโหลด และโหลดที่เปลี่ยนแปลงตามผลตอบสนองของระบบควบคุมตำแหน่ง ส่วนโหลดที่ใช้ในระบบจริงเกิดจากการต่อแกนของมอเตอร์อีกตัวหนึ่งเข้ากับแกนของมอเตอร์ที่ทำการทดลอง โดยใช้มอเตอร์รุ่น Super U 718 ของบริษัท Sanyo Denki ซึ่งเป็นมอเตอร์รุ่นเดียวกับที่ใช้ในการทดลอง นอกจากนี้ยังได้ทำการทดลองเปรียบเทียบกับระบบที่ควบคุมด้วยตัวควบคุม PI โดยเปรียบเทียบทั้งการจำลองผลตอบสนองด้วยคอมพิวเตอร์ และทดลองในระบบจริง การเปรียบเทียบกระทำทั้งในขณะ ไม่มีโหลด และมีโหลด พารามิเตอร์ของตัวควบคุม PI ถูกออกแบบโดยวิธีซิมเมตริกัลออฟติมัม (symmetrical optimum) ซึ่งได้แสดงไว้ในภาคผนวก จ.

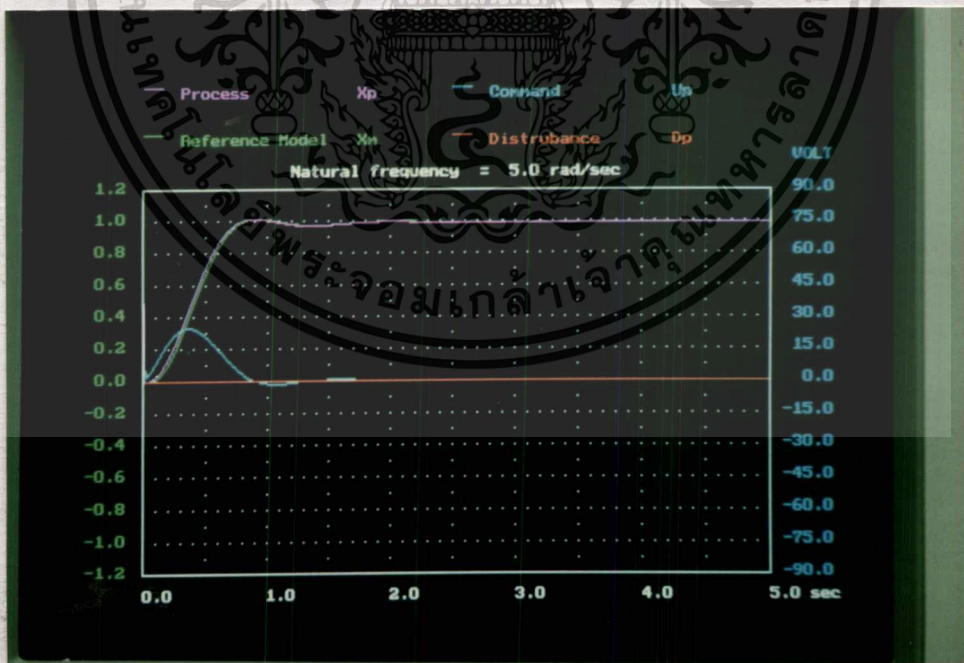
4.2. ผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์

4.2.1. ผลการทดลองในขณะ ไม่มีโหลด

เนื่องจาก โมเดลอ้างอิง ได้ถูกออกแบบให้สามารถตอบสนองต่อคำสั่งอินพุตด้วยความถี่ธรรมชาติใดๆ ซึ่งความถี่ธรรมชาติจะเป็นตัวกำหนดความเร็วการตอบสนองต่อคำสั่งอินพุต นั่นคือสามารถเปลี่ยนแปลงอัตราเร่งของผลตอบสนองได้ ในการทดลองสามารถเปลี่ยนความถี่ธรรมชาติเป็นค่าใดๆ ก็ได้ . ในที่นี้จึงเลือกใช้ความถี่ธรรมชาติ 2 ค่า คือ 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที



รูปที่ 4.1 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที
ไรส์ไทม์ 1.3 วินาที เซ็ตตลิงไทม์ 2.7 วินาที โอเวอร์ชู้ท 2 %

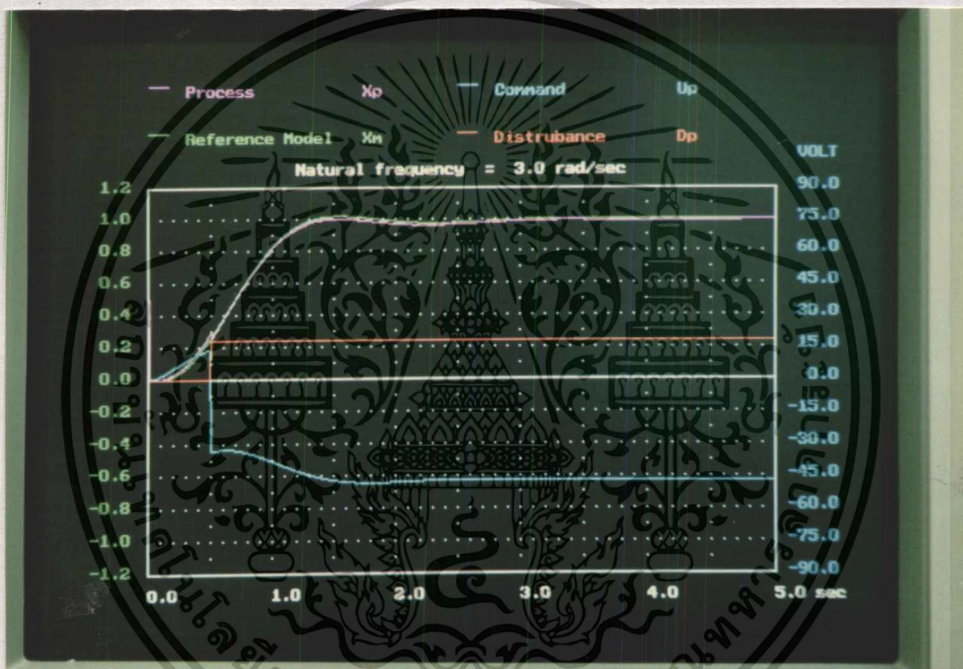


รูปที่ 4.2 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที
ไรส์ไทม์ 0.8 วินาที เซ็ตตลิงไทม์ 1.7 วินาที โอเวอร์ชู้ท 2 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2. ผลการทดลองในขณะมีโหลด

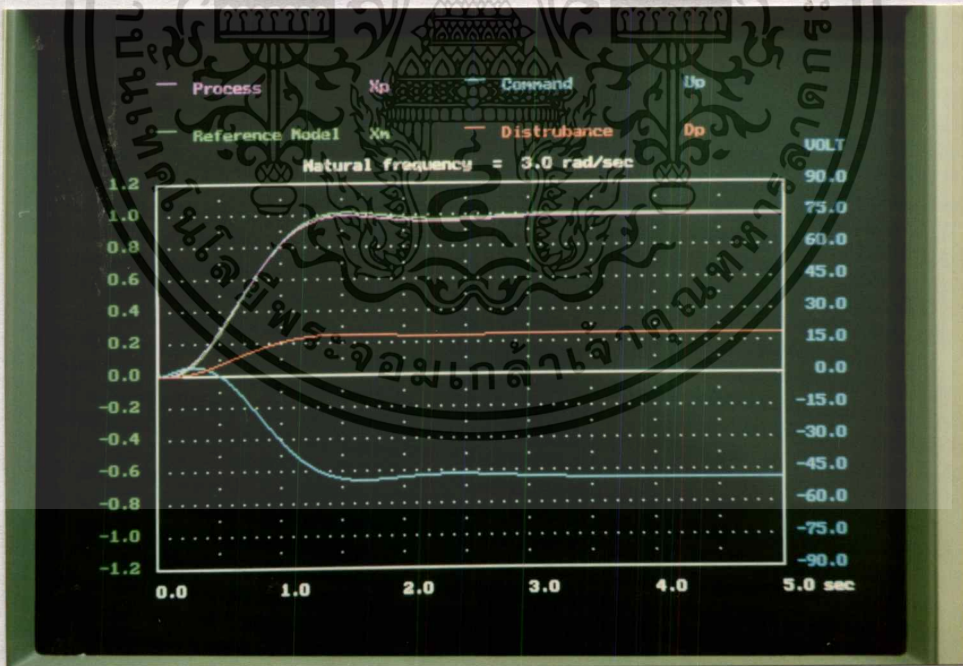
โหลดที่ใช้ในการทดลองมี 2 แบบคือ สเตปโหลด และ โหลดที่เปลี่ยนแปลงตามผลตอบสนองของระบบควบคุมตำแหน่ง ขนาดของโหลดที่ใช้ทดลองคือ 25 % ของคำสั่งอินพุทเมื่อทดลองสเตปโหลด และ 25 % ของผลตอบสนองเมื่อ โหลดเปลี่ยนแปลงตามผลตอบสนองของระบบควบคุมตำแหน่ง



รูปที่ 4.3 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และมีสเตปโหลดขนาด 25 % ของคำสั่งอินพุท
ไรส์ไทม์ 1.35 วินาที เซ็ตติงไทม์ 2.7 วินาที โอเวอร์ชูท 2 %



รูปที่ 4.4 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที และมีสเตรปโหลดขนาด 25 % ของคำสั่งอินพุท ไรส์ไทม์ 0.85 วินาที เซ็ตตลิงไทม์ 1.8 วินาที ไม่มีโอเวอร์ชูท



รูปที่ 4.5 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และมีโหลดขนาด 25 % ของผลตอบสนอง ไรส์ไทม์ 1.35 วินาที เซ็ตตลิงไทม์ 2.7 วินาที โอเวอร์ชูท 2 %



รูปที่ 4.6 แสดงผลการทดลองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที และมีโพลขนาด 25 % ของผลตอบสนอง

ไรส์ไทม์ 0.85 วินาที เซ็ตตลิงไทม์ 1.8 วินาที ไม่มีโอเวอร์ชูท

ช่วงเวลาการตอบสนองและโอเวอร์ชูทได้ถูกรวบรวมไว้ในตาราง 4.1

ความถี่ ธรรมชาติ (เรเดียน /วินาที)	ไม่มีโพล			สเต็ปโพล			โพลตามผลตอบสนอง		
	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)
3	1.3	2.7	2	1.35	2.7	2	1.35	2.7	2
5	0.8	1.7	2	0.85	1.8	0	0.85	1.8	0

t_R = ไรส์ไทม์ t_S = เซ็ตตลิงไทม์เมื่อค่าผิดพลาดต่ำกว่า 2 % M_P = โอเวอร์ชูท

ตารางที่ 4.1 แสดงช่วงเวลาผลตอบสนองและ โอเวอร์ชูทเมื่อควบคุมด้วย MRAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3. ผลการทดลองเปรียบเทียบเกี่ยวกับตัวควบคุม PI ในขณะไม่มีโหลด

ตัวควบคุม PI เป็นตัวควบคุมที่เกิดจากทฤษฎีควบคุมดั้งเดิม และมีใช้ในระบบควบคุมอุตสาหกรรมทั่วไป ในวิทยานิพนธ์นี้จึงทำการทดลองเปรียบเทียบระหว่างระบบที่ควบคุมด้วย MRAC และระบบที่ควบคุมด้วยตัวควบคุม PI ค่าพารามิเตอร์ต่างๆ ของตัวควบคุม PI ถูกออกแบบโดยวิธี ซิมเมตริกคอลลอพติซึม และนำมาเปรียบเทียบกับผลตอบสนองของระบบที่ควบคุมด้วย MRAC เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที



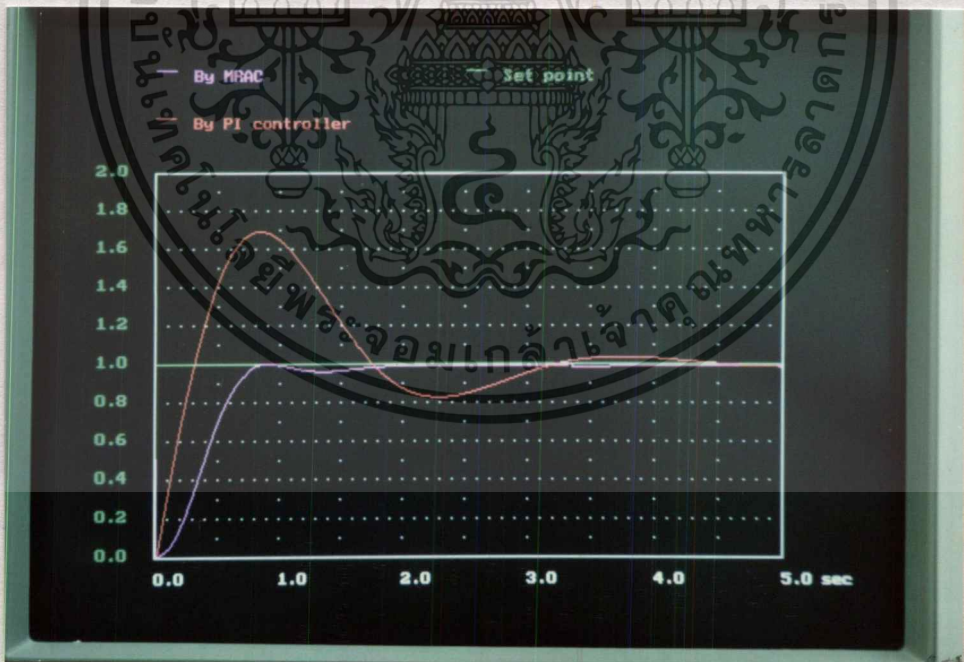
รูปที่ 4.7 แสดงผลตอบสนองเปรียบเทียบระหว่างตัวควบคุม MRAC และตัวควบคุม PI ควบคุมด้วย MRAC มีไรส์ไทม์ 0.8 วินาที เซ็ตตลิ่งไทม์ 1.7 วินาที โอเวอร์ชูต 2 % ควบคุมด้วย PI มีไรส์ไทม์ 0.45 วินาที เซ็ตตลิ่งไทม์ 2.1 วินาที โอเวอร์ชูต 18 %

4.2.4. ผลการทดลองเปรียบเทียบเกี่ยวกับตัวควบคุม PI ในขณะมีโหลด

การทดลองเพื่อเปรียบเทียบ จะทำการทดลองกับโหลดแบบเดียวกับการทดลองที่ผ่านมาแล้วในหัวข้อ 4.2.2. คือ ใช้สแตปโหลดขนาด 25 % ของคำสั่งอินพุท และโหลดขนาด 25 % ของ ผลตอบสนองของระบบควบคุมตำแหน่ง และนำมาเปรียบเทียบกับผลตอบสนองของระบบควบคุมตำแหน่งเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที



รูปที่ 4.8 แสดงผลตอบสนองเปรียบเทียบเมื่อมีสแตปโหลดขนาด 25 % ของคำสั่งอินพุต ความคุมด้วย MRAC มีไรส์ไทม์ 0.85 วินาที เซ็ตตลิ่งไทม์ 1.8 วินาที ไม่มีโอเวอร์ชูท ความคุมด้วย PI มีไรส์ไทม์ 0.45 วินาที เซ็ตตลิ่งไทม์ 2.4 วินาที โอเวอร์ชูท 50 %



รูปที่ 4.9 แสดงผลตอบสนองเปรียบเทียบเมื่อมีโหลดขนาด 25 % ของผลตอบสนอง ความคุมด้วย MRAC มีไรส์ไทม์ 0.85 วินาที เซ็ตตลิ่งไทม์ 1.8 วินาที ไม่มีโอเวอร์ชูท ความคุมด้วย PI มีไรส์ไทม์ 0.35 วินาที เซ็ตตลิ่งไทม์ 4.2 วินาที โอเวอร์ชูท 70 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองเปรียบเทียบระหว่างระบบที่ควบคุมด้วย MRAC เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที กับระบบที่ควบคุมด้วยตัวควบคุม PI

ตัวควบคุม	ไม่มีโหลด			สแตปโหลด			โหลดตามผลตอบสนอง		
	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)
MRAC	0.8	1.7	2	0.85	1.8	0	0.85	1.8	0
PI	0.45	2.1	18	0.45	2.4	50	0.35	4.2	70

t_R = ไรส์ไทม์ t_S = เซ็ตติงไทม์เมื่อค่าผิดพลาดต่ำกว่า 2% M_P = โอเวอร์ชูท

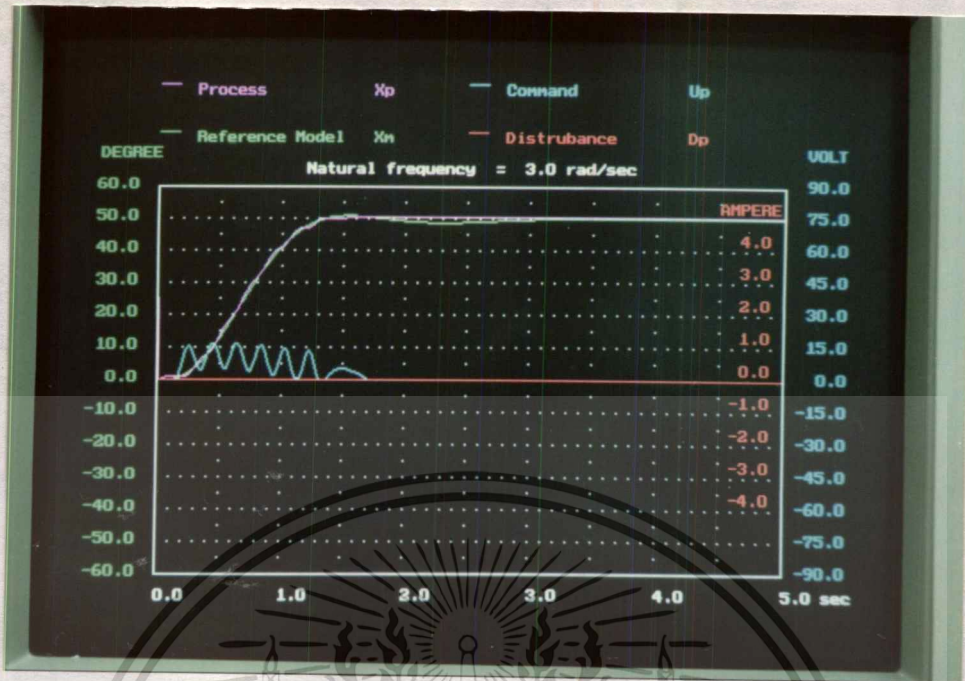
ตารางที่ 4.2 เปรียบเทียบช่วงเวลาผลตอบสนองและ โอเวอร์ชูทเมื่อควบคุมด้วย MRAC และ PI

4.3. ผลการทดลองในระบบจริง

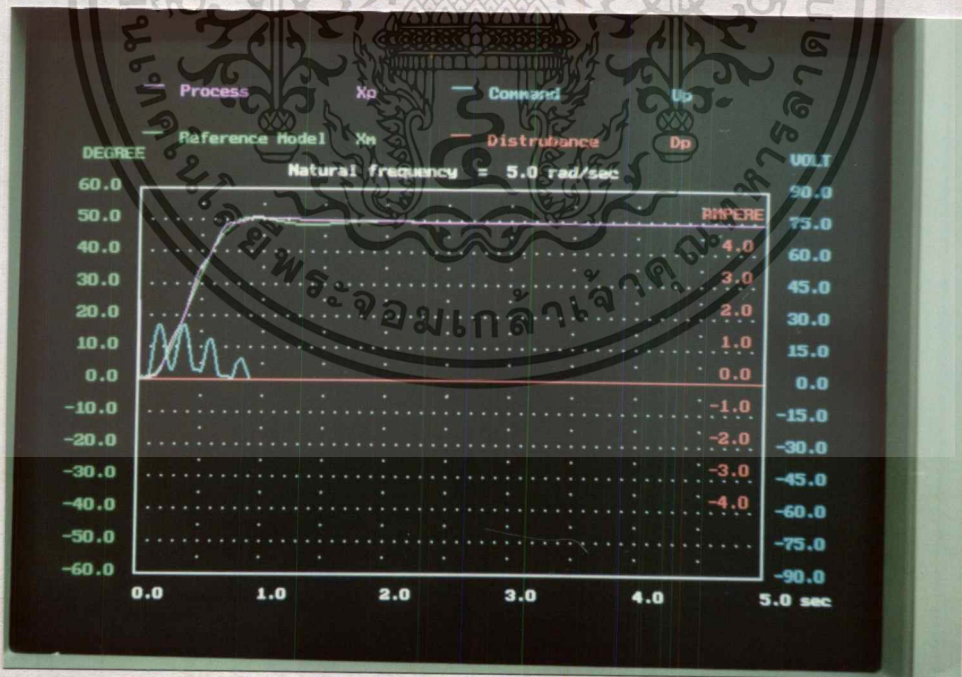
การทดลองในระบบจริง จะกระทำเช่นเดียวกับการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ คือทดลองในขณะ ไม่มีโหลด มีโหลด และเปรียบเทียบกับระบบที่ควบคุมด้วยตัวควบคุม PI

4.3.1. ผลการทดลองในขณะ ไม่มีโหลด

การทดลองในระบบจริง จะกำหนดให้โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที เช่นเดียวกับการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ และกำหนดให้มอเตอร์หมุนเป็นมุม 50 องศา 150 องศา และ 300 องศา



รูปที่ 4.10 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที
มอเตอร์หมุนเบี่ยงมุม 50 องศา
โวลต์ใหม่ 1.3 วินาที เซ็ตดิ่งใหม่ 1.5 วินาที ไม่มีโอเวอร์ชุต

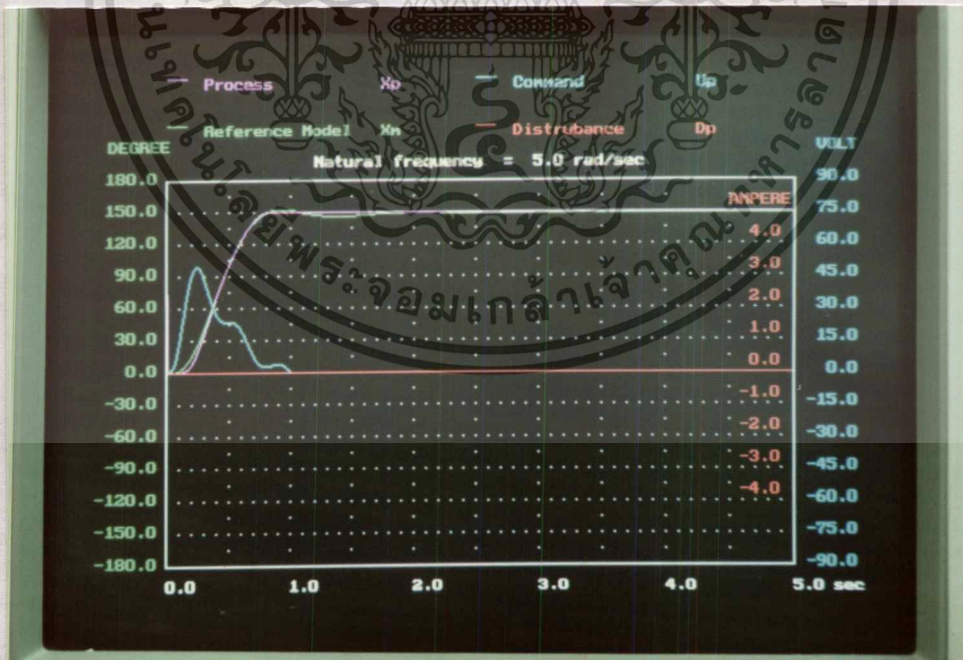


รูปที่ 4.11 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที
มอเตอร์หมุนเบี่ยงมุม 50 องศา
โวลต์ใหม่ 0.9 วินาที เซ็ตดิ่งใหม่ 0.9 วินาที ไม่มีโอเวอร์ชุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงผลตอบสนองเมื่อ โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที
มอเตอร์หมุนเบี่ยงมุม 150 องศา
โวลต์ใหม่ 1.4 วินาที เซ็ตตั้งใหม่ 1.4 วินาที ไม่มีโอเวอร์ชูท

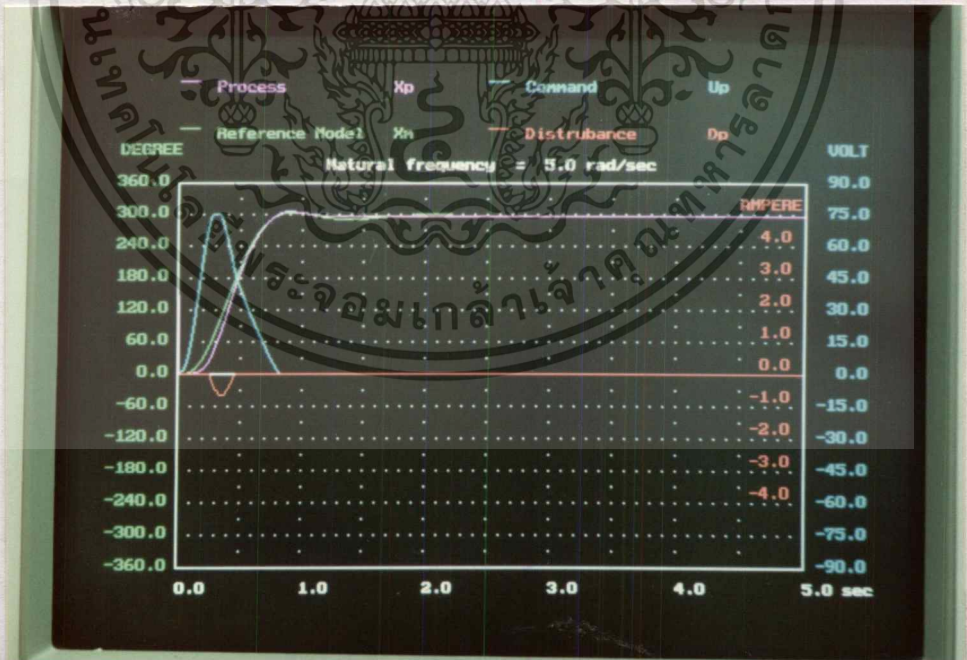


รูปที่ 4.13 แสดงผลตอบสนองเมื่อ โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที
มอเตอร์หมุนเบี่ยงมุม 150 องศา
โวลต์ใหม่ 0.8 วินาที เซ็ตตั้งใหม่ 0.8 วินาที ไม่มีโอเวอร์ชูท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที
มอเตอร์หมุนเป็นมุม 300 องศา
โวลต์ใหม่ 1.4 วินาที เซ็ตคั้งใหม่ 1.4 วินาที ไม่มีโอเวอร์ชู้ท

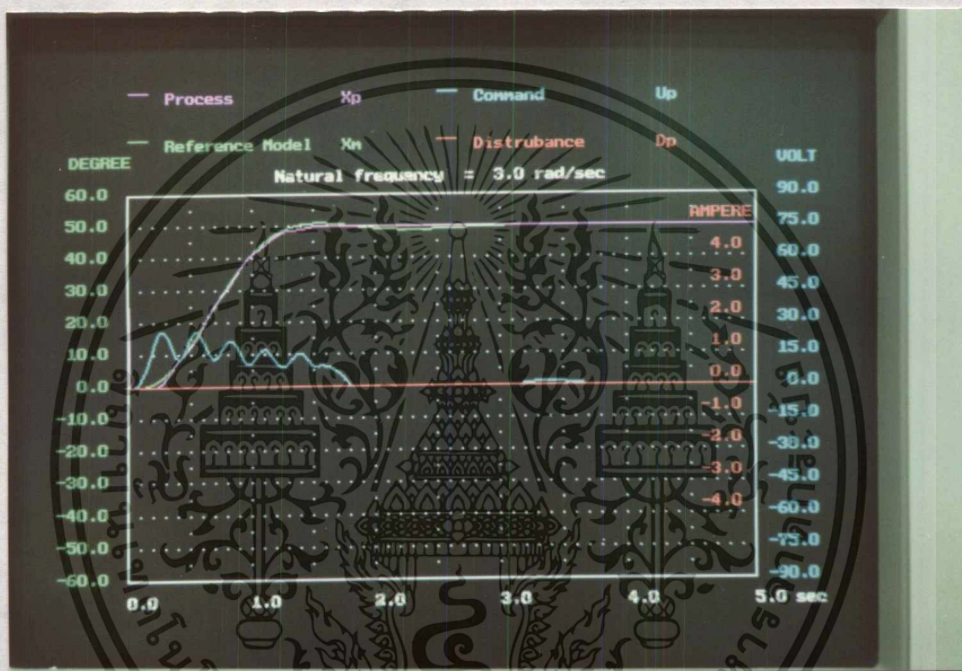


รูปที่ 4.15 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที
มอเตอร์หมุนเป็นมุม 300 องศา
โวลต์ใหม่ 0.8 วินาที เซ็ตคั้งใหม่ 1.0 วินาที โอเวอร์ชู้ท 2 %

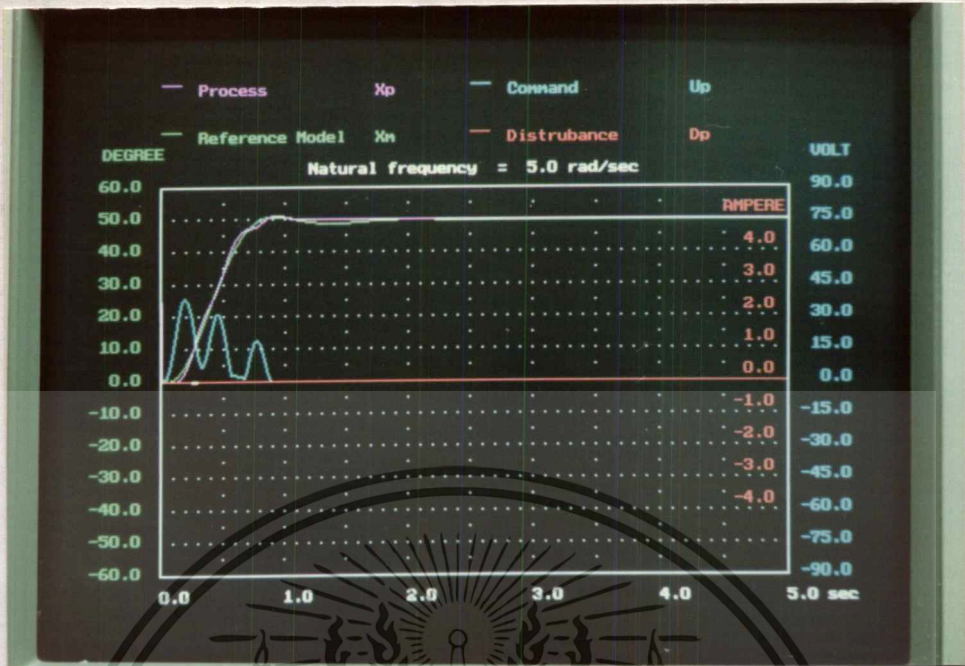
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2. ผลการทดลองในขณะมีโหลด

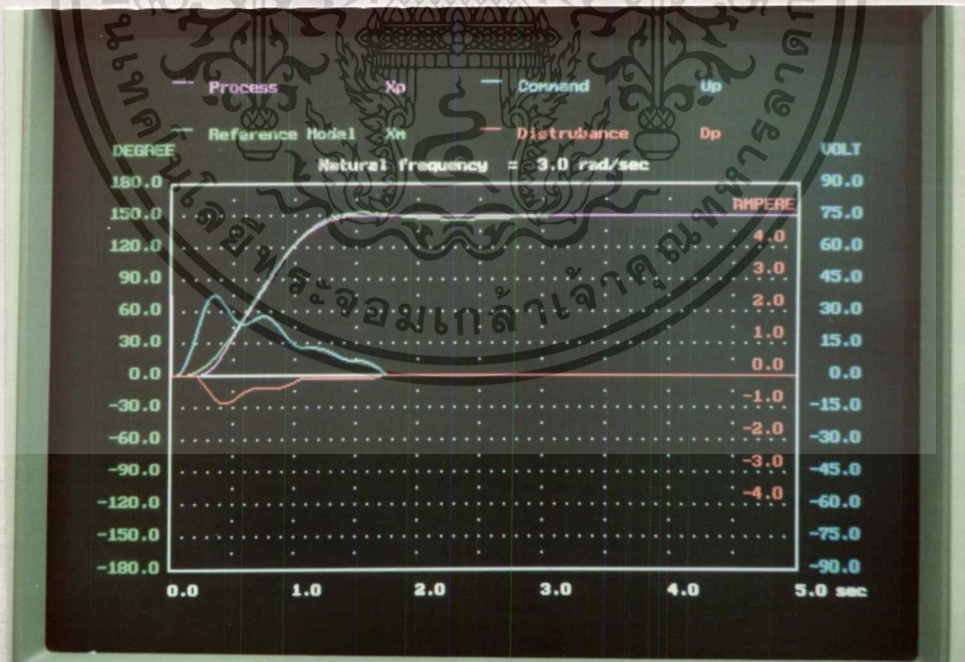
โหลดที่ใช้ในการทดลองเกิดจากการต่อแกนของมอเตอร์อีกตัวหนึ่ง ซึ่งเป็นมอเตอร์รุ่นเดียวกับมอเตอร์ที่ทำการทดลองเข้ากับแกนของมอเตอร์ที่ทำการทดลอง ตั้งให้มอเตอร์ตัวใหม่จะทำหน้าที่เป็นเครื่องกำเนิดไฟฟ้า ขนาดของโหลดเปลี่ยนแปลงโดยเปิดหรือปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า



รูปที่ 4.16 แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา ไรล์ใหม่ 1.5 วินาที เซ็ตตลิ่งใหม่ 1.5 วินาที ไม่มีโอเวอร์ชูท



รูปที่ 4.17 แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา ไรส์ไทม์ 0.9 วินาที เซ็ตตลิ่งใหม่ 1.0 วินาที โอเวอร์ชุต 1 %

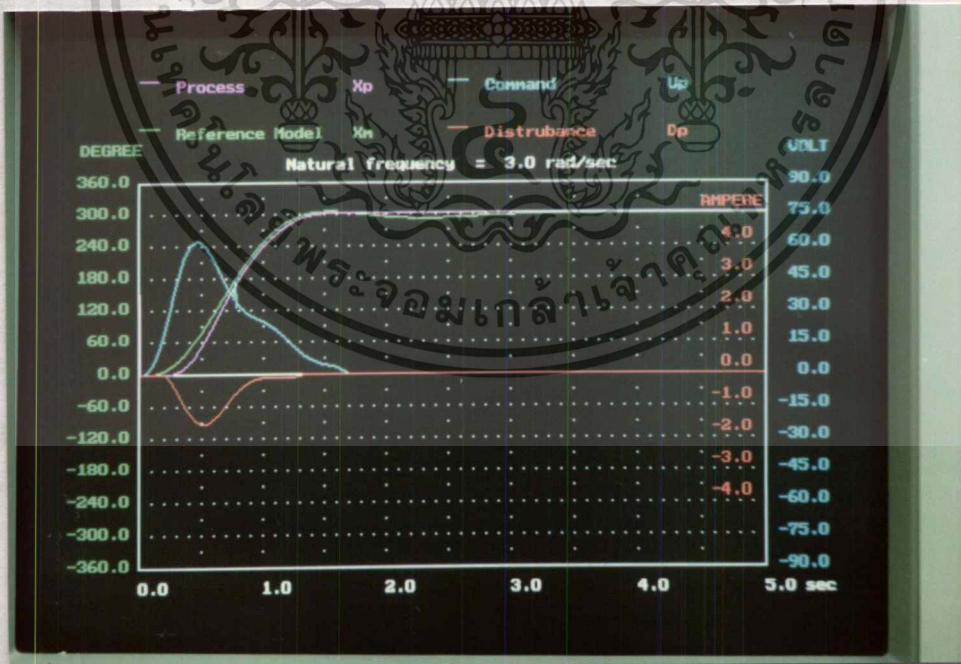


รูปที่ 4.18 แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา ไรส์ไทม์ 1.4 วินาที เซ็ตตลิ่งใหม่ 1.4 วินาที ไม่มีโอเวอร์ชุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงผลตอบสนองเมื่อเปิดจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเบ็บบม 150 องศา ไรล์ใหม่ 0.8 วินาที เซ็ตดิ่งใหม่ 1.2 วินาที โอเวอร์ชุต 2%

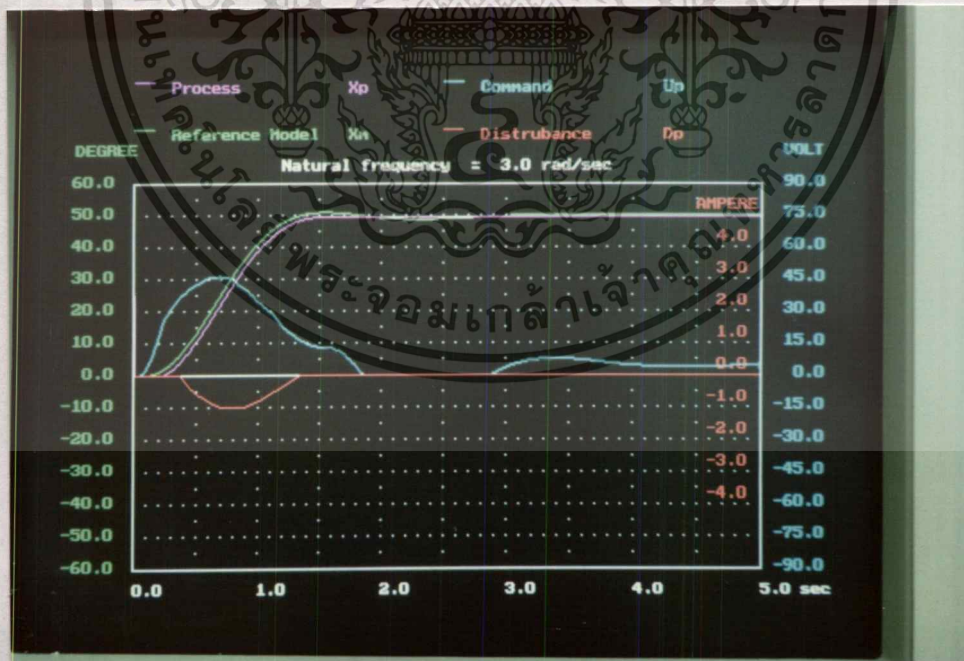


รูปที่ 4.20 แสดงผลตอบสนองเมื่อเปิดจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเบ็บบม 300 องศา ไรล์ใหม่ 1.4 วินาที เซ็ตดิ่งใหม่ 1.4 วินาที ไม่มีโอเวอร์ชุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 แสดงผลตอบสนองเมื่อเปิดจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา ไรส์ไทม์ 0.9 วินาที เซ็ตตลิงไทม์ 1.3 วินาที โอเวอร์ชูท 5 %

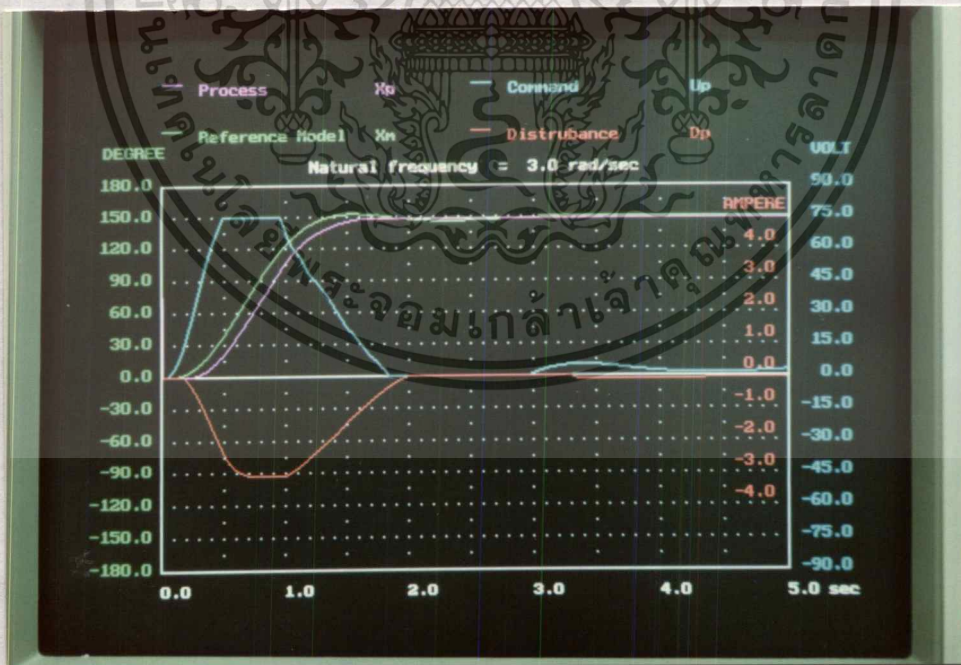


รูปที่ 4.22 แสดงผลตอบสนองเมื่อเปิดจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา ไรส์ไทม์ 1.5 วินาที เซ็ตตลิงไทม์ 1.5 วินาที ไม่มีโอเวอร์ชูท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

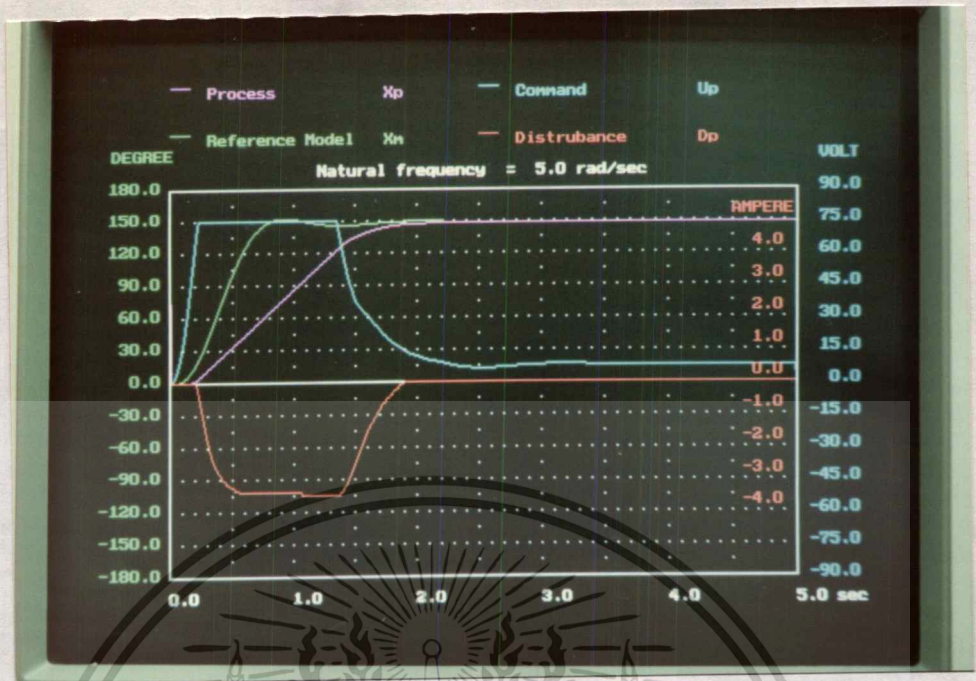


รูปที่ 4.23 แสดงผลตอบสนองเมื่อเปิดชั่วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา ไรส์ไทม์ 0.9 วินาที เซ็ตดลิ่งไทม์ 0.9 วินาที ไม่มีโอเวอร์ชูท

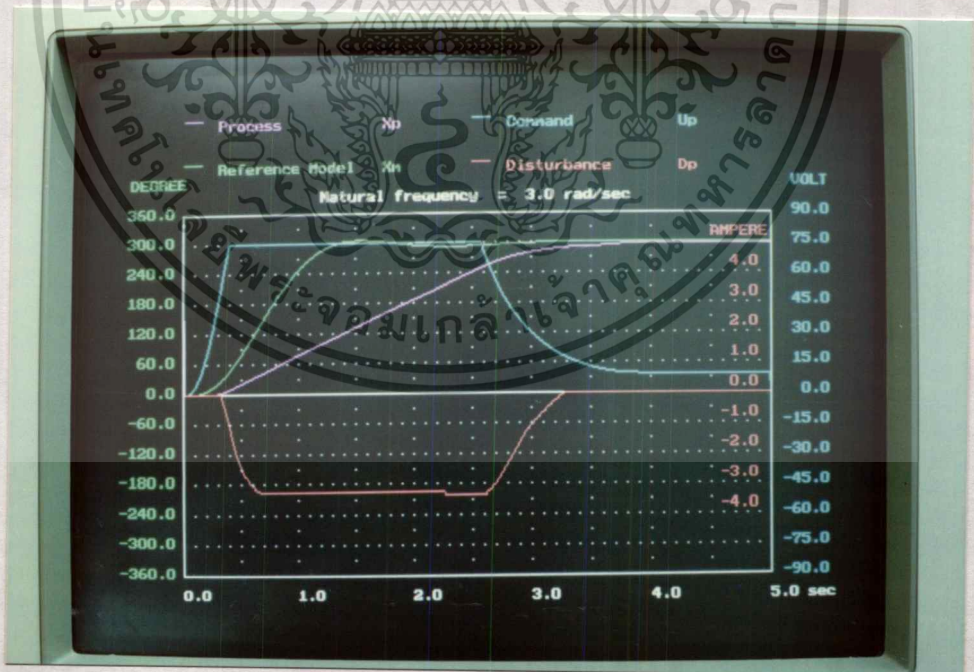


รูปที่ 4.24 แสดงผลตอบสนองเมื่อเปิดชั่วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา ไรส์ไทม์ 1.6 วินาที เซ็ตดลิ่งไทม์ 1.6 วินาที ไม่มีโอเวอร์ชูท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 แสดงผลตอบสนองเมื่อปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 150 องศา ไรล์ใหม่ 2.1 วินาที เซ็ตตลงใหม่ 2.1 วินาที ไมมีโอเวอร์ชูต



รูปที่ 4.26 แสดงผลตอบสนองเมื่อปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า โมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา ไรล์ใหม่ 3.75 วินาที เซ็ตตลงใหม่ 3.75 วินาที ไมมีโอเวอร์ชูต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 แสดงผลตอบสนองเมื่อเปิดจ่ายไฟของเครื่องกำเนิดไฟฟ้า
โมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 300 องศา
โวลต์ใหม่ 3.1 วินาที เซ็ตคั้งใหม่ 3.1 วินาที ไม่มีโอเวอร์ชูท

ช่วงเวลาการตอบสนองและโอเวอร์ชูทของผลการทดลองในระบบจริง เมื่อโมเดลอ้างอิง
มีความถี่ธรรมชาติ 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที ทั้งในขณะไม่มีโหลด และมี
โหลด เมื่อมอเตอร์หมุนเป็นมุมต่างๆได้ถูกรวบรวมไว้ในตารางที่ 4.3

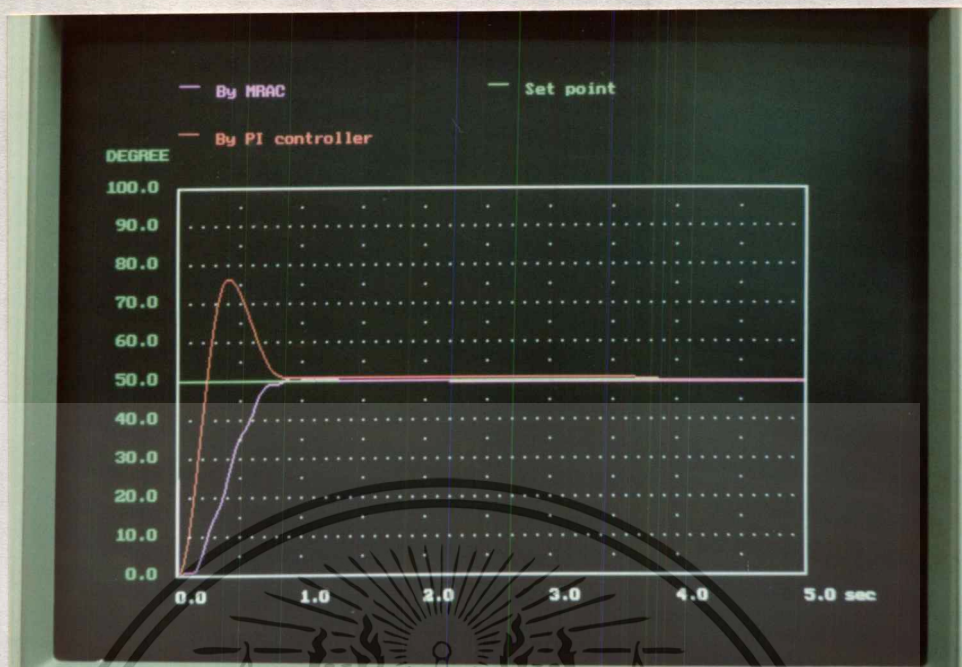
ความถี่ ธรรมชาติ (เรเดียน /วินาที)	หมุน องศา	ไม่มีโหลด			เปิดหัวเครื่องกำเนิดไฟ			ปิดหัวเครื่องกำเนิดไฟ		
		t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)
3	50	1.3	1.5	0	1.5	1.5	0	1.5	1.5	0
	150	1.4	1.4	0	1.4	1.4	0	1.6	1.6	0
	300	1.4	1.4	0	1.4	1.4	0	3.75	3.75	0
5	50	0.9	0.9	0	0.9	1.0	1	0.9	0.9	0
	150	0.8	0.8	0	0.8	1.2	2	2.1	2.1	0
	300	0.8	1.0	2	0.9	1.3	5	3.1	3.1	0

t_R = ไรล์ใหม่ t_S = เซ็ตคั้งใหม่เมื่อค่าผิดพลาดต่ำกว่า 2 % M_P = โอเวอร์ชูท

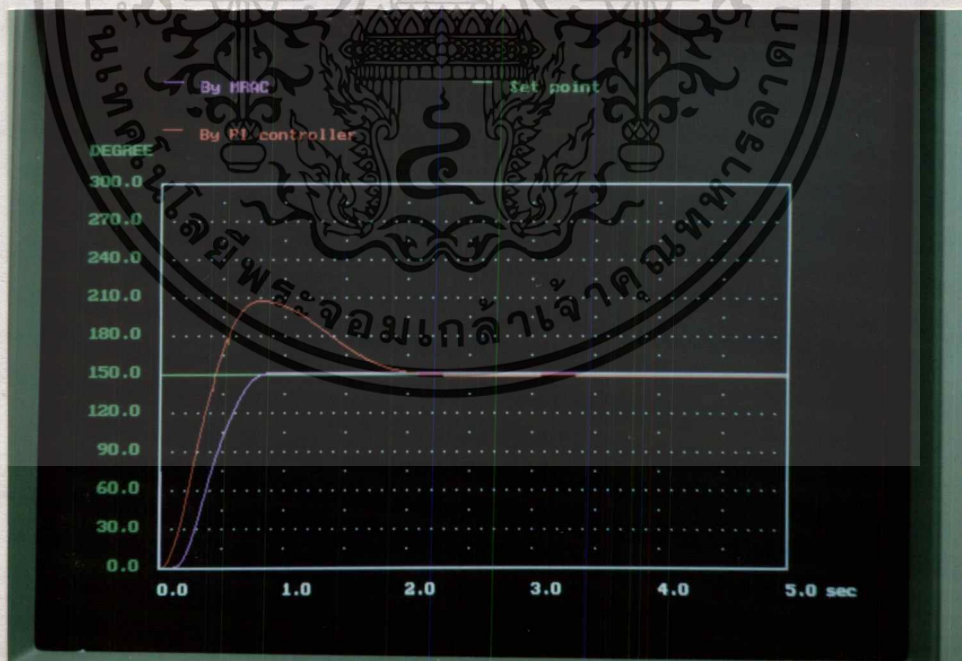
ตารางที่ 4.3 แสดงช่วงเวลาผลตอบสนองและ โอเวอร์ชูทเมื่อควบคุมด้วย MRAC
เมื่อมอเตอร์หมุน 50 องศา 150 องศา และ 300 องศา

4.3.3. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะที่ไม่มีโหลด

ระบบที่ควบคุมด้วยตัวควบคุม PI ได้ถูกทดลองในลักษณะเดียวกันเพื่อนำผลการทดลองมาเปรียบเทียบกับระบบที่ควบคุมด้วย MRAC โดยจะเปรียบเทียบกับผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที

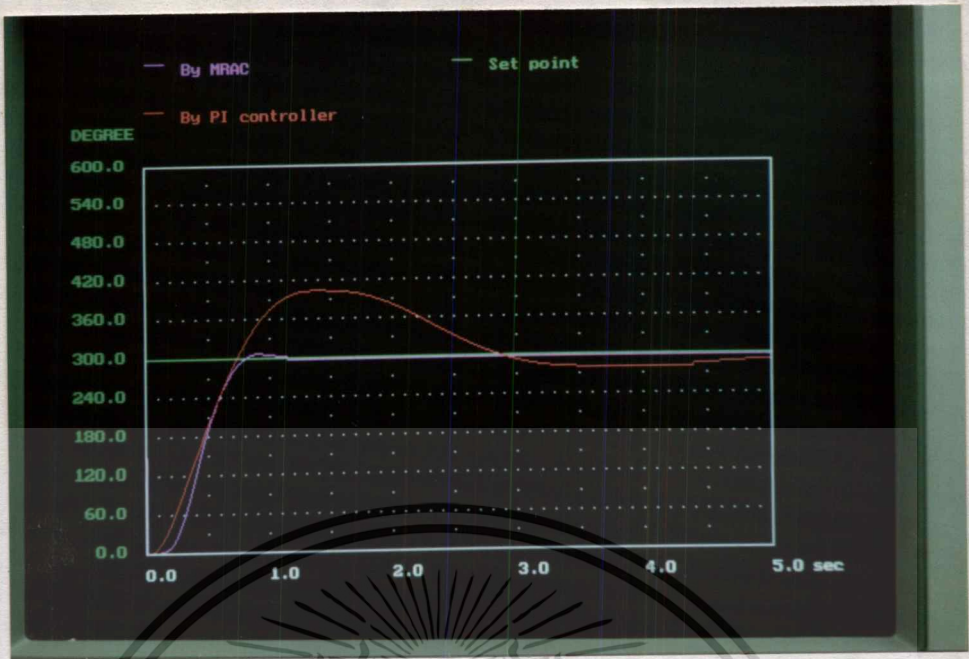


รูปที่ 4.28 แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 50 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.9 วินาที เซ็ตตลิงไทม์ 0.9 วินาที ไม่มีโอเวอร์ชูท
ควบคุมด้วย PI มีไรส์ไทม์ 0.25 วินาที เซ็ตตลิงไทม์ 0.9 วินาที โอเวอร์ชูท 55 %



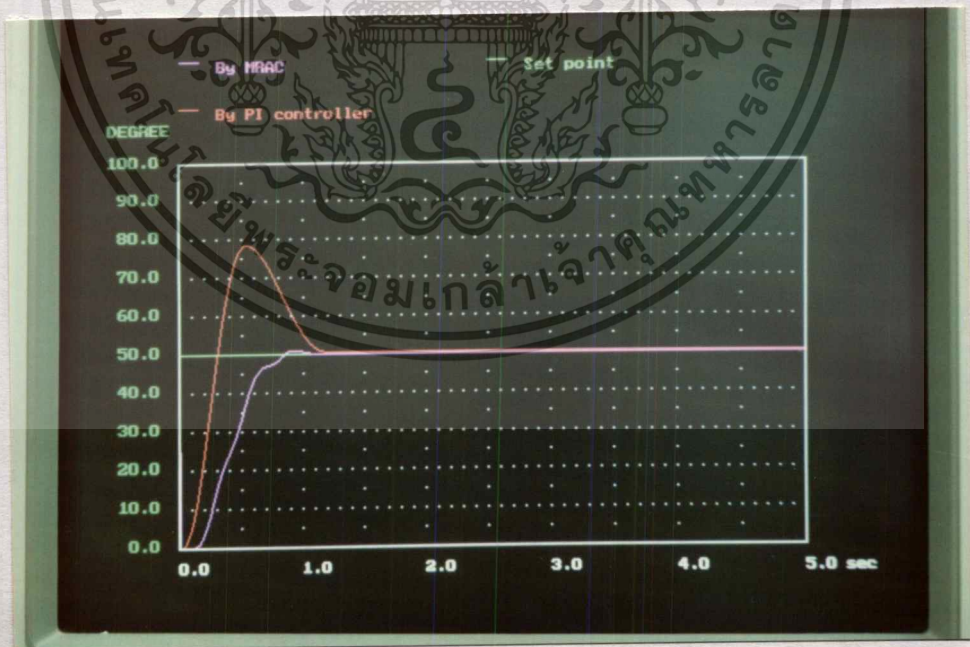
รูปที่ 4.29 แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 150 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.8 วินาที เซ็ตตลิงไทม์ 0.8 วินาที ไม่มีโอเวอร์ชูท
ควบคุมด้วย PI มีไรส์ไทม์ 0.45 วินาที เซ็ตตลิงไทม์ 1.9 วินาที โอเวอร์ชูท 39 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 แสดงผลตอบสนองเปรียบเทียบเมื่อมอเตอร์หมุนเป็นมุม 300 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.8 วินาที เซ็ตตลิ่งไทม์ 1.0 วินาที โอเวอร์ชูท 2 %
ควบคุมด้วย PI มีไรส์ไทม์ 0.7 วินาที เซ็ตตลิ่งไทม์ > 5.0 วินาที โอเวอร์ชูท 35 %

4.3.4. ผลการทดลองเปรียบเทียบกับตัวควบคุม PI ในขณะมีโหลด

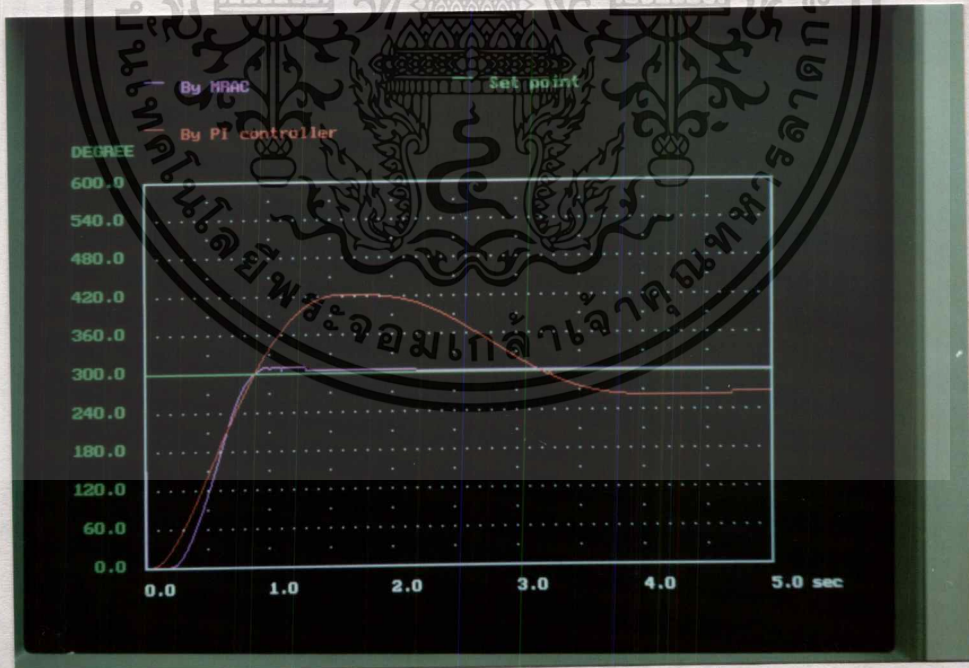


รูปที่ 4.31 แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดชั่วเครื่องกำเนิดไฟ มอเตอร์หมุน 50 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.9 วินาที เซ็ตตลิ่งไทม์ 1.0 วินาที โอเวอร์ชูท 1 %
ควบคุมด้วย PI มีไรส์ไทม์ 0.3 วินาที เซ็ตตลิ่งไทม์ 1.2 วินาที โอเวอร์ชูท 58 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดหัวเครื่องกำเนิดไฟ มอเตอร์หมุน 150 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.8 วินาที เซ็ตตลิ่งใหม่ 1.2 วินาที โอเวอร์ชู้ท 2 %
ควบคุมด้วย PI มีไรส์ไทม์ 0.55 วินาที เซ็ตตลิ่งใหม่ > 5.0 วินาที โอเวอร์ชู้ท 45 %

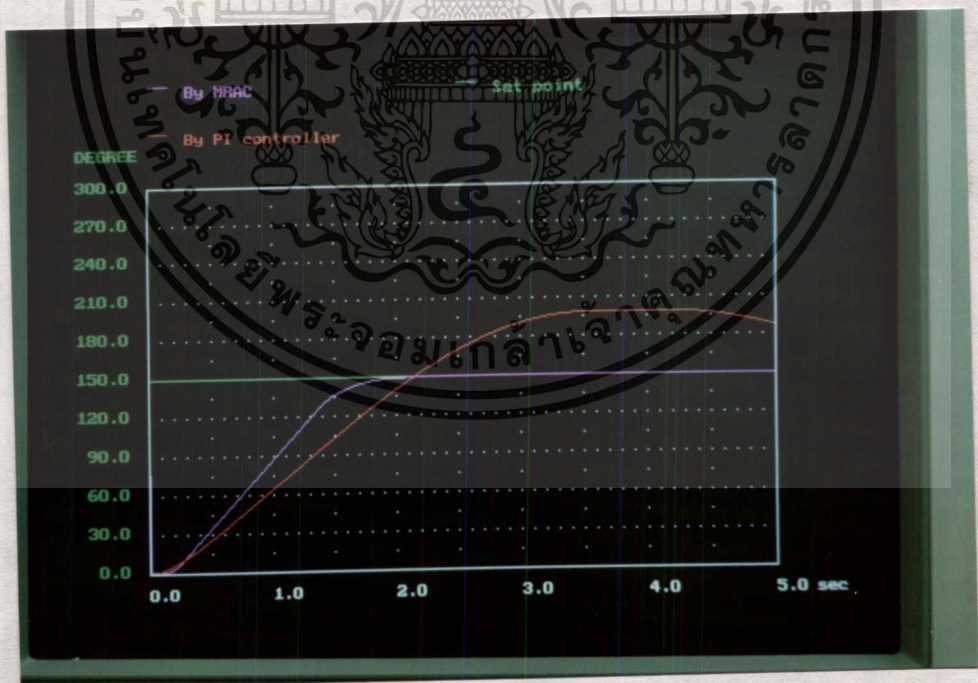


รูปที่ 4.33 แสดงผลตอบสนองเปรียบเทียบเมื่อเปิดหัวเครื่องกำเนิดไฟ มอเตอร์หมุน 300 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.9 วินาที เซ็ตตลิ่งใหม่ 1.3 วินาที โอเวอร์ชู้ท 5 %
ควบคุมด้วย PI มีไรส์ไทม์ 0.85 วินาที เซ็ตตลิ่งใหม่ > 5.0 วินาที โอเวอร์ชู้ท 40 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 แสดงผลตอบสนองเปรียบเทียบเมื่อปิดหัวเครื่องกำเนิดไฟ มอเตอร์หมุน 50 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 0.9 วินาที เซ็ตตลิ่งไทม์ 0.9 วินาที ไม่มีโอเวอร์ชูท
ควบคุมด้วย PI มีไรส์ไทม์ 0.8 วินาที เซ็ตตลิ่งไทม์ 3.5 วินาที โอเวอร์ชูท 28 %



รูปที่ 4.35 แสดงผลตอบสนองเปรียบเทียบเมื่อปิดหัวเครื่องกำเนิดไฟ มอเตอร์หมุน 150 องศา
ควบคุมด้วย MRAC มีไรส์ไทม์ 2.1 วินาที เซ็ตตลิ่งไทม์ 2.1 วินาที ไม่มีโอเวอร์ชูท
ควบคุมด้วย PI มีไรส์ไทม์ 2.15 วินาที เซ็ตตลิ่งไทม์ > 5.0 วินาที โอเวอร์ชูท 32 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.36 แสดงผลตอบสนองเปรียบเทียบเมื่อปิดขั้วเครื่องกำเนิดไฟ มอเตอร์หมุน 300 องศาควบคุมด้วย MRAC มีไรส์ไทม์ 3.1 วินาที เซ็ตตลิ่งไทม์ 3.1 วินาที ไม่มีโอเวอร์ชูต ควบคุมด้วย PI มีไรส์ไทม์ 4.0 วินาที เซ็ตตลิ่งไทม์ > 5.0 วินาที โอเวอร์ชูต > 20 %

ผลการทดลองในระบบจริงเปรียบเทียบระหว่างระบบที่ควบคุมด้วย MRAC เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที กับระบบที่ควบคุมด้วยตัวควบคุม PI ได้ถูกรวบรวมไว้ในตารางที่ 4.4

ตัวควบคุม	หมุน องศา	ไม่มีโหลด			เปิดหัวเครื่องกำเนิดไฟ			ปิดหัวเครื่องกำเนิดไฟ		
		t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)	t_R (วินาที)	t_S (วินาที)	M_P (%)
MRAC	50	0.9	0.9	0	0.9	1.0	1	0.9	0.9	0
	150	0.8	0.8	0	0.8	1.2	2	2.1	2.1	0
	300	0.8	1.0	2	0.9	1.3	3	3.1	3.1	0
PI	50	0.25	0.9	55	0.3	1.2	58	0.8	3.5	28
	150	0.45	1.9	39	0.55	> 5	45	2.15	> 5	32
	300	0.7	> 5	35	0.85	> 5	40	4.0	> 5	>20

t_R = ไรส์ไทม์ t_S = เซ็ตติงไทม์เมื่อค่าผิดพลาดต่ำกว่า 2% M_P = โอเวอร์ชูท

ตารางที่ 4.4 เปรียบเทียบช่วงเวลาผลตอบสนองและโอเวอร์ชูทเมื่อควบคุมด้วย MRAC และ PI เมื่อมอเตอร์หมุน 50 องศา 150 องศา และ 300 องศา

นอกจากผลการทดลองที่ได้เสนอมานี้แล้ว ได้ทำการทดลองในระบบจริงเพิ่มเติมเพื่อแสดงความสามารถของการควบคุมด้วยวิธี MRAC อีก โดยเพิ่มความถี่ธรรมชาติของโมเดลอ้างอิงเป็น 10 เรเดียน/วินาที และ 17 เรเดียน/วินาที และให้มอเตอร์หมุนเป็นมุม 50 องศา การทดลองกระทำแบบเดียวกับที่ผ่านมาแล้ว คือ ทดลองในขณะไม่มีโหลด และในขณะมีโหลด

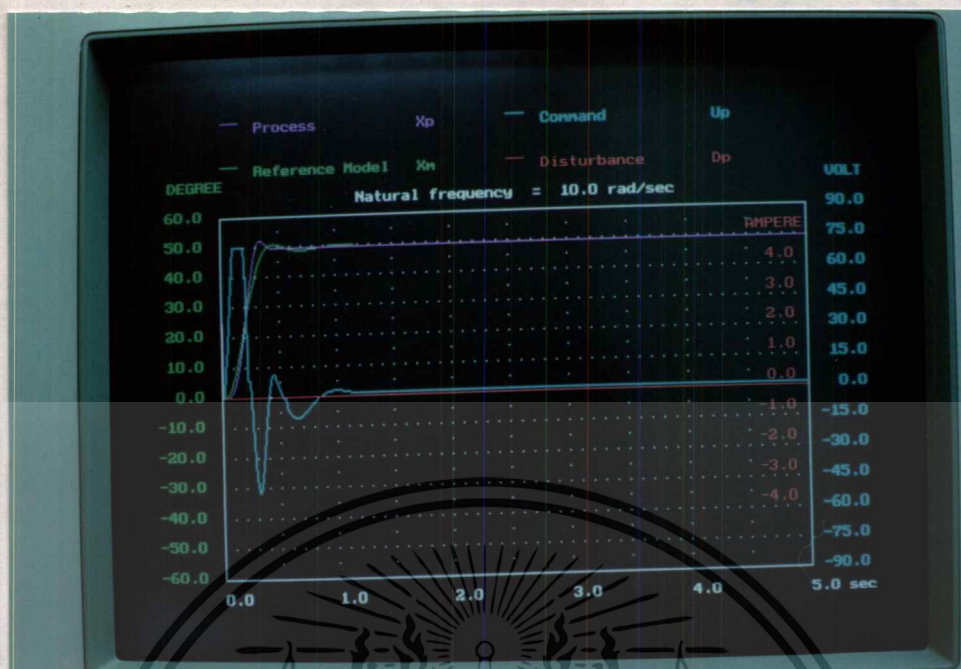


รูปที่ 4.37 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที
 มอเตอร์หมุนเบี่ยงมุม 50 องศา
 ไรส์ไทม์ 0.35 วินาที เซ็ตตลิงไทม์ 0.5 วินาที โอเวอร์ชูท 10 %



รูปที่ 4.38 แสดงผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที
 มอเตอร์หมุนเบี่ยงมุม 50 องศา
 ไรส์ไทม์ 0.2 วินาที เซ็ตตลิงไทม์ 1.05 วินาที โอเวอร์ชูท 25 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า
โมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา
โวลต์ใหม่ 0.3 วินาที เซ็ตคั้งใหม่ 0.4 วินาที โอเวอร์ชูต 10 %

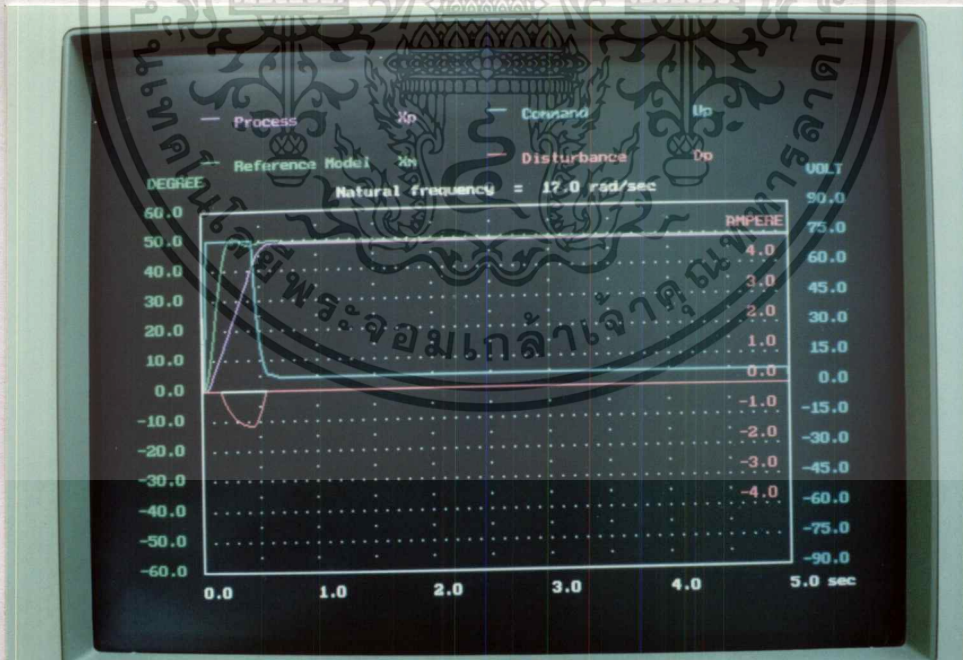


รูปที่ 4.40 แสดงผลตอบสนองเมื่อเปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า
โมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา
โวลต์ใหม่ 0.25 วินาที เซ็ตคั้งใหม่ 1.2 วินาที โอเวอร์ชูต 40 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 แสดงผลตอบสนองเมื่อเปิดหัวจ่ายไฟของเครื่องกำเนิดไฟฟ้า
โมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา
โวลต์ใหม่ 0.55 วินาที เซ็ตตั้งใหม่ 1.1 วินาที ไม่มีโอเวอร์ชูท



รูปที่ 4.42 แสดงผลตอบสนองเมื่อเปิดหัวจ่ายไฟของเครื่องกำเนิดไฟฟ้า
โมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา
โวลต์ใหม่ 0.55 วินาที เซ็ตตั้งใหม่ 0.6 วินาที ไม่มีโอเวอร์ชูท

ความถี่ ธรรมชาติ (เรเดียน /วินาที)	หมุม องศา	ไม่มีไหลด			เปิดชั่วเครื่องกำเนิดไฟ			ปิดชั่วเครื่องกำเนิดไฟ		
		t_R (วินาที)	t_S (วินาที)	M_p (%)	t_R (วินาที)	t_S (วินาที)	M_p (%)	t_R (วินาที)	t_S (วินาที)	M_p (%)
5	50	0.9	0.9	0	0.9	1.0	1	0.9	0.9	0
10	50	0.35	0.5	10	0.3	0.4	10	0.55	1.1	0
17	50	0.2	1.05	25	0.25	1.2	40	0.55	0.6	0

t_R = ไรส์ใหม่ t_S = เซ็ตตลิ่งใหม่เมื่อค่าผิดพลาดต่ำกว่า 2 % M_p = โอเวอร์ชูท

ตารางที่ 4.5 แสดงช่วง เวลาผลตอบสนองและ โอเวอร์ชูทเมื่อควบคุมด้วย MRAC

4.4. สรุป

จากผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ จะเห็นได้ว่าในขณะไม่มีไหลดผลตอบสนองของระบบควบคุมตำแหน่ง เมื่อใช้ตัวควบคุมทั้งสองแบบสามารถเข้าสู่คำสั่งอินพุตได้เหมือนกัน แต่การควบคุมด้วย MRAC มีข้อดีกว่าคือ สามารถเปลี่ยนแปลงความเร็วการเข้าสู่คำสั่งอินพุตได้ ซึ่งหมายถึงสามารถเปลี่ยนอัตราเร่งของผลตอบสนองได้โดยเปลี่ยนความถี่ธรรมชาติของโมเดลอ้างอิง ดังจะเห็นได้จากผลการทดลองซึ่งได้แสดงความถี่ธรรมชาติไว้ 2 ค่าคือ 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที จากตารางที่ 4.1 เมื่อใช้ความถี่ธรรมชาติ 3 เรเดียน/วินาที ผลตอบสนองจะมีไรส์ใหม่ 1.3 วินาที เซ็ตตลิ่งใหม่ 2.7 วินาที และเมื่อใช้ความถี่ธรรมชาติ 5 เรเดียน/วินาที ไรส์ใหม่จะลดเหลือเพียง 0.8 วินาที และเซ็ตตลิ่งใหม่เหลือเพียง 1.7 วินาที เมื่อทำการทดลองในขณะมีไหลด ซึ่งไหลดที่ใช้มี 2 แบบ คือ สเตปไหลดขนาด 25 % ของคำสั่งอินพุต และไหลดขนาด 25 % ของผลตอบสนอง จากตารางที่ 4.1 จะเห็นได้ว่าผลตอบสนองไม่เปลี่ยนแปลงไปจากในขณะไม่มีไหลดเลย คือ ผลตอบสนองจะมีไรส์ใหม่ 1.35 วินาที เซ็ตตลิ่งใหม่ 2.7 วินาที เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3

เรเดียน/วินาที และโรสใหม่ 0.85 วินาที เช็ตตลิ่งใหม่ 1.7 วินาที เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที นอกจากนี้แล้วในการทดลองทุกแบบผลตอบสนองจะมีโอเวอร์ชูทไม่เกิน 2 % การเปรียบเทียบกับระบบที่ควบคุมด้วยตัวควบคุม PI จะใช้ผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที จากตารางที่ 4.2 ในขณะที่ไม่มีโพลดผลตอบสนองของระบบที่ควบคุมด้วยตัวควบคุม PI จะมีโรสใหม่ 0.45 วินาที เช็ตตลิ่งใหม่ 2.1 วินาที และมีโอเวอร์ชูท 18 % เมื่อมีสเตปโพลดผลตอบสนองจะมีโรสใหม่เท่าเดิม แต่เช็ตตลิ่งใหม่เพิ่มขึ้นเป็น 2.4 วินาที และมีโอเวอร์ชูท 50 % เมื่อใช้โพลดที่เปลี่ยนแปลงตามผลตอบสนอง โรสใหม่จะลดลงเหลือ 0.35 วินาที แต่เช็ตตลิ่งใหม่เพิ่มขึ้นเป็น 4.2 วินาที และมีโอเวอร์ชูทสูงถึง 70 % จึงทำให้สามารถสรุปได้ว่า แม้ผลตอบสนองของระบบที่ควบคุมด้วยตัวควบคุม PI จะมีโรสใหม่เร็วกว่า แต่เช็ตตลิ่งใหม่จะล่าช้ากว่ามาก นั่นคือผลตอบสนองจะเข้าสู่คำสั่งอินพุตได้ช้ากว่าระบบที่ควบคุมด้วยวิธี MRAC และเมื่อมีโพลดมากกระทำเช็ตตลิ่งใหม่จะเพิ่มขึ้น นั่นหมายถึงผลตอบสนองต้องใช้เวลาเข้าสู่คำสั่งอินพุตนานขึ้น นอกจากนี้ผลตอบสนองของระบบที่ควบคุมด้วยตัวควบคุม PI จะมีโอเวอร์ชูทสูงถึง 18 % ในขณะที่ไม่มีโพลด และเพิ่มขึ้นเป็น 50 % และ 70 % เมื่อมีสเตปโพลด และโพลดที่เปลี่ยนแปลงตามผลตอบสนองมากกระทำตามลำดับ

เมื่อทำการทดลองในระบบจริงๆ ซึ่งทำการทดลองเหมือนกับการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ คือ ให้ความถี่ธรรมชาติของโมเดลอ้างอิงมีค่าเป็น 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที นอกจากนี้ยังได้กำหนดให้มอเตอร์หมุนเป็นมุม 50 องศา 150 องศา และ 300 องศา สำหรับการทดลองในขณะที่ไม่มีโพลด ระบบที่ควบคุมด้วย MRAC ยังคงมีผลตอบสนองต่อคำสั่งอินพุตคือ มีผลตอบสนองเหมือนกับผลตอบสนองของโมเดลอ้างอิง และผลตอบสนองสามารถเข้าสู่คำสั่งอินพุตได้ในเวลาเท่ากัน ไม่ว่าจะกำหนดให้มอเตอร์หมุนกี่องศาก็ตาม ดังจะเห็นได้จากตารางที่ 4.3 เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที ผลตอบสนองจะมีโรสใหม่ประมาณ 1.4 วินาที เช็ตตลิ่งใหม่ประมาณ 1.4 - 1.5 วินาที เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที จะมีโรสใหม่ประมาณ 0.8 - 0.9 วินาที และเช็ตตลิ่งใหม่ประมาณ 0.9 - 1.0 วินาที เมื่อเปรียบเทียบกับผลตอบสนองของระบบที่ควบคุมด้วยตัวควบคุม PI ความเร็วการเข้าสู่คำสั่งอินพุตจะเปลี่ยนแปลงไปตามจำนวนองศาที่มอเตอร์หมุน จากตารางที่ 4.4 ถ้าให้มอเตอร์หมุนเป็นมุม 50 องศา จะมีโรสใหม่ 0.25 วินาที เช็ตตลิ่งใหม่ 3.2 วินาที ถ้าให้มุมเป็นมุม 150 องศา จะมีโรสใหม่ 0.45 วินาที เช็ตตลิ่งใหม่ 0.9 วินาที และเมื่อให้มุมเป็นมุม 300 องศา จะมีโรสใหม่ 0.7 วินาที และเช็ตตลิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหม่มากกว่า 5 วินาที ดังนั้นจะเห็นได้ว่า ถ้าให้มอเตอร์หมุนน้อยผลตอบสนองจะเข้าสู่คำสั่ง อินพุทเร็ว ถ้าให้มอเตอร์หมุนมากผลตอบสนองจะเข้าสู่คำสั่งอินพุทช้า นอกจากนี้แล้วผลตอบสนองยังมีโอเวอร์ชูทค่อนข้างสูงประมาณ 35 % - 55 %

การทดลองในขณะมีโหลด โหลดที่ใช้ในการทดลองเกิดจากการต่อแกนของมอเตอร์ที่ทำการทดลอง เข้ากับแกนของมอเตอร์รุ่นเดียวกันอีกตัวหนึ่ง ดังนั้นมอเตอร์ตัวใหม่จะทำหน้าเป็น เครื่องกำเนิดไฟฟ้า ขนาดของโหลดเลือกโดยเปิดหรือปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า ขึ้นแรกทำการทดลองโดยเปิดขั้วจ่ายไฟเครื่องกำเนิดไฟฟ้า ระบบที่ควบคุมด้วย MRAC ยังคงมีผลตอบสนองดี เหมือนกับผลการทดลองในขณะไม่มีโหลด และสามารถนำผลตอบสนองเข้าสู่คำสั่งอินพุทได้ในเวลาเท่ากันไม่ว่ามอเตอร์จะหมุนกี่องศาก็ตาม ดังจะเห็นได้จากตารางที่ 4.3 ผลตอบสนองมีไรส์ไทม์ประมาณ 1.4 - 1.5 วินาที เซ็ตตลิ่งไทม์ประมาณ 1.4 - 1.5 วินาที เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที และเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที ผลตอบสนองมีไรส์ไทม์ประมาณ 0.8 - 0.9 วินาที เซ็ตตลิ่งไทม์ประมาณ 1.0 - 1.3 วินาที เช่นเดียวกับการทดลองในขณะไม่มีโหลด แม้ว่าตัวควบคุม PI จะสามารถนำผลตอบสนองเข้าสู่คำสั่งอินพุทได้ แต่เวลาการเข้าสู่คำสั่งอินพุทจะแปรตามองศาที่มอเตอร์ต้องหมุน และต้องใช้เวลามากกว่าในขณะไม่มีโหลด ดังจะเห็นได้จากตารางที่ 4.4 เมื่อให้มอเตอร์หมุน 50 องศา จะมีไรส์ไทม์ 0.3 วินาที เซ็ตตลิ่งไทม์ 1.2 วินาที เมื่อให้หมุน 150 องศา จะมีไรส์ไทม์ 0.55 วินาที เซ็ตตลิ่งไทม์มากกว่า 5 วินาที และเมื่อให้หมุน 300 องศา จะมีไรส์ไทม์ 0.85 วินาที เซ็ตตลิ่งไทม์มากกว่า 5 วินาที นอกจากนี้ยังมีโอเวอร์ชูทประมาณ 40 % - 58 % ในขณะที่การควบคุมด้วย MRAC จะมีโอเวอร์ชูทไม่เกิน 5 % เมื่อเพิ่มขนาดของโหลด โดยปิดขั้วจ่ายไฟของเครื่องกำเนิดไฟฟ้า การควบคุมด้วย MRAC ยังคงสามารถนำผลตอบสนองเข้าสู่คำสั่งอินพุทได้ดี แม้ว่าในบางครั้งจะช้ากว่าผลตอบสนองของโมเดลอ้างอิงก็ตาม ทั้งนี้เนื่องจากเป็นขีดความสามารถสูงสุดที่มอเตอร์สามารถทำได้ ดังจะสังเกตได้จากสัญญาณควบคุมของ MRAC (u_p) จะมีค่า 75 โวลต์ ซึ่งเป็นค่าโวลต์เตจสูงสุดของมอเตอร์ จากตารางที่ 4.3 เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 3 เรเดียน/วินาที มอเตอร์หมุนเป็นมุม 50 องศา และ 150 องศา ผลตอบสนองมีไรส์ไทม์ 1.5 วินาที เซ็ตตลิ่งไทม์ 1.6 วินาที ซึ่งใกล้เคียงกับผลการทดลองในขณะไม่มีโหลด เมื่อให้มอเตอร์หมุน 300 องศา ผลตอบสนองจะล่าช้ากว่าโมเดลอ้างอิง คือ มีไรส์ไทม์และเซ็ตตลิ่งไทม์ 3.75 วินาที เมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที และให้มอเตอร์หมุน 50 องศา จะมีไรส์ไทม์และเซ็ตตลิ่งไทม์ 0.9 วินาที ซึ่งเท่ากับผลการทดลองในขณะไม่มีโหลด เมื่อให้มอเตอร์หมุน 150 องศา และ 300

องศา ผลตอบสนองจะช้ากว่าโมเดลอ้างอิง คือ เมื่อให้หมุน 150 องศา จะมีโรสใหม่และเช็ดคดิ่งใหม่ 2.1 วินาที และเมื่อให้หมุน 300 องศา จะมีโรสใหม่และเช็ดคดิ่งใหม่ 3.1 วินาที ส่วนระบบที่ควบคุมด้วยตัวควบคุม PI ผลตอบสนองจะเข้าสู่คำสั่งอินพุตได้ช้ากว่าการควบคุมด้วย MRAC มาก จากตารางที่ 4.4 เมื่อให้มอเตอร์หมุน 50 องศา จะมีโรสใหม่ 0.8 วินาที เช็ดคดิ่งใหม่ 3.5 วินาที เมื่อให้มอเตอร์หมุน 150 องศา มีโรสใหม่ 2.15 วินาที เช็ดคดิ่งใหม่มากกว่า 5 วินาที เมื่อให้มอเตอร์หมุน 300 องศา มีโรสใหม่ 4 วินาที เช็ดคดิ่งใหม่มากกว่า 5 วินาที และผลตอบสนองยังมีโอเวอร์ชูทมากกว่า 20 % ในขณะที่การควบคุมด้วย MRAC จะมีโอเวอร์ชูทไม่เกิน 2 %

จากผลการทดลองทำให้สามารถสรุปได้ว่า ผลตอบสนองของระบบที่ควบคุมด้วย MRAC สามารถเข้าสู่คำสั่งอินพุตได้ในเวลาเท่ากันไม่แปรตามจำนวนองศาที่มอเตอร์ต้องหมุน แต่แปรตามความถี่ธรรมชาติของโมเดลอ้างอิง นอกจากนี้ในกรณีที่เกินขีดความสามารถของมอเตอร์ผลตอบสนองจะล่าช้าไปบ้าง ซึ่งเป็นข้อดีเหนือกว่าการควบคุมด้วยตัวควบคุม PI ที่ผลตอบสนองจะแปรตามจำนวนองศาที่มอเตอร์หมุน ผลตอบสนองของระบบที่ควบคุมด้วยตัวควบคุม PI จะใช้เวลามากกว่าระบบที่ควบคุมด้วย MRAC นอกจากนี้ผลตอบสนองของระบบที่ควบคุมด้วย MRAC จะมีโอเวอร์ชูทไม่เกิน 5 % ในขณะที่ตัวควบคุม PI จะทำให้เกิดโอเวอร์ชูทระหว่าง 20 % - 70 %

เมื่อนำผลการทดลองส่วนเพิ่มเติมมาเปรียบเทียบ ในขณะที่ไม่มีโหลดผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที จะมีผลตอบสนองที่เร็วกว่าเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที คือมีโรสใหม่ 0.35 วินาที เช็ดคดิ่งใหม่ 0.5 วินาที แต่จะมีโอเวอร์ชูทสูงถึง 10 % และเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที จะมีโรสใหม่ 0.2 วินาที แต่เช็ดคดิ่งใหม่เพิ่มขึ้นเป็น 1.05 วินาที นอกจากนี้ผลตอบสนองจะมีโอเวอร์ชูทสูงถึง 25 % และผลตอบสนองจะสั้นก่อนเข้าสู่คำสั่งอินพุต เมื่อทำการทดลองในขณะที่มีโหลดโดยเปิดขั้วเครื่องกำเนิดไฟฟ้า ผลตอบสนองจะมีโรสใหม่และเช็ดคดิ่งใหม่ใกล้เคียงกับในขณะที่ไม่มีโหลด แต่ขนาดโอเวอร์ชูทของผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 17 เรเดียน/วินาที จะเพิ่มขึ้นเป็น 40 % และผลตอบสนองยังคงสั้นก่อนเข้าสู่คำสั่งอินพุตเหมือนกับการทดลองในขณะที่ไม่มีโหลด เมื่อทำการทดลองโดยปิดขั้วเครื่องกำเนิดไฟฟ้า ผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 10 เรเดียน/วินาที และ 17 เรเดียน/วินาทีจะมีโรสใหม่เพิ่มขึ้นเป็น 0.55 วินาที เท่ากัน และมีเช็ดคดิ่งใหม่เท่ากับ 1.1 วินาที และ 0.6 วินาที ตามลำดับ

จากผลการทดลองที่นำมาเปรียบเทียบจะเห็นได้ว่า ไม่ควรใช้โมเดลอ้างอิงที่มีความถี่ธรรมชาติมากกว่า 17 เรเดียน/วินาที เพราะแม้ว่าผลตอบสนองจะมีไรส์ไทม์เร็วกว่า แต่เซตตลิงไทม์จะไม่เร็วกว่าผลตอบสนองเมื่อโมเดลอ้างอิงมีความถี่ธรรมชาติ 5 เรเดียน/วินาที นอกจากนี้ผลตอบสนองจะมีโอเวอร์ชูตสูงกว่า 10 % และผลตอบสนองจะสั้นก่อนเข้าสู่คำสั่งอินพุต

ความถี่ธรรมชาติของโมเดลอ้างอิงที่เหมาะสมในการใช้งาน คือ ความถี่ที่ต่ำกว่า 10 เรเดียน/วินาที เพราะผลตอบสนองของระบบควบคุมตำแหน่งจะมีเซตตลิงไทม์เร็ว โอเวอร์ชูตต่ำกว่า 10 % และผลตอบสนองจะไม่สั้นก่อนเข้าสู่คำสั่งอินพุต



บทที่ 5. บทสรุป

ในวิทยานิพนธ์นี้ได้เสนอการควบคุมตำแหน่งมอเตอร์กระแสตรงด้วยวิธี MRAC โดยเริ่มนำเสนอตั้งแต่แนวความคิดการควบคุมด้วยวิธี MRAC โครงสร้างของระบบควบคุมตำแหน่งมอเตอร์กระแสตรง การออกแบบโมเดลอ้างอิงด้วยวิธี ITAE ทฤษฎีพื้นฐานของ MRAC ทั้งชนิดต่อเนื่องและไม่ต่อเนื่อง การออกแบบคอนโทรลอัลกอริทึมของ MRAC ทั้งชนิดต่อเนื่องและไม่ต่อเนื่อง โดยวิธีที่ 2 ของไลฟูนอฟ การแปลงระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง โดยวิธีโบลีเนียร์ทรานส์ฟอร์มเมชัน การสร้างตัวแปรสถานะของระบบควบคุมตำแหน่งจากสัญญาณควบคุมและจำนวนองศาที่มอเตอร์หมุนโดยตัวสังเกตสถานะ โครงสร้างและส่วนประกอบต่างๆ ของวงจรรีบชนิด PWM คอนเวอร์เตอร์ ซึ่งประกอบด้วยวงจรรีเลย์สัญญาณสามเหลี่ยม วงจรเปรียบเทียบ วงจรรับเบส และวงจรถูก้าลัง การเชื่อมต่อระหว่างคอนโทรลอัลกอริทึมในคอมพิวเตอร์และระบบควบคุมตำแหน่งจริงโดยผ่านวงจรรีเลย์เฟส ซึ่งประกอบด้วย วงจรดีโค๊ดเดอร์ วงจรเอาต์พุต และวงจรรีเลย์เฟส ซึ่งวงจรรีเลย์เฟสจะแปลงสัญญาณควบคุมจากคอนโทรลอัลกอริทึมจากดิจิตอลให้เป็นอนาล็อกและส่งต่อไปยังวงจรรีเลย์ PWM คอนเวอร์เตอร์ วงจรรีเลย์ประกอบด้วย 2 ส่วนคือ วงจรนับความถี่ ซึ่งจะนับสัญญาณจากพัลส์เอ็นโค๊ดเดอร์ทำให้ทราบตำแหน่งของมอเตอร์ และวงจรวัดกระแสจะรับสัญญาณจาก DCCT และแปลงเป็นดิจิตอลทำให้คอนโทรลอัลกอริทึมทราบค่ากระแสที่มอเตอร์ใช้

ผลการทดลองได้เสนอไว้ในบทที่ 4. ซึ่งมีทั้งผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ และผลการทดลองในระบบจริง โดยทำการทดลองทั้งในขณะไม่มีโหลด มีโหลด และเปรียบเทียบกับตัวควบคุม PI ซึ่งตัวควบคุม PI ถูกออกแบบโดยวิธีซิมเมตริกคอลลอปติวมัมจากผลการทดลองโดยจำลองผลตอบสนองด้วยคอมพิวเตอร์ ในขณะไม่มีโหลดการควบคุมทั้งสองวิธีสามารถนำผลตอบสนองเข้าสู่คำสั่งอินพุตได้เหมือนกัน แต่การควบคุมด้วยวิธี MRAC มีข้อดีกว่าคือ สามารถเปลี่ยนแปลงอัตราเร่งของผลตอบสนองได้ โดยเปลี่ยนความถี่ธรรมชาติของโมเดลอ้างอิงซึ่งในการทดลองได้เสนอไว้ 2 ค่าคือ 3 เรเดียน/วินาที และ 5 เรเดียน/วินาที เมื่อทดลองเปรียบเทียบในขณะมีโหลด ระบบที่ควบคุมด้วย MRAC ยังคงมีผลตอบสนองดีเหมือนกับการไม่มีโหลด ส่วนระบบที่ควบคุมด้วยตัวควบคุม PI จะมีผลตอบสนองเบี่ยงเบนไปจากคำสั่งอินพุตก่อนที่จะกลับเข้าสู่คำสั่งอินพุต เมื่อทำการทดลองในระบบจริงจะสั่งให้มอเตอร์หมุนเป็นมุมเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

50 องศา 150 องศา และ 300 องศา การทดลองในขณะไม่มีโหลด ระบบที่ควบคุมด้วย MRAC สามารถนำผลตอบสนองเข้าสู่คำสั่งอินพุตได้ดี และสามารถเข้าสู่คำสั่งอินพุตได้ในเวลาเท่ากันไม่ว่าจะสั่งใหม่มอเตอร์หมุนเป็นมุมกึ่งศักราชก็ตาม ซึ่งเป็นข้อดีอีกข้อหนึ่งของการควบคุมโดยวิธีนี้ ในขณะที่การควบคุมด้วยตัวควบคุม PI การเข้าสู่คำสั่งอินพุตจะแปรตามจำนวนองศาที่มอเตอร์ต้องหมุน และเมื่อทำการทดลองในขณะมีโหลด การควบคุมด้วยวิธี MRAC ยังคงนำผลตอบสนองเข้าสู่คำสั่งอินพุตได้ดีเหมือนในขณะไม่มีโหลด แม้ว่าในบางครั้งจะล่าช้าไปบ้างทั้งนี้เนื่องจากขนาดของโหลดเป็นโหลดสูงสุดของมอเตอร์ ดังจะสังเกตได้จากสัญญาณควบคุมและกระแสที่มอเตอร์ใช้จะเป็นค่าสูงสุดของมอเตอร์ ในขณะที่การเข้าสู่คำสั่งอินพุตของระบบที่ควบคุมด้วยตัวควบคุม PI จะแปรตามขนาดของโหลด จำนวนองศาที่มอเตอร์ต้องหมุน และจะล่าช้ากว่าการควบคุมด้วย MRAC มาก

นอกจากนี้ยังได้ทำการทดลองเพิ่มเติมเพื่อหาความสามารถสูงสุดของการควบคุมด้วย MRAC จากผลการทดลองทำให้สามารถสรุปว่า ไม่ควรใช้โมเดลอ้างอิงที่มีความถี่ธรรมชาติเกิน 17 เรเดียน/วินาที เพราะจะทำให้ผลตอบสนองของระบบควบคุมตำแหน่งมีโอเวอร์ชูมากกว่า 10 % และผลตอบสนองจะล่าช้าก่อนเข้าสู่คำสั่งอินพุต

จากผลการทดลองจะสามารถสรุปข้อดีของการควบคุมด้วย MRAC ที่เหนือกว่าการควบคุมด้วยตัวควบคุม PI คือ การควบคุมด้วย MRAC สามารถเปลี่ยนแปลงอัตราเร่งของผลตอบสนองได้โดยเปลี่ยนความถี่ธรรมชาติของโมเดลอ้างอิง ผลตอบสนองของระบบควบคุมตำแหน่งสามารถติดตามผลตอบสนองของโมเดลอ้างอิงได้แม้ว่าจะมีโหลดมากกระทำก็ตาม นอกจากนี้เวลาเข้าสู่คำสั่งอินพุตจะไม่แปรตามโหลดและจำนวนองศาที่มอเตอร์ต้องหมุน

ในส่วนสุดท้ายของวิทยานิพนธ์ได้เสนอทฤษฎีต่างๆ ที่กล่าวถึงในวิทยานิพนธ์ ได้แก่ การวิเคราะห์เสถียรภาพของไลฟูนอฟ กฎเกณฑ์ของซิลเวสเตอร์สำหรับเมตริกสมมาตรของเลขจำนวนจริง ทฤษฎีการล่มสัญญาณ การสามารถสังเกตได้ การออกแบบตัวควบคุม PI โดยวิธีหิมนเมตริกคอลออปติมิซึม โปรแกรมจำลองผลตอบสนอง และควบคุมตำแหน่งโดย MRAC ขนาดพิกัดของมอเตอร์ และผลงานวิจัยที่ได้รับการเผยแพร่ในระหว่างศึกษาระดับปริญญาโท

กิตติกรรมประกาศ

ผู้เขียนขอขอบพระคุณอย่างสูงต่อ อาจารย์ ดร. โยธิน เปรมปราณีรัชต์ อาจารย์ที่ปรึกษา
 ที่ให้แนวความคิดในงานวิจัยเรื่องนี้ อีกทั้งท่านยังได้ให้คำปรึกษาและทุ่มเทสติปัญญาความคิดความ
 สามารถอย่างเต็มกำลัง ให้กับผู้เขียนมาโดยตลอดตั้งแต่ต้นจนจบการศึกษา และขอขอบพระคุณ
 อาจารย์ วันชัย วีรรุจา อาจารย์ วรศักดิ์ จิตรภักดี และอาจารย์ เกียรติวรรณ ทรงสัตย์
 ที่ได้ให้คำปรึกษาวิธีแก้ไขปัญหาต่างๆ จนวิทยานิพนธ์ฉบับนี้สมบูรณ์และมีคุณภาพมากขึ้น นอกจากนี้
 นี้ขอขอบพระคุณอาจารย์ และเจ้าหน้าที่ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ให้ความเอื้อเฟื้อจนวิทยานิพนธ์
 ฉบับนี้สำเร็จสมบูรณ์



เอกสารอ้างอิง

- [1] Katsuhiko Ogata, "Modern Control Engineering", Printice - Hall of India, New Delhi, 1985
- [2] Katsuhiko Ogata, "Discrete - Time Control Systems", Printice - Hall International Inc., Englewood Cliffs, New Jersey, 1987
- [3] Stephen P. Banks, "Control System Engineering", Printice - Hall International, 1986
- [4] Friedrich Frohr and Fritz Orthenburger, "Introduction to Electronic Control Engineering", Siemens Aktiengesellschaft Heyden & Son Ltd., 1981
- [5] James H. Aylor, Robert L. Ramcy, Gerald Cook, "Design and Application of a Microprocessor PID Predictor Controller", IEEE Transaction on Industrial Electronics and Control Instrumentation, Vol. IECI - 27, No. 3, August 1980
- [6] Kenji Tamaki, Kiyoshi Ohishi, Kouhei Ohnishi, Kunio Miyachi, "Microprocessor - Based Robust Control of a Dc Servo Motor", IEEE Control System Magazine, October 1986
- [7] C. J. Harris and S. A. Billings, "Self - Tuning and Adaptive Control : Theory and Application", Peter Peregrinus Ltd., England, 1981
- [8] G. Ambrosino, G. Celentano, F. Garofalo, "Variable Structure Model Reference Adaptive Control Systems", Int. J. Control, Vol. 39, No. 6, 1984, pp 1339 - 1349
- [9] Hans Bulter, Ger Handered and Job Van Amerongen, "Model Reference Adaptive Control of a Direct Drive Dc Motor", IEEE Control System Magazine, January 1989, pp 80 - 84

- [10] Karl Johan Astrom and Jorn Wittenmark, "Adaptive Control", Addison - Wesley Publishing Company, 1989
- [11] V. V. Calam, " Adaptive Control Systems Techniques and Applications", Marcel Dekker, Taiwan, 1989
- [12] Tadashi Egami, Hideaki Morita, Takeshi Tsuchiya, " Efficiency Optimized Model Reference Adaptive Control System for a Dc Motor", IEEE Transaction on Industrial Electronics, Vol. 37, No. 1, February 1990
- [13] Yasuhiko Dote, " Application of Modern Control Techniques to Motor Control", Proceeding of the IEEE, Vol. 76, No. 4, April 1988
- [14] Takashi Yahagi, " A Method for Adaptive Control Using a Minicomputer", IEEE Transaction on Industrial Electronics and Control Instrumentation, Vol. IECI - 26, No. 1, February 1979
- [15] Barry W. Johnson, James H. Aylor, " Design of an Adaptive Controller for Microcomputer Implementation", IEEE Transaction on Industrial Electronics, Vol. IE - 33, No. 1, February 1986
- [16] D. Grant Fisher and Dale E. Seborg, " Multivariable Coputer Control a Case Study ", North Holland Publishing Company, 1976
- [17] Louis A. Dessaint, Bernard J. Hebert, Hong Le-Huy, Gianni Cavuoti, " A DSP - Based Adaptive Controller for a Smooth Positioning System", IEEE Transaction on Industrial Electronics, Vol. 37, No. 5, October 1990
- [18] Shankar Sastry, Marc Bodson, " Adaptive Control Stability", Convergence, and Robustness, Printice - Hall International Inc., Englewood Cliffs, New Jersey, 1989
- [19] Ned Mohan, Tore M. Undeland, William P. Robbins, " Power Electronics Converters, Application and Design", John Wiley & Son, Singapore, 1989

- [20] J. G. Kassakian, D. H. Naunin, " Design of Control Systems for DC Drives", Springer - Verlag Berlin Heidelberg, 1990
- [21] Takashi Kenjo, " Power Electronics for the Microprocessor Age", Oxford, 1990
- [22] "Dc Motor Speed Controls Servo Systems", Electro - Craft Corporation, England
- [23] Brian W. Kernighan, Dennis M. Ritchie, " The C Programming Language", Printice - Hall Inc., Englewood Cliffs, New Jersey
- [24] Herbert Schildt, " Advanced Turbo C", Borland - Osborne / McGraw - Hill, California, 1987
- [25] " Turbo C Reference Guide, User Guide ", Borland International Inc., California, 1987
- [26] โยธิน เปรมปราณีรัชต์, " ระบบเซอร์โวและอิเล็กทรอนิกส์คอนโทรลมอเตอร์ ", ตำราชุดวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2533

ภาคผนวก ก.

การวิเคราะห์เสถียรภาพของไลปูนอฟ (Liapunov's Stability Analysis)

การวิเคราะห์เสถียรภาพโดยวิธีของไลปูนอฟ [1,2] เป็นวิธีที่สำคัญสำหรับการวิเคราะห์เสถียรภาพของระบบควบคุม การวิเคราะห์เสถียรภาพโดยวิธีของไลปูนอฟมี 2 วิธี เรียกว่า วิธีที่ 1 และวิธีที่ 2 โดยวิธีที่ 1 จะต้องหาคำตอบของสมการดิฟเฟอเรนเชียล (differential equation) สำหรับระบบต่อเนื่อง หรือสมการดิฟเฟอเรนซ์ (difference equation) สำหรับระบบไม่ต่อเนื่อง และใช้คำตอบสำหรับการวิเคราะห์เสถียรภาพ ในทางตรงข้ามวิธีที่ 2 ไม่จำเป็นต้องหาคำตอบ แต่จะพิจารณาจากสมการสถานะ ดังนั้นวิธีที่ 2 จึงมีประโยชน์อย่างมากในทางปฏิบัติ

แม้ว่าจะมีหลาย ๆ วิธีที่มีประสิทธิภาพสำหรับการวิเคราะห์เสถียรภาพของระบบควบคุม เช่น กฎเกณฑ์เสถียรภาพของจูรี่ (Jury's stability criterion) และกฎเกณฑ์เสถียรภาพของรูธ-ฮิวริวิต (Routh - Hurwitz's stability criterion) แต่วิธีเหล่านี้ถูกจำกัดให้ใช้ได้เฉพาะกับระบบเชิงเส้นที่ไม่แปรตามเวลา (linear time-invariant system) เท่านั้น แต่วิธีที่ 2 ของไลปูนอฟไม่ได้ถูกจำกัดไว้เฉพาะระบบเชิงเส้นที่ไม่แปรตามเวลาเท่านั้น แต่วิธีนี้สามารถนำไปใช้กับระบบใดก็ได้ ทั้งระบบเชิงเส้น ระบบไม่เชิงเส้น (nonlinear) ทั้งที่ไม่แปรตามเวลา และแปรตามเวลา (time-varying) วิธีที่ 2 ของไลปูนอฟถูกเรียกอีกชื่อหนึ่งว่า วิธีตรงของไลปูนอฟ (direct method of Liapunov)

จากทฤษฎีดั้งเดิมของระบบกลไก (classical theory of mechanics) เราทราบว่าระบบที่สั่น (vibratory system) จะมีเสถียรภาพถ้าพลังงานรวมของระบบลดลงตลอดเวลาจนกระทั่งถึงสถานะสมดุล (equilibrium state) วิธีที่ 2 ของไลปูนอฟก็มาจากพื้นฐานเดียวกันนี้ นั่นคือ ถ้าระบบใดมีเสถียรภาพ พลังงานที่สะสมไว้ในระบบนั้นจะลดลงอย่างต่อเนื่องเมื่อเวลาเพิ่มขึ้นจนมีค่าน้อยที่สุดที่สถานะสมดุล และไลปูนอฟได้เสนอสมการพลังงานขึ้น เรียกว่า สมการไลปูนอฟ (Liapunov equation) ซึ่งสามารถใช้งานได้มากกว่าเป็นสมการของพลังงานเพียงเดียว ในความเป็นจริงแล้วสมการสเกลาร์ใดๆ ที่เป็นไปตามทฤษฎีเสถียรภาพของไลปูนอฟ ซึ่งจะกล่าวถึงต่อไปสามารถถือว่าเป็นสมการไลปูนอฟได้

ก่อนที่จะพิจารณาสมการไลปูนอฟ จำเป็นต้องเข้าใจความหมายของคำบางคำก่อน

การมีความหมายเป็นบวกของสเกลาร์ฟังก์ชัน (positive definiteness of scalar function) สเกลาร์ฟังก์ชัน $V(x)$ จะถูกเรียกว่ามีความหมายเป็นบวกในช่วง Ω ซึ่งรวมทั้งจุดกำเนิดของสเปซสเตต (state space) ถ้า $V(x) > 0$ สำหรับสถานะ x (state x) ทั้งหมดที่ไม่เป็น 0 ในช่วง Ω และ $V(0) = 0$

การมีความหมายเป็นลบของสเกลาร์ฟังก์ชัน (negative definiteness of scalar function) สเกลาร์ฟังก์ชัน $V(x)$ จะถูกเรียกว่ามีความหมายเป็นลบ ถ้า $-V(x)$ มีความหมายเป็นบวก

การมีความหมายกึ่งเป็นบวกของสเกลาร์ฟังก์ชัน (positive semidefiniteness of scalar function) สเกลาร์ฟังก์ชัน $V(x)$ จะถูกเรียกว่ามีความหมายกึ่งเป็นบวก ถ้าฟังก์ชันนั้นเป็นบวกสำหรับทุกสถานะ x ในช่วง Ω ยกเว้นที่จุดกำเนิด และที่สถานะ x อื่นๆ มันเป็น 0

การมีความหมายกึ่งเป็นลบของสเกลาร์ฟังก์ชัน (negative semidefiniteness of scalar function) สเกลาร์ฟังก์ชัน $V(x)$ จะถูกเรียกว่ามีความหมายกึ่งเป็นลบ ถ้า $-V(x)$ มีความหมายกึ่งเป็นบวก

การไม่มีความหมายของสเกลาร์ฟังก์ชัน (indefiniteness of scalar function) สเกลาร์ฟังก์ชัน $V(x)$ จะถูกเรียกว่าไม่มีความหมาย ถ้าฟังก์ชันนั้นทั้งมีความหมายเป็นบวกและความหมายเป็นลบในช่วง Ω ไม่ว่าช่วง Ω นั้นจะมีขนาดเล็กเท่าใดก็ตาม

สมการไลปูนอฟ (Liapunov function) คือ สเกลาร์ฟังก์ชันที่มีความหมายเป็นบวกในช่วง และมีค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลาที่มีความหมายเป็นลบหรือกึ่งเป็นลบ

ทฤษฎีเสถียรภาพของไลปูนอฟ (Liapunov Theorem on Asymptotic Stability)

ทฤษฎีที่ 1

สมมติระบบที่พิจารณาโดย $\dot{x} = f(x)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง $f(0) = 0$

ถ้ามีสเกลาร์ฟังก์ชัน $V(x)$ ซึ่งมีค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลา และเป็นไปตามเงื่อนไขต่อไปนี้

1. $V(x)$ มีความหมายเป็นบวก
2. $V'(x)$ มีความหมายเป็นลบ

ดังนั้นระบบนั้นจะมีเสถียรภาพแบบอสมโทติก (asymptotically stable)

ทฤษฎีที่ 2

สมมติระบบที่พิจารณาโดย $\dot{x} = f(x)$

ซึ่ง $f(0) = 0$

ถ้ามีสเกลาร์ฟังก์ชัน $V(x)$ ซึ่งมีค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลา และเป็นไปตามเงื่อนไขต่อไปนี้

1. $V(x)$ มีความหมายเป็นบวก
2. $V'(x)$ มีความหมายกึ่งเป็นลบ

ดังนั้นระบบนั้นจะมีเสถียรภาพ

ทฤษฎีที่ 3

สมมติระบบที่พิจารณาโดย $\dot{x} = f(x)$

ซึ่ง $f(0) = 0$

ถ้ามีสเกลาร์ฟังก์ชัน $W(x)$ ซึ่งมีค่าดิฟเฟอเรนเชียลอันดับหนึ่งเทียบกับเวลา และเป็นไปตามเงื่อนไขต่อไปนี้

1. $W(x)$ มีความหมายเป็นบวกในบางช่วงรอบจุดกำเนิด
2. $W'(x)$ มีความหมายเป็นบวกในช่วงเดียวกับเงื่อนไขที่ 1.

ดังนั้นระบบนั้นจะไม่มีเสถียรภาพ

ทฤษฎีที่ 4

สมมติระบบไม่ต่อเนื่องซึ่งพิจารณาโดย $x(k+1) = f(x(k))$

ซึ่ง x เป็น n เวกเตอร์

และ $f(x)$ เป็น n vector และ $f(0) = 0$

สมมติมีสเกลาร์ฟังก์ชันและเป็นไปตามเงื่อนไขต่อไปนี้

1. $V(x) > 0$ เมื่อ $x \neq 0$

2. $\Delta V(x) < 0$ เมื่อ $x \neq 0$ ซึ่ง

$$\begin{aligned}\Delta V(x(k)) &= V(x(k+1)) - V(x(k)) \\ &= V(f(x(k))) - V(x(k))\end{aligned}$$

3. $V(0) = 0$

4. $V(x) \rightarrow \infty$ เมื่อ $\|x\| \rightarrow \infty$

ดังนั้น ระบบนั้นจะมีเสถียรภาพแบบอิมโทติก (asymptotically stable) และ $V(x)$ เป็นสมการไลปูนอว

การวิเคราะห์เสถียรภาพโดยวิธีไลปูนอวของระบบต่อเนื่องไม่แปรตามเวลาเชิงเส้น (Liapunov stability analysis of linear time-invariant continuous time system)

สมมติระบบซึ่งถูกพิจารณาโดย $\dot{x} = Ax$

เมื่อ x เป็นเวกเตอร์สถานะ (state vector, n vector)

และ A เป็น $n \times n$ เมตริกซ์คงที่และเป็นนอนอิงกูลาร์

ดังนั้นจะมีสภาวะสมดุลที่จุดกำเนิด ($x = 0$) เพียงที่เดียว เลือกสมการไลปูนอวเป็น

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

เมื่อ P เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงซึ่งมีความหมายเป็นบวก และค่าดีเทอร์มิแนนต์เทียบกับเวลาอันดับหนึ่งของ $V(x)$ คือ

$$\begin{aligned}
 V'(x) &= (x^T)'Px + x^TPx' \\
 &= (Ax)^TPx + x^TPAx \\
 &= x^TA^TPx + x^TPAx \\
 &= x^T(A^TP + PA)x
 \end{aligned}$$

เนื่องจาก $V(x)$ ถูกเลือกให้มีความหมายเป็นบวก ถ้าต้องการให้ระบบมีเสถียรภาพแบบอสมโทติก $V'(x)$ ต้องมีความหมายเป็นลบ

$$V'(x) = x^T[-Q]x$$

ซึ่ง $Q = A^TP + PA$ มีความหมายเป็นบวก

การทดสอบว่าเมตริกซ์ขนาด $n \times n$ มีความหมายเป็นบวกหรือไม่ ทดสอบโดยใช้กฎเกณฑ์ซิลเวสเตอร์ (Sylvester's criterion)

การวิเคราะห์เสถียรภาพโดยวิธีไลฟูนอฟของระบบไม่ต่อเนื่อง ไม่แปรตามเวลาเชิงเส้น (Liapunov stability analysis of linear time-invariant discrete time system)

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

ซึ่ง x เป็นเวกเตอร์สถานะ (state vector, n vector)

และ G เป็น $n \times n$ เมตริกซ์คงที่และเป็นนอนซิงกูลาร์

ดังนั้น ที่จุดกำเนิด ($x = 0$) จะเป็นสภาวะสมดุล เลือกสมการไลฟูนอฟ

$$V(x(k)) = x(k)^TPx(k)$$

ซึ่ง P เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงซึ่งมีความหมายเป็นบวก และ

$$\begin{aligned}
\Delta V(x(k)) &= V(x(k+1)) - V(x(k)) \\
&= (x(k+1))^T P x(k+1) - x(k)^T P x(k) \\
&= (Gx(k))^T P G x(k) - x(k)^T P x(k) \\
&= x(k)^T G^T P G x(k) - x(k)^T P x(k) \\
&= x(k)^T (G^T P G - P) x(k)
\end{aligned}$$

เนื่องจาก $V(x(k))$ ถูกเลือกให้มีความหมายเป็นบวก ถ้าต้องการให้ระบบมีเสถียรภาพแบบอซีมโทติก $\Delta V(x(k))$ ต้องมีความหมายเป็นลบ นั่นคือ

$$\Delta V(x(k)) = x(k)^T [-Q] x(k)$$

ซึ่ง

$$Q = G^T P G - P \quad \text{มีความหมายเป็นบวก}$$

ภาคผนวก ข.

กฎเกณฑ์ซิลเวสเตอร์สำหรับเมตริกซ์สมมาตรของเลขจำนวนจริง
(Sylvester's Criterion for Real Symmetrical Matrix)

เงื่อนไขที่จำเป็นและเพียงพอสำหรับการมีความหมายเป็นบวก การมีความหมายเป็นลบ การมีความหมายกึ่งเป็นบวก และการมีความหมายกึ่งเป็นลบของเมตริกซ์สมมาตรของเลขจำนวนจริง ได้ถูกให้ไว้โดย J.J. Sylvester เรียกว่า กฎเกณฑ์ซิลเวสเตอร์ [1,2]

เนื่องจากในวิทยานิพนธ์นี้ อ้างถึงการมีความหมายเป็นบวก และการมีความหมายกึ่งเป็นบวกของเมตริกซ์สมมาตรของเลขจำนวนจริงเท่านั้น ดังนั้นในที่นี้จะกล่าวถึงเพียง 2 ความหมายเท่านั้น

กฎเกณฑ์ซิลเวสเตอร์สำหรับการมีความหมายเป็นบวกของเมตริกซ์สมมาตรของเลขจำนวนจริง

สมมติ A เป็นเมตริกซ์สมมาตรของเลขจำนวนจริงขนาด $n \times n$ เงื่อนไขที่จำเป็นและเพียงพอสำหรับการมีความหมายเป็นบวก คือ ค่าดีเทอร์มิแนนต์ (determinant) ของเมตริกซ์ A และของเมตริกซ์ย่อยที่อยู่ติดกันขนาด $k \times k$ มีค่ามากกว่า 0 โดยที่ $k = 1, 2, \dots, n-1$ และ เมตริกซ์ย่อยเริ่มจากมุมบนซ้ายของเมตริกซ์ A

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \dots & \dots & a_{nn} \end{bmatrix}, \quad a_{ij} = a_{ji}$$

$$a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} > 0, \quad \dots, \quad |A| > 0$$

กฎเกณฑ์ที่ลเวสเตอร์สำหรับการมีความหมายกึ่งเป็นบวกของเมตริกซ์สมมาตรของเลขจำนวนจริง
เงื่อนไขที่จำเป็นและเพียงพอสำหรับการมีความหมายกึ่งเป็นบวกคือ เมตริกซ์ A ต้องเป็น
ซิงกูลาร์ และค่าดีเทอร์มิแนนต์ของเมตริกซ์ย่อยต้องไม่เป็นลบ

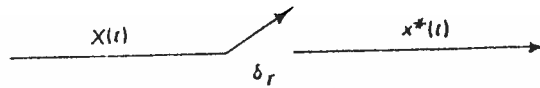
$$a_{11} \geq 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \geq 0, \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \geq 0, \dots, \quad |A| = 0$$



ภาคผนวก ค.

ทฤษฎีการสุ่มสัญญาณ (Sampling Theorem)

การสุ่มสัญญาณต่อเนื่อง เป็นสัญญาณไม่ต่อเนื่อง ความถี่ของการสุ่มจะต้องมีค่าสูงเพียงพอ เพื่อสัญญาณไม่ต่อเนื่องที่ได้จะสามารถรักษาคุณสมบัติของสัญญาณต่อเนื่องดั้งเดิมได้ [2]

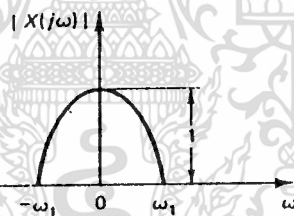


รูปที่ ค.1 แสดงการสุ่มสัญญาณ

$x(t)$ คือ สัญญาณต่อเนื่องเดิม

$x^*(t)$ คือ สัญญาณที่ถูกสุ่ม

สมมติสเปกตรัม (spectrum) ของสัญญาณต่อเนื่องแสดงดังรูป



รูปที่ ค.2 แสดงสเปกตรัมของสัญญาณต่อเนื่อง

W_1 คือ ความถี่สูงสุดของสัญญาณต่อเนื่อง

ทฤษฎีการสุ่มสัญญาณกล่าวว่า จะสามารถสร้างสัญญาณต่อเนื่องเดิม ได้จากสัญญาณที่ถูกสุ่ม ถ้า $W_s > 2W_1$ เมื่อ W_s คือความถี่ของการสุ่มสัญญาณ เพื่อพิสูจน์ทฤษฎีการสุ่มสัญญาณ เราจะต้องหาสเปกตรัมความถี่ (frequency spectrum) ของสัญญาณที่ถูกสุ่ม ($x^*(t)$) และ Laplace transform ของสัญญาณที่ถูกสุ่ม คือ

$$X^*(s) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(s + jW_s k) \quad (\text{ค.1})$$

แทนค่า s ด้วย $j\omega$

$$\begin{aligned} X^*(j\omega) &= \frac{1}{T} \sum_{k=-\infty}^{\infty} X(j\omega + j\omega_s k) \\ &= \dots + X(j(\omega - \omega_s))/T + X(j\omega)/T + X(j(\omega + \omega_s))/T \\ &\quad + \dots \end{aligned} \quad (\text{ค.2})$$

เนื่องจาก $X^*(s)$ จะเกิดขึ้นซ้ำกันทุกๆ ω_s ซึ่งสามารถพิสูจน์ได้จาก

$$X^*(s) = \frac{1}{T} \sum_{h=-\infty}^{\infty} X(s + j\omega_s h) \quad (\text{ค.3})$$

$$X^*(s + j\omega_s k) = \frac{1}{T} \sum_{h=-\infty}^{\infty} X(s + j\omega_s k + j\omega_s h) \quad (\text{ค.4})$$

ให้ $k + h = m$ ดังนั้นจากสมการสุดท้ายจะได้

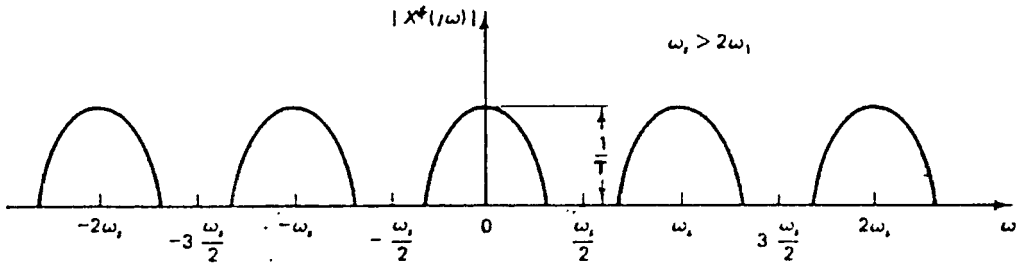
$$X^*(s + j\omega_s k) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X(s + j\omega_s m) = X^*(s) \quad (\text{ค.5})$$

จะได้

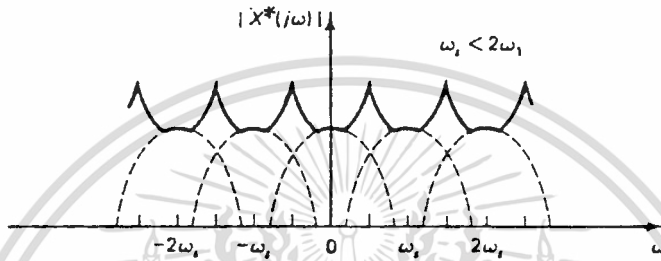
$$X^*(s) = X^*(s \pm j\omega_s k) \quad ; \quad k = 0, 1, 2, \dots \quad (\text{ค.6})$$

ซึ่งแสดงว่า $X^*(s)$ จะเกิดขึ้นซ้ำกันทุกๆ ω_s นั้นแสดงว่า ถ้า $X(s)$ มีโหลที่ $s = s_1$ ทำให้ $X^*(s)$ จะมีโหลที่ $s = s_1 \pm j\omega_s k$ โดยที่ $k = 0, 1, 2, \dots$

รูป ค.3 เป็นกราฟอธิบายทฤษฎีการสุ่ม โดยวาดกราฟระหว่างสเปกตรัมสัญญาณที่ถูกสุ่ม $|X^*(j\omega)|$ และ ω และแบ่งออกเป็น 2 กรณี คือ $\omega_s > 2\omega_1$ และ $\omega_s < 2\omega_1$ กราฟแต่ละรูปประกอบด้วย $|X(j\omega)|/T$ ซ้ำกันไปทุกๆ $\omega_s = 2\pi/T$ ถ้า $\omega_s < 2\omega_1$ สเปกตรัมของ $|X(j\omega)|$ จะเกิดการซ้อนทับกันทำให้ไม่เห็นรูปร่างจริงของ $|X(j\omega)|$ ส่วนในกรณี $\omega_s > 2\omega_1$ จะไม่เกิดการซ้อนทับกัน และจะเกิดสเปกตรัมขึ้นทุกๆ ω_s



(ก)



(ข)

รูปที่ ค.3 แสดงสเปกตรัมของสัญญาณที่ถูกสุ่ม

รูปที่ ค.3.1 เมื่อ $\omega_s > 2\omega_1$

รูปที่ ค.3.2 เมื่อ $\omega_s < 2\omega_1$

แม้ว่าในทางทฤษฎีจะต้องการเพียง $\omega_s > 2\omega_1$ แต่ในทางปฏิบัติมักจะเลือก ω_s มีค่าสูงกว่านี้มาก

ภาคผนวก ง.

การสามารถสังเกตได้ (Observability)

สมมติระบบถูกพิจารณาโดย [1,2]

$$X(k+1) = AX(k) + BU(k) \quad (ง.1)$$

$$Y(k) = CX(k) + DU(k) \quad (ง.2)$$

$X(k)$ คือ ตัวแปรสถานะ n เวกเตอร์

$U(k)$ คือ สัญญาณควบคุม r เวกเตอร์

$Y(k)$ คือ สัญญาณที่ต้องการควบคุม m เวกเตอร์

A เมตริกซ์ขนาด $n \times n$

B เมตริกซ์ขนาด $n \times r$

C เมตริกซ์ขนาด $m \times n$

D เมตริกซ์ขนาด $m \times r$

เราสามารถหาค่าของ $X(k)$ และ $Y(k)$ โดย

$$X(k) = A^k X(0) + \sum_{j=0}^{k-1} A^{k-j-1} BU(j) \quad (ง.3)$$

$$Y(k) = CA^k X(0) + \sum_{j=0}^{k-1} CA^{k-j-1} BU(j) + DU(k) \quad (ง.4)$$

เนื่องเราทราบค่าของ A, B, C, D และ $U(k)$ ดังนั้นจึงทราบค่าเทอมที่ 2 และ 3 ด้านขวาของสมการ (ง.4) และสามารถนำมาลบออกจาก $Y(k)$ ได้ ทำให้การพิจารณาเงื่อนไขที่จำเป็นและเพียงพอสำหรับการสามารถสังเกตได้ไม่จำเป็นต้องนำ $U(k)$ มาพิจารณา ระบบที่นำมาพิจารณาจึงเหลือเพียง

$$X(k+1) = AX(k) \quad (ง.5)$$

$$Y(k) = CX(k) \quad (ง.6)$$

และสามารถหาค่าของ $X(k)$ และ $Y(k)$ โดย

$$X(k) = A^k X(0) \quad (ง.7)$$

$$Y(k) = CA^k X(0) \quad (ง.8)$$

ระบบจะถูกเรียกว่าสามารถสังเกตได้อย่างสมบูรณ์ (completely observable) ถ้าทุกๆ สถานะเริ่มต้น $X(0)$ สามารถสังเกตได้จากเอาต์พุตภายในช่วงจำกัด $Y(k), Y(k-1), Y(k-2), \dots, Y(k-n+1)$ เมื่อ n คือ อันดับของระบบ ประโยชน์ของการสังเกตได้คือการสร้างตัวแปรสถานะ (state variable) ที่ไม่สามารถวัดได้โดยตรงจากระบบจริงจากตัวแปรสถานะที่สามารถวัดได้

ถ้าระบบสามารถสังเกตได้อย่างสมบูรณ์หมายความว่า เราสามารถหาค่า $x_1(0), x_2(0), \dots, x_n(0)$ ได้จาก $Y(0), Y(1), \dots, Y(n-1)$ หรือ

$$\begin{aligned} Y(0) &= CX(0) \\ Y(1) &= CA X(0) \\ &\vdots \\ Y(n-1) &= CA^{n-1} X(0) \end{aligned}$$

หรือ

$$\begin{bmatrix} Y(0) \\ Y(1) \\ \vdots \\ Y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_n(0) \end{bmatrix}$$

เนื่องจาก $Y(k)$ เป็น m เวกเตอร์ ดังนั้นการอ้างอิงถึง $Y(0), Y(1), \dots, Y(n-1)$ จะทำให้ได้สมการจำนวน nm สมการที่อ้างอิงถึง $X(0)$ เพื่อที่จะหาค่าของ $X(0)$ ค่า $Y(k)$ แต่ละค่าจะต้องสามารถเขียนสมการ n สมการที่อ้างอิงถึง $X(0)$ นั่นคือเมตริกซ์

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

จะมีขนาด $nm \times n$ และต้องมีแรงค์เท่ากับ n เนื่องจากแรงค์ของทรานส์โพสเมตริกซ์จะเท่ากับแรงค์ของเมตริกซ์เดิม ดังนั้น

$$\text{rank} [C^T ; A^T C^T ; \dots ; (A^T)^{n-1} C^T] = n \quad (\text{ง.9})$$

เรียกเมตริกซ์นี้ว่าเมตริกซ์สามารถสังเกตได้ (observability matrix) เราจึงสรุปได้ว่า ถ้าแรงค์ของเมตริกซ์สามารถสังเกตได้ของระบบใดๆ มีค่าเท่ากับ n ระบบนั้นจะสามารถสังเกตได้

ภาคผนวก จ.

การออกแบบตัวควบคุม PI

การออกแบบตัวควบคุมชนิดไม่ต่อเนื่อง (discrete controller) มีหลายวิธี เช่น แปลงระบบที่ต้องการควบคุมเป็นชนิดไม่ต่อเนื่องก่อน แล้วจึงออกแบบตัวควบคุมไม่ต่อเนื่องโดยวิธีต่างๆ เช่น รุทโลกัส (root locus) หรือ โบทไดโอดแกรม (Bode diagram) หรือออกแบบตัวควบคุมชนิดต่อเนื่องสำหรับระบบที่ต้องการควบคุมชนิดต่อเนื่องก่อน แล้วจึงแปลงตัวควบคุมเป็นชนิดไม่ต่อเนื่อง ซึ่งการทำโดยวิธีนี้จะต้องใส่วงจรโฮลด์ (hold circuit) ไว้ระหว่างตัวควบคุมและระบบที่ต้องการควบคุม วงจรโฮลด์จะทำให้เกิดเวลาล่าช้า (time lag) ขึ้นในระบบที่ต้องการควบคุม ดังนั้นในการออกแบบตัวควบคุมชนิดไม่ต่อเนื่องต้องออกแบบเพื่อส่วนนี้ด้วย

ทรานส์เฟอฟังก์ชันของวงจรโฮลด์คือ

$$G_h(s) = (1 - e^{-sT})/s \quad (จ.1)$$

ซึ่ง

$$\begin{aligned} e^{-sT} &= \frac{1 - (sT)/2 + (sT)^2/8 - \dots}{1 + (sT)/2 + (sT)^2/8 + \dots} \\ &= \frac{1 - (sT)/2}{1 + (sT)/2} \end{aligned} \quad (จ.2)$$

$$\text{ดังนั้น } G_h(s) = T/(sT/2 + 1) \quad (จ.3)$$

เนื่องจากการแปลงจากระบบต่อเนื่องเป็นระบบไม่ต่อเนื่อง จะต้องปรับอัตราขยายดีซี (DC gain) อีกครั้งหนึ่ง ดังนั้นทรานส์เฟอฟังก์ชันของวงจรโฮลด์จะเขียนเป็น

$$G_h(s) = 1/(sT/2 + 1) \quad (จ.4)$$

จากบทที่ 2. ทรานส์เฟอฟังก์ชันของมอเตอร์กระแสตรง คือ

$$G_M(s) = \frac{K_T}{s^2 L J + s J R + K_T K_E} \quad (จ.5)$$

ซึ่งสามารถประมาณเป็นระบบอันดับ 1 เป็น [19]

$$G_M(s) = \frac{K_M}{1 + s T_M} \quad (จ.6)$$

$$\left. \begin{aligned} \text{โดยที่ } K_M &= 1/K_E = 4.546 \\ T_M &= J R / K_T K_E = 1.321 \times 10^{-2} \end{aligned} \right\} \quad (จ.7)$$

ดังนั้นทรานส์เฟอฟังก์ชันทั้งหมดของระบบควบคุมแบบลูปเปิด คือ

$$G_P(s) = \frac{K_A}{s N} \left[\frac{K_M}{(1+s T_M)} \right] \left[\frac{1}{(1+s T/2)} \right] \quad (จ.8)$$

เลือกใช้ช่วงเวลากการสุ่ม $T = 0.01$ วินาที แทนค่าต่างๆ ลงในสมการ (จ.8)

$$G_P(s) = \frac{1}{s(20)} \left[\frac{4.546}{1+s(1.321 \times 10^{-2})} \right] \left[\frac{1}{1+s(5 \times 10^{-3})} \right] \quad (จ.9)$$

จากสมการ (จ.8) และ (จ.9) ค่าเวลาดังที่ T_M และ $T/2$ จะมีค่าน้อยมากเมื่อเทียบกับ N ดังนั้นผลรวมของการหน่วงเวลา T_M และ $T/2$ สามารถรวมกันได้ [22] และจัดรูปสมการใหม่เพื่อความสะดวกในการออกแบบตัวควบคุมโดยวิธีซิมเมตริกคอลออปติ้ม (symmetrical optimum) [22]

$$G_P(s) = \frac{A_S}{s T_O (1+s T_E)} \quad (จ.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\left. \begin{aligned} \text{โดยที่ } A_S &= K_M K_A = 4.546 \\ T_o &= N = 20 \\ T_E &= T_M + T/2 = 2.321 \times 10^{-2} \end{aligned} \right\} \quad (จ. 11)$$

เลือกใช้ตัวควบคุม PI ซึ่งมีทรานส์เฟอฟังก์ชันเป็น

$$G_C(s) = A_R (1+sT_n) / sT_n \quad (จ. 12)$$

โดยวิธีซิมเมตริกคอนนอปติ้ม พารามิเตอร์ของตัวควบคุมคือ

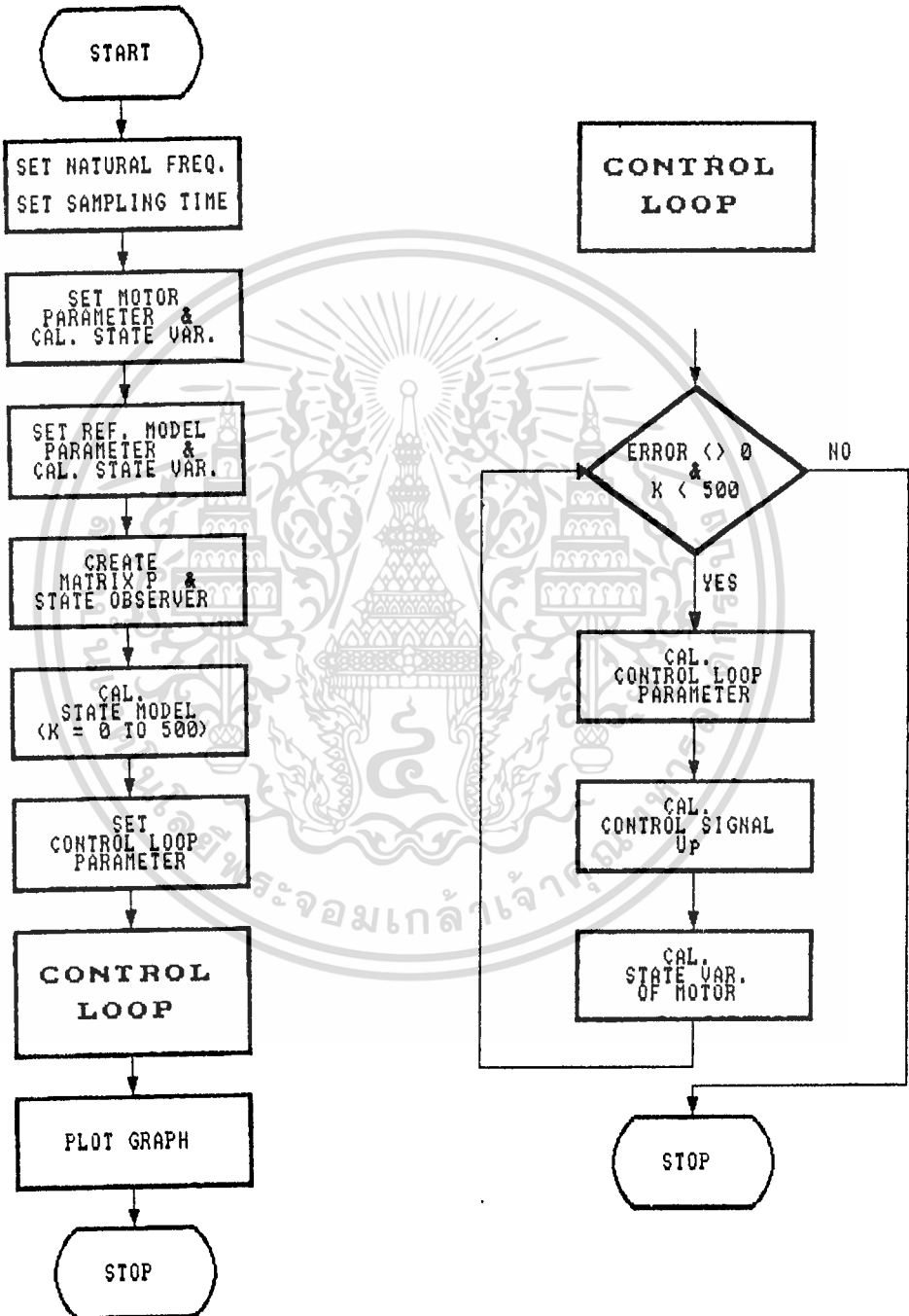
$$\left. \begin{aligned} A_R &= T_o / 2A_S T_E = 94.78 \\ T_n &= 4T_E = 9.284 \times 10^{-2} \end{aligned} \right\} \quad (จ. 13)$$

หลังจากนั้นจึงแปลงตัวควบคุมชนิดต่อเนื่องเป็นชนิดไม่ต่อเนื่อง โดยวิธี ไบลิเนียร์ทรานส์ฟอร์มเมชัน

$$G_C(z) = \frac{A_R}{2T_n} \left[\frac{(T+2T_n) + z^{-1}(T-2T_n)}{1 - z^{-1}} \right] \quad (จ. 14)$$

ภาคผนวก จ.

โปรแกรมจำลองผลตอบสนอง และความคุมตำแหน่งโดย MRAC



รูปที่ จ. 1 แผนภาพแสดงลำดับการทำงานของโปรแกรมจำลองผลตอบสนองด้วยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* โปรแกรมจำลองผลตอบสนองด้วยคอมพิวเตอร์ */
```

```
#include "stdio.h"
```

```
#include "conio.h"
```

```
#include "math.h"
```

```
#include "graphics.h"
```

```
#include "process.h"
```

```
#include "dos.h"
```

```
#define N 4
```

```
#define n 3
```

```
float max, x[3][501], xp[3][501], up[501], R[N], L[N], h[N], yp[501];
```

```
float ica[N][N], p[7], maxd, A[4], maxu;
```

```
float Bp[N][N], cab[N][N], dp[501];
```

```
main()
```

```
{
```

```
float cabu[N][N][N], cabd[N][N][N], ih[N][N], ig[N][N], B[N];
```

```
float c[N], F[N], g[N], maxx, maxp, u, w, m1, m2, m3, m4;
```

```
float a2, a3, a4, a5, a6, ka, kt, r, ind, inter, ke, gear;
```

```
float T, T1, T2, T3, T4, e1, e2, e3, e4, c1u, c2u, c3u;
```

```
float x4, x5, h4, h5, h6, c4, c5, c6;
```

```
float t1, t2, t3, tu, td, aa0, aa1, aa2;
```

```
int i, j, k, m;
```

```
double temp;
```

```
struct time now;
```

```

/* กำหนดความถี่ธรรมชาติของโมเดลอ้างอิง */
printf("\nNatural frequency Wn : ");
scanf("%f",&w);

/* กำหนดช่วงเวลาการสุ่ม = 0.01 วินาที */
T = 0.01;
printf("\nSampling time : %10.7f\n",T);
T2 = T*T;
T3 = T2*T;
T4 = T3*T;

/* กำหนดพารามิเตอร์ของมอเตอร์กระแสตรง */
ka = 1.0; /* อัตราขยายของวงจรถ้าลิ่ง */
kt = 2.24e-2; /* ค่าคงที่ทอร์ค */
ke = 0.22; /* ค่าคงที่โวลต์เตจ */
r = 3.1; /* ค่าความต้านทานขดลวด */
ind = 4.7e-3; /* ค่าความเหนี่ยวนำขดลวด */
gear = 20; /* อัตราทดของเฟืองทดรอบ */
inter = 2.1e-5; /* ค่าความเฉื่อย */

/* คำนวณค่าคงที่สมการสถานะชนิดต่อเนื่องของมอเตอร์ */
a2 = kt*ke/(ind*inter);
a3 = r/ind;
a4 = ka*kt/(gear*ind*inter);
a5 = 1/(gear*inter);
a6 = r/(gear*ind*inter);

```

/* คำนวณค่าคงที่สมการสถานะชนิดไม่ต่อเนื่องของมอเตอร์ */

$$R[0] = 8 + 4*a3*T + 2*a2*T2;$$

$$R[1] = (-24 - 4*a3*T + 2*a2*T2)/R[0];$$

$$R[2] = (24 - 4*a3*T - 2*a2*T2)/R[0];$$

$$R[3] = (-8 + 4*a3*T - 2*a2*T2)/R[0];$$

$$L[0] = (a4*T3)/R[0];$$

$$L[1] = (3*a4*T3)/R[0];$$

$$L[2] = L[1];$$

$$L[3] = L[0];$$

$$F[0] = (2*a5*T2 + a6*T3)/R[0];$$

$$F[1] = (2*a5*T2 + 3*a6*T3)/R[0];$$

$$F[2] = (-2*a5*T2 + 3*a6*T3)/R[0];$$

$$F[3] = (-2*a5*T2 + a6*T3)/R[0];$$

$$h[0] = L[0];$$

$$h[1] = L[1] - R[1]*h[0];$$

$$h[2] = L[2] - R[1]*h[1] - R[2]*h[0];$$

$$h[3] = L[3] - R[1]*h[2] - R[2]*h[1] - R[3]*h[0];$$

$$g[0] = F[0];$$

$$g[1] = F[1] - R[1]*g[0];$$

$$g[2] = F[2] - R[1]*g[1] - R[2]*g[0];$$

$$g[3] = F[3] - R[1]*g[2] - R[2]*g[1] - R[3]*g[0];$$

```

for(i = 0; i <= 3; i++)
{
    Bp[i][1] = h[i];
    Bp[i][2] = g[i];
}

```

/* คำนวณค่าคงที่สมการสถานะของโมเดลอ้างอิงชนิดต่อเนื่อง */

m1 = w*w*w;

m2 = 2.15*w*w;

m3 = 1.75*w;

m4 = m1;

/* คำนวณค่าคงที่สมการสถานะของโมเดลอ้างอิงชนิดไม่ต่อเนื่อง */

A[0] = 8+4*m3*T+2*m2*T2+m1*T3;

A[1] = (-24-4*m3*T+2*m2*T2+3*m1*T3)/A[0];

A[2] = (24-4*m3*T-2*m2*T2+3*m1*T3)/A[0];

A[3] = (-8+4*m3*T-2*m2*T2+m1*T3)/A[0];

B[0] = (m4*T3)/A[0];

B[1] = (3*m4*T3)/A[0];

B[2] = B[1];

B[3] = B[0];

c[0] = B[0];

c[1] = B[1] - A[1]*c[0];

c[2] = B[2] - A[1]*c[1] - A[2]*c[0];

c[3] = B[3] - A[1]*c[2] - A[2]*c[1] - A[3]*c[0];

```
/* สร้างเมตริกซ์ P โดยใช้โปรแกรมย่อย guass() */
```

```
guass();
```

```
/* สร้างตัวสังเกตสถานะ (state observer) โดยใช้โปรแกรมย่อย observer() */
```

```
observer();
```

```
o/* สร้างเมตริกซ์สำหรับใช้กับตัวสังเกตสถานะ */
```

```
for(i = 1; i <= 3; i++)
```

```
{
```

```
for(j = 1; j <= 3; j++)
```

```
{
```

```
for(m = 1; m <= 3; m++)
```

```
{
```

```
cabu[i][j][m] = 0.0; cabd[i][j][m] = 0.0;
```

```
}
```

```
}
```

```
}
```

```
for(i = 1; i <= 3; i++)
```

```
{
```

```
for(m = 1; m <= 3; m++)
```

```
{
```

```
for(j = m; j <= 3; j++)
```

```
{
```

```
cabu[i][j][m] = ica[i][j]*cab[m][1];
```

```
cabd[i][j][m] = ica[i][j]*cab[m][2];
```

```
}
```

```
}
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i = 1; i <= 3; i++)
{
    for(j = 1; j <= 3; j++)
    {
        ih[i][j] = ica[i][j]*h[0];
        ig[i][j] = ica[i][j]*g[0];
    }
}

/* กำหนดพารามิเตอร์เริ่มต้นสำหรับคำนวณผลตอบสนองของโมเดลอ้างอิง */
x[0][0] = x[1][0] = x[2][0] = 0.0;
u = 1.0;
c1u = c[1]*u;
c2u = c[2]*u;
c3u = c[3]*u;
x[2][1] = c3u;
x[1][1] = c2u;
x[0][1] = c1u;

/* คำนวณผลตอบสนองของโมเดลอ้างอิง */
printf("Calculate response\n");
maxx = x[0][0];
for(k = 1; k <= 500; k++)
{
    x[2][k+1] = c3u - A[1]*x[2][k] - A[2]*x[1][k] - A[3]*x[0][k];
    x[1][k+1] = x[2][k] + c2u;
    x[0][k+1] = x[1][k] + c1u;
}

```

```

/* หาค่าผลตอบสนองมากที่สุดเพื่อใช้กำหนดสเกลกราฟ */
if(maxx < x[0][k])
    maxx = x[0][k];
}

/* กำหนดค่าเริ่มต้นผลตอบสนองของระบบควบคุมตำแหน่ง */
for(i = 0; i <= 500; i++)
    dp[i] = 0.0;

up[0] = up[1] = 0.0;
xp[0][0] = xp[1][0] = xp[2][0] = 0.0;
xp[0][1] = xp[1][1] = xp[2][1] = 0.0;

/* กำหนดและคำนวณค่าเริ่มต้นและพารามิเตอร์ของอุปกรณ์ควบคุม */
maxp = xp[0][0];
maxu = up[0];
maxd = 0.0;

h4 = h[1]*p[1] + h[2]*p[2] + h[3]*p[3];
h5 = h[1]*p[2] + h[2]*p[4] + h[3]*p[5];
h6 = h[1]*p[3] + h[2]*p[5] + h[3]*p[6];
aa2 = h[1]*h4 + h[2]*h5 + h[3]*h6;

/* จับเวลาเริ่มต้น */
gettime(&now);
printf("start time%02d:%02d:%02d:%02d\n",now.ti]hour,now.ti]min,
now.ti]sec,now.ti]hund);

```

```

/* ลูปควบคุม */
for(k = 2; k <= 500; k++)
{
    e1 = x[0][k] - xp[0][k];
    e2 = x[1][k] - xp[1][k];
    e3 = x[2][k] - xp[2][k];
    e4 = e1*R[3] + e2*R[2] + e3*R[1];
    c4 = c1u - g[1]*dp[k];
    c5 = c2u - g[2]*dp[k];
    c6 = c3u - g[3]*dp[k];
    x4 = (R[3]-A[3])*x[0][k] + (R[2]-A[2])*x[1][k]
        + (R[1]-A[1])*x[2][k];
    x5 = x4 + c6;
    t1 = p[1]*c4 + p[2]*c5 + p[3]*x5;
    t2 = p[2]*c4 + p[4]*c5 + p[5]*x5;
    t3 = p[3]*c4 + p[5]*c5 + p[6]*x5;

    aa0 = t1*(c4+2*e2) + t2*(c5+2*e3) + t3*(x5-2*e4);
    aa1 = 2*(h4*e2 + h5*e3 - h6*e4 + h[1]*t1 + h[2]*t2 + h[3]*t3);
    temp = aa1*aa1 - 4*aa0*aa2;
    temp = sqrt(temp);
    up[k] = (temp + aa1)/(2*aa2);

    /* หาขนาด  $u_p$  สูงสุดเพื่อกำหนดสเกลกราฟ */
    t1 = (up[k] < 0)? -up[k]: up[k];
    if(maxu < t1)
        maxu = t1;
}

```

```

/* โปรแกรมจำลองผลตอบสนอง สามารถหา  $y_p(k)$  ได้จาก  $X_p(k)$  และ  $U_p(k)$  */
yp[k] = xp[0][k] + h[0]*up[k] + g[0]*dp[k];

```

```

/* สร้างตัวแปรสถานะ (  $X_p(k+1)$  ) ของระบบควบคุมตำแหน่ง */

```

```

for(i = 1; i <= 3; i++)
{
    t2 = t3 = tu = td = 0.0;
    for(j = 1; j <= 3; j++)
    {
        for(m = 1; m <= j; m++)
        {
            tu += cabu[i][j][m]*up[k+m-j];
            td += cabd[i][j][m]*dp[k+m-j];
        }
        t2 += ih[i][j]*up[k+1-j];
        t3 += ig[i][j]*dp[k+1-j];
    }
    xp[i-1][k+1] = tu + td - t2 - t3;
}
for(i = 1; i <= 3; i++)
{
    t1 = 0.0;
    for(j = 1; j <= 3; j++)
    {
        t1 += ica[i][j]*yp[k+1-j];
    }
    xp[i-1][k+1] += t1;
}

```

```

/* หาผลตอบสนองของระบบควบคุมตำแหน่งสูงสุด เพื่อใช้กำหนดสเกลกราฟ */
t1 = (xp[0][k] < 0)? -xp[0][k]: xp[0][k];
if(maxp < t1)
    maxp = t1;
}
/* จบโปรแกรม */

/* จับเวลาสิ้นสุด */
gettime(&now);
printf("final time%02d:%02d:%02d\n",now.tilhour,now.tilmin,
now.tilsec,now.tilhund);

/* หาขนาดสูงสุดระหว่างผลตอบของ โมเดลอ้างอิงและผลตอบสนองของระบบควบคุมตำแหน่ง */
max = maxx;
if(max < maxp)
    max = maxp;

/* เรียกโปรแกรมย่อยวาดกราฟ */
plot();
}

/* จบโปรแกรมหลัก */

```

```

/* โปรแกรมย่อย วาดกราฟ */
plot()
{
float scale;
int  gldriver, glmode, i, j;
int  x1, x2, y1, y2, dx;

/* เปลี่ยนจอภาพเป็นกราฟฟิกโหมด */
gldriver = DETECT;
clrscr();
initgraph(&gldriver,&glmode," ");

/* วาดตารางกราฟ */
line(60,80,560,80);
line(60,320,560,320);
line(60,80,60,320);
line(560,80,560,320);
line(60,200,560,200);
for(i = 60; i <= 560; i += 50)
{
for(j = 80; j <= 310; j += 10)
line(i,j,i,j);
}
for (i = 80; i <= 320; i += 20)
{
for(j = 60; j <= 550; j += 10)
line(j,i,j,i);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* กำหนดสเกลสำหรับกราฟของผลตอบสนองของโมเดลอ้างอิงและระบบควบคุมตำแหน่ง */
scale = 1.2;
if(max > scale)
    scale = 1.2*max;

/* วาดกราฟผลตอบสนองของระบบควบคุมตำแหน่ง */
setcolor(10);
dx = 1;
x1 = 60;
y1 = 200 - (int)(120*xp[0][0]/scale);
for (i = 1; i <= 500; i++)
{
    x2 = x1 + dx;
    y2 = 200 - (int)(120*xp[0][i]/scale);
    line(x1,y1,x2,y2);
    x1 = x2;
    y1 = y2;
}
line(60,10,75,10);
outtextxy(90,10,"Process          Xp");

/* วาดกราฟผลตอบสนองของโมเดลอ้างอิง */
setcolor(12);
dx = 1;
x1 = 60;
y1 = 200 - (int)(120*x[0][0]/scale);

```

```

for (i = 1; i <= 500; i++)
{
    x2 = x1 + dx;
    y2 = 200 - (int)(120*x[0][i]/scale);
    line(x1,y1,x2,y2);
    x1 = x2;
    y1 = y2;
}
line(60,30,75,30);
outtextxy(90,30,"Reference Model   Xm");

/* กำหนดสเกล และวาดกราฟสัญญาณควบคุม Up */
maxu *= 1.2;
setcolor(11);
dx = 1;
x1 = 60;
y1 = 200 - (int)(120*up[0]/maxu);
for (i = 1; i <= 500; i++)
{
    x2 = x1 + dx;
    y2 = 200 - (int)(120*up[i]/maxu);
    line(x1,y1,x2,y2);
    x1 = x2;
    y1 = y2;
}
line(60,50,75,50);
outtextxy(90,50,"Command           Up");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* กำหนดสเกล และวาดกราฟสัญญาณรบกวน  $d_p$  */
maxd *= 1.2;
setcolor(13);
dx = 1;
x1 = 60;
y1 = 200 - (int)(120*dp[0]/maxd);
for (i = 1; i <= 500; i++)
{
    x2 = x1 + dx;
    y2 = 200 - (int)(120*dp[i]/maxd);
    line(x1,y1,x2,y2);
    x1 = x2;
    y1 = y2;
}
line(60,70,75,70);
outtextxy(90,70,"Disturbance      Dp");
getch();

/* เปลี่ยนจอภาพเป็นเท็กซ์โหมด และจบโปรแกรมย่อยวาดกราฟ */
cleardevice();
closegraph();
textmode(3);
}

```

```

/* โปรแกรมย่อย คูณเมตริกซ์ */
mulmat(float c[][N], float d[][N], float e[][N], int n1, int n2, int n3)
{
    int    i, j, k;
    float  temp;

    for(i = 1; i <= n1; i++)
    {
        for(j = 1; j <= n3; j++)
        {
            temp = 0.0;
            for(k = 1; k <= n2; k++)
                temp += c[i][k]*d[k][j];
            e[i][j] = temp;
        }
    }
}

```

```

/* โปรแกรมย่อย อินเวอร์เมตริกซ์ */
invlmat(float a[][N])
{
    int i, j, k, k1, id[N];
    float p, t;

    for(i = 1; i <= n; i++)
        id[i] = i;
    for(k = 1; k <= n; k++)
    {
        pivot(a,k,id);
        p = a[k][k];
        a[k][k] = 1;
        for(j = 1; j <= n; j++)
            a[k][j] /= p;
        for(i = 1; i <= n; i++)
        {
            if(i != k)
            {
                for(j = 1; j <= n; j++)
                {
                    if(j != k)
                        a[i][j] -= a[i][k]*a[k][j];
                }
                a[i][k] /= -p;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(j = 1; j <= n-1; j++)
{
    if(j  $\neq$  id[j])
    {
        for(k = j+1; k <= n; k++)
        {
            if(j == id[k])
            k1 = k;
        }
        for(i = 1; i <= n; i++)
        {
            t = a[i][j];
            a[i][j] = a[i][k1];
            a[i][k1] = t;
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* โปรแกรมย่อย pivot สำหรับใช้ในโปรแกรมย่อย invmat */
pivot(float b[][N], int kk, int im[])
{
    float abso, p, t;
    int j, k, m;

    k = kk;
    p = (b[k][k] < 0)? -b[k][k]: b[k][k];
    m = k;

    for(j = k; j <= n; j++)
    {
        abso = (b[j][k] < 0)? -b[j][k]: b[j][k];
        if(p < abso)
        {
            p = abso;
            m = j;
        }
    }
    if(p == 0)
        printf("cannot solve\n");

    if(m != k)
    {
        for(j = 1; j <= n; j++)
        {
            t = b[k][j];
            b[k][j] = b[m][j];

```

```

    b[m][j] = t;
}
j = im[k];
im[k] = im[m];
im[m] = j;
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* โปรแกรมย่อย สำหรับสร้างตัวแปรสถานะของระบบควบคุมตำแหน่ง */
observer()
{
    int    i, j;
    float  g1[N][N], g2[N][N], g3[N][N];

    /*  $A_p^{-1} \Rightarrow G1$  */
    g1[1][1] = -R[2]/R[3];
    g1[1][2] = -R[1]/R[3];
    g1[1][3] = -1/R[3];
    g1[2][1] = 1;
    g1[2][2] = 0;
    g1[2][3] = 0;
    g1[3][1] = 0;
    g1[3][2] = 1;
    g1[3][3] = 0;

    /* เตรียมพื้นที่สำหรับเก็บ  $A_p^{-2}$  และ  $A_p^{-3}$  */
    for(i = 1; i <= n; i++)
    {
        for(j = 1; j <= n; j++)
        {
            g2[i][j] = 0.0;
            g3[i][j] = 0.0;
        }
    }
}

```

```

/* คำนวณ  $A_p^{-2}$  และ  $A_p^{-3}$  */
mulmat(g1,g1,g2,3,3,3);
mulmat(g2,g1,g3,3,3,3);

/* สร้างเมตริกซ์  $C_p A_p$  */
printf("creat matrix ica before invert\n");
for(j = 1; j <= 3; j++)
    ica[1][j] = g1[1][j];
for(j = 1; j <= 3; j++)
    ica[2][j] = g2[1][j];
for(j = 1; j <= 3; j++)
    ica[3][j] = g3[1][j];

/* อินเวอร์ส  $C_p A_p$  */
invmat(ica);
mulmat(g1,Bp,g1,3,3,2); /*  $A_p^{-1} B_p$  */
mulmat(g2,Bp,g2,3,3,2); /*  $A_p^{-2} B_p$  */
mulmat(g3,Bp,g3,3,3,2); /*  $A_p^{-3} B_p$  */

/* สร้างเมตริกซ์  $C_p A_p^{-1} B_p \dots C_p A_p^{-3} B_p$  */
printf("\nmatrix CpiAp^1-3Bp\n");
for(j = 1; j <= 2; j++)
{
    cab[1][j] = g1[1][j];
    cab[2][j] = g2[1][j];
    cab[3][j] = g3[1][j];
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/* โปรแกรมย่อยสร้างเมตริกซ์ P */

guass()

{

double ma[7][7], q[7], mul, temp , d[4];

float r[4];

int i, j, k, id[7];

/* นำข้อมูลเมตริกซ์ A_p มาหยังโปรแกรมย่อย */

for(i = 1; i <= 3; i++)

 r[i] = R[i];

/* สร้างเมตริกซ์ $A_p^T P A_p - P$ */

for(i = 1; i <= 6; i++)

{

 for(j = 1; j <= 6; j++)

 ma[i][j] = 0.0;

}

ma[1][1] = 1.0;

ma[1][3] = -2.0*r[2];

ma[1][4] = -1.0;

ma[1][6] = r[2]*r[2];

ma[2][2] = 1.0;

ma[2][3] = -r[1];

ma[2][5] = -r[2] - 1.0;

ma[2][6] = r[1]*r[2];

ma[3][2] = -1.0;

ma[3][3] = -r[3];

ma[3][6] = r[2]*r[3];

```

ma[4][4] = 1.0;
ma[4][5] = -2.0*r[1];
ma[4][6] = r[1]*r[1] - 1.0;
ma[5][3] = -1.0;
ma[5][5] = -r[3];
ma[5][6] = r[1]*r[3];
ma[6][1] = -1.0;
ma[6][6] = r[3]*r[3];

```

```

/* กำหนดเมตริกซ์ Q */

```

```

q[1] = q[4] = q[6] = -1.0;
q[2] = q[3] = q[5] = 0.0;

```

```

/* หาเมตริกซ์ P โดยวิธี guass elimination */

```

```

for(i = 1; i <= 6; i++)
    id[i] = i;
for(k = 1; k <= 6-1; k++)
{
    for(i = k+1; i <= 6; i++)
    {
        pivot2(ma,k,id);
        if(ma[i][k] != 0.0)
        {
            mul = -ma[k][k]/ma[i][k];
            for(j = k; j <= 6; j++)
            {
                ma[i][j] *= mul;    ma[i][j] += ma[k][j];
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    q[i] *= mul;
    q[i] += q[k];
}
}
}

p[6] = q[6]/ma[6][6];
for(i = 6-1; i >= 1; i--)
{
    temp = 0.0;
    for(j = 6; j >= i+1; j--)
        temp += ma[i][j]*p[j];
    p[i] = (q[i] - temp)/ma[i][i];
}

/* ทดสอบการมีความหมายเป็นบวกของเมตริกซ์ P */
printf("Positive definite testing\n");
d[1] = p[1];
d[2] = p[1]*p[4] - p[2]*p[2];
d[3] = p[1]*(p[4]*p[6]-p[5]*p[5]) - p[2]*(p[2]*p[6]-p[3]*p[5]);
d[3] += p[3]*(p[2]*p[5]-p[4]*p[3]);

if(d[1] > 0.0 && d[2] > 0.0 && d[3] > 0.0)
    printf("Can use\n");
else printf("Cannot use\n");

printf("Data Normalization\n");

mul = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(i = 1; i <= 6; i++)  
{  
    temp = (p[i] < 0)? -p[i]: p[i];  
    if(mul < temp)  
        mul = temp;  
}  
for(i = 1; i <= 6; i++)  
    p[i] /= mul;  
}
```



```

/* โปรแกรมย่อย pivot2 สำหรับใช้ในโปรแกรมย่อย guass */
pivot2(double b[][7], int kk, int im[])
{
    double abso, pp, t;
    int j, k, m, ln;

    ln = 6;
    k = kk;
    pp = (b[k][k] < 0)? -b[k][k]: b[k][k];
    m = k;
    for(j = k; j <= ln; j++)
    {
        abso = (b[j][k] < 0)? -b[j][k]: b[j][k];
        if(pp < abso)
        {
            pp = abso;
            m = j;
        }
    }
    if(pp == 0)
        printf("cannot solve\n");

    if(m != k)
    {
        for(j = 1; j <= ln; j++)
        {
            t = b[k][j];
            b[k][j] = b[m][j];

```

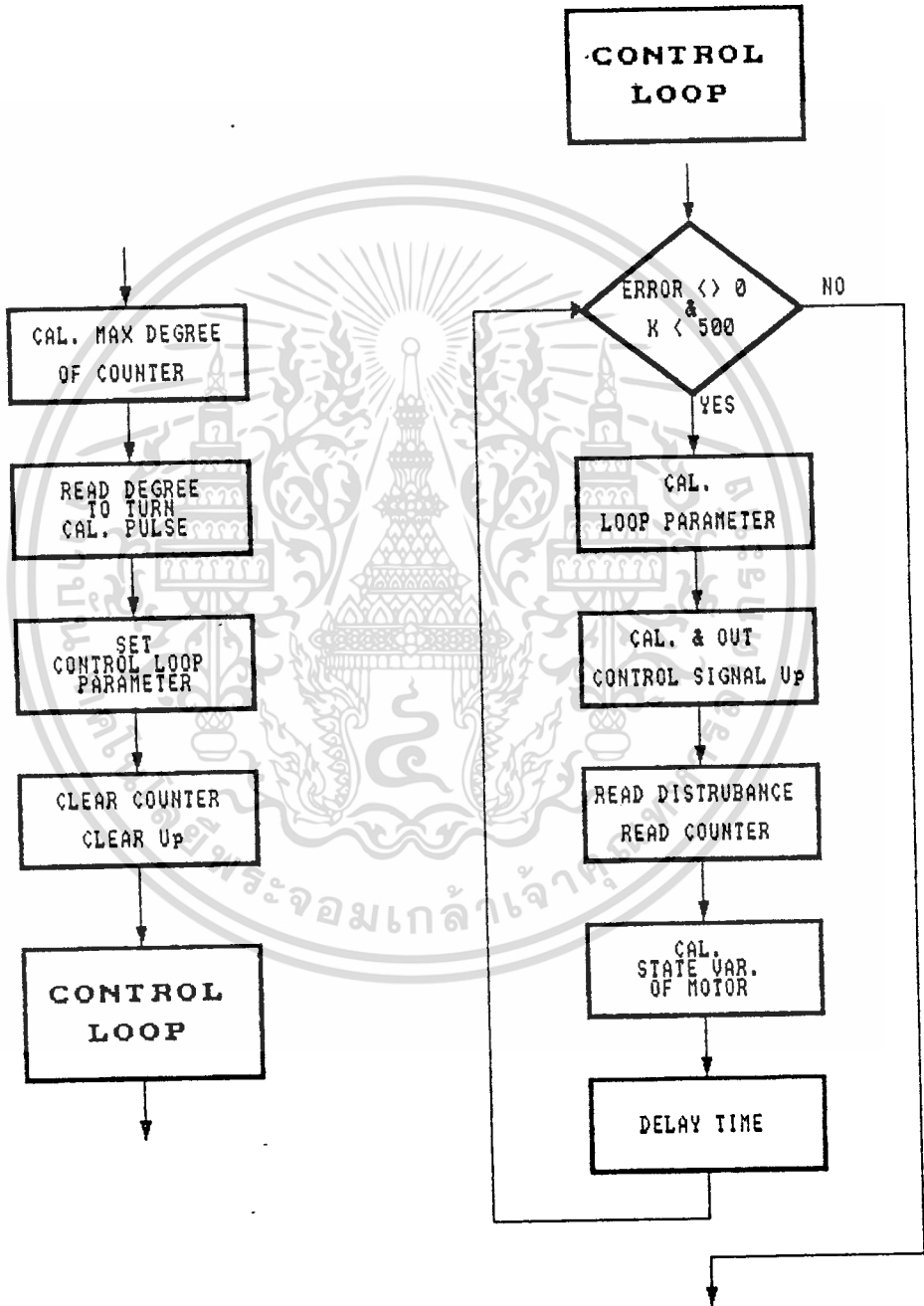
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        b[m][j] = t;
    }
    j = im[k];
    im[k] = im[m];
    im[m] = j;
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

/* โปรแกรมใช้งานจริง เฉพาะส่วนควบคุม และส่วนที่แตกต่างจาก โปรแกรมจำลองผล
 ตอบสนอง */



รูปที่ ๑.2 แผนภาพแสดงลำดับการทำงานในระบบจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* กำหนดพอร์ตเพื่อใช้ติดต่อกับอุปกรณ์ภายนอก */
#define pIout    0x2f0    /* พอร์ตสำหรับส่งสัญญาณควบคุม  $U_p$  */
#define pIin1    0x2f1    /* พอร์ตสำหรับรับ 8 บิตล่างของวงจรมับความถี่ */
#define pIin2    0x2f2    /* พอร์ตสำหรับรับ 8 บิตบนของวงจรมับความถี่ */
#define pIclr    0x2f3    /* พอร์ตสำหรับเคลียร์วงจรมับความถี่ */
#define pIadc    0x2f4    /* พอร์ตสำหรับรับค่ากระแสโมเตอร์จาก ADC */

/* ตัวแปรที่ต้องกำหนดเพิ่ม */
struct time    time1, time2;
int    noload, nomax, nomin, tempin, io, dat[4];
char    out;
float    pos, pulse, deg, in, in1, in2, dtime, dif;

/* คำนวณจำนวนองศาที่วงจรมับความถี่สามารถนับได้สูงสุด */
t2    = (32768.0*360.0)/(1000.0*gear);    /* องศาสูงสุด */
t1    = 1e5;

/* รับจำนวนองศาที่ต้องให้หมุน ซึ่งต้องไม่มากกว่าจำนวนองศาสูงสุด */
while(t1 > t2)
{
    printf("How many degree to turn (< %10.5f): ", t2);
    scanf("%f", &deg);
    t1 = (deg < 0)? -deg: deg;
}

```

```

/* คำนวณค่าพัลส์ที่วางจนวนับความถี่ต้องนับ และตำแหน่งที่มอเตอร์ต้องหยุดหมุน */
pulse = 1000.0*gear*deg/360.0;
pos = 32768.0 + pulse;
pulse = (pulse < 0.0)? -pulse: pulse;

/* กำหนดทิศทางการหมุนของมอเตอร์ */
if (deg >= 0.0)
    u = 1.0;          /* มอเตอร์หมุนเดินหน้า */
else
    u = -1.0;        /* มอเตอร์หมุนถอยหลัง */

/* กำหนดพารามิเตอร์สำหรับลูปควบคุม เพิ่มเติมจากโปรแกรมจำลองผลตอบสนอง */
maxu = 75.0;          /* สัญญาณควบคุมมากที่สุด = +/- 75 โวลต์ */
noload = 30;          /* DC CT 3.4 A = 2.5 V */
nomax = 127 + noload; /* full load = 3.4 A */
nomin = 127 - noload; /* 5 V = 256 */
perc = 100.0/(256 - nomax); /* no load = 0.8 A = 30 */

in = 0.0;
dtime = T*100.0;
out = (char)(127);
outportb(plout,out); /* ให้สัญญาณควบคุมอยู่ที่ 0 โวลต์ */
outportb(plclr,out); /* เคลียร์วางจนวนับความถี่ */
dat[0] = dat[1] = dat[2] = 0;
k = 3;

```

```

/* ลูปควบคุม */
while(pos Ω= in && k <= 499)
{
    /* คำานวณพหาวามิตอร์ต่างๆ ที่ใช้ในลูปควบคุม */
    gettimeofday(&time1);    /* จับเวลาเริ่มต้นของแต่ละลูป */

    e1 = x[0][k] - xp[0][k];
    e2 = x[1][k] - xp[1][k];
    e3 = x[2][k] - xp[2][k];
    e4 = e1*R[3] + e2*R[2] + e3*R[1];

    c4 = c1u - g[1]*dp[k];
    c5 = c2u - g[2]*dp[k];
    c6 = c3u - g[3]*dp[k];
    x4 = (R[3]-A[3])*x[0][k] + (R[2]-A[2])*x[1][k]
        + (R[1]-A[1])*x[2][k];
    x5 = x4 + c6;

    t1 = p[1]*c4 + p[2]*c5 + p[3]*x5;
    t2 = p[2]*c4 + p[4]*c5 + p[5]*x5;
    t3 = p[3]*c4 + p[5]*c5 + p[6]*x5;

    aa0 = t1*(c4+2*e2) + t2*(c5+2*e3) + t3*(x5-2*e4);
    aa1 = 2*(h4*e2 + h5*e3 - h6*e4 + h[1]*t1 + h[2]*t2 + h[3]*t3);
    temp = aa1*aa1 - 4*aa0*aa2;
    temp = sqrt(temp);
    up[k] = (temp + aa1)/(2*aa2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* จำกัดสัญญาณควบคุม up สูงสุด 75 โวลต์ และต่ำสุด -75 โวลต์ */
if(up[k] > maxu)
    up[k] = maxu;
if(up[k] < -maxu)
    up[k] = -maxu;

/* ถ้าสัญญาณควบคุมกลับขั้ว ให้สัญญาณควบคุมอยู่ที่ 0 โวลต์ ก่อนจึงกลับขั้วสัญญาณควบคุม */
if(up[k-1]*up[k] < 0)
{
    out = (char)(127);
    outportb(p1out,out);
}

/* แปลงสัญญาณควบคุม up เป็นเลขไบนารี และส่งออกทางพอร์ต p1out */
up[k] += maxu;
io = (int)(1.7*up[k]);          /* 256 / 2maxu = 1.70667 */
out = (char)(io);
outportb(p1out,out);
up[k] -= maxu;

/* รับ distrubance จากพอร์ต p1adc และหาค่าเฉลี่ยก่อนนำไปใช้ในการคำนวณ */
tempin = inportb(p1adc);
tempin = (dat[2] + dat[1] + dat[0] + tempin)/4;
dat[2] = dat[1];
dat[1] = dat[0];
dat[0] = tempin;

```

```

if(tempin > nomax || tempin < nomin)
{
    if(tempin > nomax)
        tempin -= nomax;
    else
        tempin -= nomin;
    dp[k] = (float)(tempin);
    dp[k] *= perc;
}
else dp[k] = 0.0;

/* คำนวณพารามิเตอร์สำหรับตัวสังเกตสถานะ */
for(i = 1; i <= 3; i++)
{
    t2 = t3 = tu = td = 0.0;
    for(j = 1; j <= 3; j++)
    {
        for(m = 1; m <= j; m++)
        {
            tu += cabu[i][j][m]*up[k+m-j];
            td += cabd[i][j][m]*dp[k+m-j];
        }
        t2 += ih[i][j]*up[k+1-j];
        t3 += id[i][j]*dp[k+1-j];
    }
    xp[i-1][k+1] = tu + td - t2 - t3;
}

```

```

/* รับสัญญาณจากวงจรนับความถี่ทางพอร์ต p1in2 และ p1in1 และคำนวณองค์ค่าที่หมุน */
tempin = inportb(p1in2);
in2 = (float)(tempin);
in2 *= 256.0;
tempin = inportb(p1in1);
in1 = (float)(tempin);
in = in1 + in2;
yp[k] = (in - 32768.0)/pulse;

/* คำนวณตัวแปรสถานะของระบบควบคุมตำแหน่ง */
for(i = 1; i <= 3; i++)
{
    t1 = 0.0;
    for(j = 1; j <= 3; j++)
        t1 += ica[i][j]*yp[k+1-j];
    xp[i-1][k+1] += t1;
}

/* หาขนาดผลตอบสนองของระบบควบคุมตำแหน่งสูงสุด เพื่อกำหนดสเกลกราฟ */
t1 = (xp[0][k] < 0)? -xp[0][k]: xp[0][k];
if(maxp < t1)
    maxp = t1;

k++;

/* หน่วยงานทำงานให้ช่วงเวลาเท่ากับช่วงเวลาการสุ่ม */
dif = 0.0;

```

```

while(dif < dtime)
{
    gettimeofday(&time2);
    if(time2.tv_usec < time1.tv_usec)
        time2.tv_usec += 100;
    dif = (float)(time2.tv_usec - time1.tv_usec);
}
}
/* จบโปรแกรมการทำงาน */

/* ทำให้สัญญาณควบคุม up เป็น 0 โวลต์ */
out = (char)(127);
outportb(plout, out);

```

ภาคผนวก ช.
พิกัดของมอเตอร์

พิกัดเอาต์พุต	180	วัตต์
โวลต์เตจพิกัดขดลวดอาร์มาเจอร์	75	โวลต์
กระแสพิกัดขดลวดอาร์มาเจอร์	3.4	แอมแปร์
ความเร็วพิกัด	3000	รอบ / นาที
ค่าคงที่ทอร์ค	2.24	กิโลกรัม เซ็นติเมตร / แอมแปร์
ค่าคงที่โวลต์เตจ	23	โวลต์ / 1000 รอบ / นาที
ค่าความเฉื่อย	2.1	กรัม เซ็นติเมตร วินาที ²
ความต้านทานขดลวดอาร์มาเจอร์	3.1	โอห์ม
ความเหนี่ยวนำขดลวดอาร์มาเจอร์	4.7	มิลลิเฮนรี่

ภาคผนวก ข.

ผลงานที่ได้รับการเผยแพร่ในระหว่างศึกษาปริญญาโท

ชัยศรี เอี่ยมอำไพ และ โยธิน เปรมปราณีรัตน์, "การใช้ MRAC เพื่อปรับปรุงผลตอบสนองของระบบควบคุมตำแหน่งของดีซีมอเตอร์เมื่อมีสัญญาณรบกวนจากภายนอก", การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 13 ณ มหาวิทยาลัยเชียงใหม่, หน้า 397 - 408, วันที่ 8 - 9 พฤศจิกายน 2533

