

การออกแบบและการสร้างตัวควบคุม PIDA สำหรับกระบวนการอันดับสามเพื่อ
ใช้ควบคุมความเร็วมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น

THE DESIGN AND IMPLEMENTATION OF A THIRD ORDER PLANT
CONTROLLER BY USING PIDA FOR SPEED CONTROL SEPARATLY
EXCITED DC MOTOR



เกียรติอนันต์ สุขชู
KEITANUN SUKCHU

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

ศ.พ.

ก. ๔๕๕๔

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

๒๕๕๒

พ.ศ. 2542

เลขหมู่.....
เลขทะเบียน..... 34573
วัน, เดือน, ปี..... ๑๖ พ.ย. 2542

ISBN 974-622-566-9

๒ ๑๙๖๑๐๘
๑ ๑ ๑๖

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE DESIGN AND IMPLEMENTATION OF A THIRD ORDER PLANT
CONTROLLER BY USING PIDA FOR SPEED CONTROL SEPARATLY
EXCITED DC MOTOR**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

1999

ISBN 974-622-566-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 1999

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ การออกแบบและการสร้างตัวควบคุม PIDA สำหรับควบคุมกระบวนการ
อันดับสามเพื่อใช้ควบคุมความเร็วมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น
นักศึกษา นาย เกียรติอนันต์ สุขชู
รหัสประจำตัว 38061254
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา วิศวกรรมไฟฟ้า
พ.ศ. 2542
อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ. วิทยา ทิพย์สุวรรณพร

บทคัดย่อ

การควบคุมกระบวนการที่มีอันดับสามที่มีชนิดมากกว่าหนึ่ง เมื่อใช้ตัวควบคุมแบบเดิมนั้นจะให้ผลการตอบสนองช้า และมีโอเวอร์ชูตสูง อีกทั้งระบบควบคุมอาจจะไม่เสถียร เนื่องจากโหลดเปลี่ยนแปลง หรือมีสัญญาณรบกวนอื่นๆ วิทยานิพนธ์นี้จึงได้นำเสนอ เทคนิคการออกแบบ, การสร้าง และการประยุกต์ใช้งานตัวควบคุม Proportional, Integral, Derivative และ Acceleration (PIDA) เพื่อใช้กับระบบควบคุมดังกล่าว ทำให้ได้ผลตอบสนองในสถานะชั่วคราว และเข้าสู่สถานะคงตัวได้อย่างรวดเร็ว เช่น ใช้การควบคุมตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น เป็นกรณีศึกษา ตัวควบคุมนี้ออกแบบ ด้วยตัวประมวลผลสัญญาณเชิงเลข TMS320C31 ร่วมกับวงจรเชื่อมต่อที่ออกแบบเพื่อให้สามารถทำงานได้ตามเวลาจริง ระบบควบคุมที่สร้างขึ้นมานี้มีความยืดหยุ่นในการปรับแต่งค่าพารามิเตอร์ให้เหมาะสม ได้ตามต้องการ ผลการทดลองระบบควบคุมที่สร้างขึ้นแสดงให้เห็นว่ามอเตอร์สามารถเข้าสู่อัตราเร็ว และตำแหน่งตามที่กำหนด โดยใช้เวลาน้อยกว่า 2 วินาที ทั้งในสถานะมีโหลด และไม่มีโหลด ซึ่งได้ผลใกล้เคียงกับการจำลองระบบควบคุมด้วยโปรแกรม Matlab นอกจากนี้ยังได้ทำการจำลองการควบคุมตำแหน่งของมอเตอร์เหนี่ยวนำกระแสสลับแบบชนิดสามเฟส

Thesis Title The Design and Implementation of a Third Order Plant Controller by Using PIDA for speed control Separatly Excited DC Motor

Student Mr. Keitanun Sukchu

Student ID. 38061254

Degree Master of Engineering

Programme Electrical Engineering

Year 1999

Thesis Advisord II Assoc.Prof. Vittaya Tipsuwanporn

ABSTRACT

The conventional approach for the third order plant results a slow transient response and high overshoot. In addition, the system may return to be instability due to load variation or disturbance. This thesis proposes the design technique, implementation and application of the Proportional, Integral, Derivative and Acceleration (PIDA) controller to overcome this problem. The position controlling of the separately excited DC motor is considered to be a case study in thesis. The controller was implemented by using DSP processor TMS320C31 incorporated with the designed interfacing circuit to operate in a real time mode. The whole system parameters can be changed in the designed controller giving a flexibility for practitioners to adjust the appropriate parameters for their system. The experimental results reveal that the speed reference and position of the controlled motor can reach the setup value within 2 second both with and without load. These results approximate to those obtained form the simulation with Matlab program. The speed controlling of the three phase AC induction motor is also simulated with the designed controller which yields the satisfying results.

กิตติกรรมประกาศ

การจัดทำวิทยานิพนธ์ในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความเมตตากรุณาจากท่านอาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่ได้ให้การสนับสนุนทั้งในด้านอุปกรณ์เครื่องมือ และข้อมูลที่เป็นในการศึกษาวิจัยรวมถึงสถานที่ใช้ในการทดสอบ ผู้ทำวิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านทั้งหลาย และกราบขอบพระคุณเป็นอย่างสูง

ผู้ทำวิจัยขอขอบพระคุณท่านอาจารย์ รศ. วิทยา ทิพสุวรรณพร ซึ่งได้ให้คำแนะนำต่อผู้ทำวิจัยตลอดมา ผู้ทำวิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่าน และกราบขอบพระคุณเป็นอย่างสูงที่ได้ให้การสนับสนุนทุนในการศึกษาทำวิจัยจนสำเร็จตามวัตถุประสงค์

ขอขอบคุณบริษัท C & C คอมพิวเตอร์ และคณะบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้ให้ทุนในการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า

ขอขอบพระคุณ ผู้เชี่ยวชาญ และผู้ทรงคุณวุฒิ ที่ได้ให้คำแนะนำแนวทางในการทำวิทยานิพนธ์ตลอดจนขอขอบคุณเพื่อนๆ ทุกคน และขอขอบพระคุณ บิศา มารดา ที่ได้ให้ความช่วยเหลือ และให้กำลังใจต่อผู้ทำวิจัยตลอดมา

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกๆ ท่าน

เกียรติอนันต์ สุขชู

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
คำอธิบายสัญลักษณ์	XIII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	2
1.2 วัตถุประสงค์ของวิทยานิพนธ์	2
1.3 หลักการใหม่ที่นำมาใช้ในวิทยานิพนธ์	3
1.4 รายละเอียดในวิทยานิพนธ์	3
บทที่ 2 วิเคราะห์เครื่องจักรกลไฟฟ้ากระแสตรงแบบขดลวดสนามแยกตัวกระตุ้น	5
2.1 บทนำ	5
2.2 เครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกตัวกระตุ้น	6
2.2.1 (พิจารณาทรานเซียนในวงจรสนาม)	6
2.2.2 (พิจารณาทรานเซียนในวงจรอาร์เมเจอร์)	9
2.3 แบบจำลองทางพลศาสตร์ และการควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น	12
2.4 ผลกระทบต่อความเร็วรอบของมอเตอร์	21
บทที่ 3 หลักการของระบบควบคุมป้อนกลับ	22
3.1 ความจำเป็นที่ต้องการระบบควบคุมแบบป้อนกลับ	22
3.2 เปรียบเทียบการควบคุมแบบเปิดและการควบคุมแบบป้อนกลับ	24
3.3 ความไว	26
3.4 ชนิดของระบบควบคุมพื้นฐาน	29
3.4.1 (ตัวควบคุมเชิงสัดส่วน)	29
3.4.1.1 (การควบคุมโดยใช้ตัวควบคุมเชิงสัดส่วนกับระบบอันดับหนึ่ง)	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา IV จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.4.1.2 (การชดเชยแบบป้อนไปข้างหน้า).....	33
3.4.1.3 (ตัวควบคุมเชิงสัดส่วนเมื่อสัญญาณอินพุตเป็นฟังก์ชันแรมปี).....	34
3.4.1.4 (การควบคุมเชิงสัดส่วนกับระบบอันดับสอง).....	35
3.4.2 (การควบคุมแบบอินทิกรัล).....	36
3.4.2.1 (การควบคุมแบบอินทิกรัลกับระบบอันดับที่สอง).....	38
3.4.3 (การควบคุมแบบอนุพันธ์).....	39
3.4.4 (การควบคุมแบบสัดส่วนบวกกับอินทิกรัล).....	40
3.4.4.1 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลกับระบบอันดับที่หนึ่ง).....	40
3.4.4.2 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลกับระบบอันดับที่สอง).....	41
3.4.4.3 (การควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์กับระบบอันดับที่สอง).....	42
3.4.4.4 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลบวกกับแบบอนุพันธ์กับระบบ อันดับที่สอง).....	43
3.5 การเลือกค่าอัตราขยายของตัวควบคุม.....	44
บทที่ 4 การควบคุมแบบ Proportional, Integral, Derivative, Acceleration (PIDA).....	46
4.1 การออกแบบตัวควบคุม PIDA ด้วยวิธีพิจารณาคำรากหลัก Dominant Roots.....	46
4.2 ทฤษฎีและหลักการของ PIDA	47
4.3 ขั้นตอนการออกแบบตัวควบคุม PIDA.....	48
4.4 การจำลองแบบระบบควบคุมมอเตอร์กระตริงชนิดแยกตัวกระตุ้นด้วยตัวควบคุม PIDA...49	
4.4.1 (พิจารณาวงจร Zero Order Hold (ZOH)	55
4.6 บทสรุป	59
บทที่ 5. การออกแบบระบบควบคุมดิจิทัลด้วยตัวประมวลผลเชิงสัญญาณ.....	60
5.1 นิยามของการประมวลผลเชิงเลข.....	60
5.1.1 (ทฤษฎีการสุ่มตัวอย่าง).....	60
5.1.2 (บทนิยามของการแปลงแซด (Z-Transform)).....	61
5.2 สถาปัตยกรรมหน่วยประมวลผลสัญญาณเชิงเลข TMS320C31.....	62
5.2.1 (หน่วยประมวลผลกลาง Central process unit (CPU)).....	63
5.2.1.1 (หน่วยการคูณ Multiplier).....	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

5.2.1.2 (หน่วยคำนวณทางคณิตศาสตร์ Arithmetic Logic Unit (ALU)).....	63
5.2.1.3 (หน่วยช่วยคำนวณทางคณิตศาสตร์ Auxiliary Register Arithmetic Unit).....	63
5.2.1.4 (รีจิสเตอร์ของ CPU Register File).....	63
5.2.2 (การจัดหน่วยความจำ Memory Organization).....	65
5.2.2.1 (หน่วยความจำ RAM และ ROM).....	66
5.2.2.2 (ตารางหน่วยความจำ Memory Maps).....	66
5.2.2.3 (การจัดหน่วยความจำในโมดูล Memory Address Modes).....	67
5.2.3 (การติดต่อภายใน Internal Bus Operation).....	68
5.2.4 (การติดต่อภายนอก External Bus Operation).....	68
5.2.5 (ระบบติดต่อภายนอก Peripherals).....	71
5.2.6 (ระบบถ่ายโอนข้อมูล Direct Memory Access).....	72
5.2.7 (คำสั่งของไอซี TMS320C31).....	72
5.2.8 (แอสเซมบลี).....	72
5.2.9 (ดีบั๊กเกอร์).....	73
5.3 เซ็นเซอร์และเอนโคเดอร์ในระบบควบคุมมอเตอร์.....	74
5.3.1 (อินคริमेंท์เอนโคเดอร์).....	74
5.3.1.1 (ความละเอียดของอินคริमेंท์เอนโคเดอร์).....	75
5.3.1.2 (การตรวจจับความเร็วในการหมุนของอินคริमेंท์เอนโคเดอร์).....	75
5.4 การออกแบบโครงสร้างทางฮาร์ดแวร์.....	77
5.4.1 (วงจรตรวจจับทิศทางการหมุน).....	78
5.4.2 (วงจรมับ 32 บิต).....	79
5.4.3 (วงจรรินเตอร์เฟสโมดูล).....	80
5.5 การออกแบบซอฟต์แวร์.....	81
บทที่ 6 ผลการเขียนแบบและผลการทดลอง.....	84
6.1 การออกแบบควบคุมความเร็วโดยใช้ ดีซีชอปเปอร์ (dc chopper).....	86
6.2 การทดลองควบคุมด้วยไมโครคอนโทรลเลอร์.....	88
6.3 ผลการทดลองผลตอบสนองความเร็ว.....	89
6.4 การทดลองผลตอบสนองความเร็วต่อการกลับทิศทางการหมุนในขณะมีโหลด 0.9 N-m	

สารบัญ(ต่อ)

หน้า

และขณะไม่มีโหลด	91
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ	95
7.1 ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข	95
7.2 ข้อเสนอแนะและการพัฒนาการทำงานวิจัยต่อไป.....	96
7.3 การออกแบบตัวควบคุม PIDA สำหรับการควบคุมตำแหน่งมอเตอร์เหนี่ยวนำ กระแสสลับ.....	97
7.3.1(การออกแบบตัวควบคุม PIDA และ PIDA โรบัสต์).....	97
7.3.1.1(แบบจำลองการควบคุมความเร็ว และตำแหน่งของมอเตอร์เหนี่ยวนำ.....	98
บรรณานุกรม.....	105
ภาคผนวก	107
ก. รายละเอียดวงจรโดยรวมที่ใช้ในวิทยานิพนธ์.....	107
ข. ตัวอย่างการออกแบบตัวควบคุม PIDA กับกระบวนการอันดับสามชนิดต่างๆ.....	115
ค. โปรแกรมที่ใช้ในวิทยานิพนธ์ในการควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น.....	121
ผลงานทางวิชาการขณะศึกษาระดับบัณฑิตวิทยาลัย.....	131
ประวัติผู้เขียน	132

สารบัญภาพ

ภาพที่	หน้า
2.1 โครงสร้างของเครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกตัวกระตุ้น.....	6
2.2 ทรานเซียนในวงจรสนาม.....	8
2.3 ทรานเซียน และสภาวะทางไดนามิกส์ของวงจรราร์เมเจอร์เมื่อต่อโหลด.....	9
2.4 ผลรวมความสัมพันธ์ของกระแสต่อแรงดันขดลวดสนาม.....	11
2.5 ระบบมอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนามแยกตัวกระตุ้น.....	13
2. แบบจำลองทางพลศาสตร์ของมอเตอร์แบบแยกตัวกระตุ้น	18
2.4 บล็อกไดอะแกรมของระบบมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น	18
2.5 แสดงความสัมพันธ์ความเร็วรอบ และกำลังบิดเมื่อแรงดันไฟฟ้าที่ขั้วของอาร์มาเจอร์ มีค่าคงที่...21	
3.1 การควบคุมการทำงานของมอเตอร์กระแสตรง	23
3.2 ไดอะแกรมของการควบคุมความเร็วของมอเตอร์กระแสตรง	31
3.3 การควบคุมความเร็วในการหมุนของมอเตอร์โดยใช้ตัวควบคุมเชิงสัดส่วน.....	32
3.4 การควบคุมแบบป้อนไปข้างหน้า	34
3.5 การควบคุมตำแหน่งของการหมุนทางพลศาสตร์ของมอเตอร์กระแสตรง	36
3.6 ระบบควบคุมแบบป้อนกลับโดยใช้ตัวควบคุมแบบอินทิกรัล	38
3.7 การควบคุมมอเตอร์กระแสตรงด้วยตัวควบคุมแบบอินทิกรัล	38
3.8 การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลของมอเตอร์	40
3.9 การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PI	41
3.10 การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PD.....	42
3.11 การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PID	43
4.1 โครงสร้างของระบบควบคุมในวงปิดแบบทั่วไป.....	46
4.2 โลกัศที่ครอบคลุมรากที่ยอมรับได้และตำแหน่งค่ารากในวงปิดที่ตำแหน่ง (*) ใน s-plane	48
4.3 ระบบควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นด้วยตัวควบคุม PIDA	50
4.4 รูดโลกัศจากค่าพารามิเตอร์ทางพลศาสตร์ของระบบมอเตอร์.....	51
4.5 รูดโลกัศที่ได้จากการแก้สมการคุณลักษณะตามวิธีการของ PIDA	51
4.6 ผลตอบสนองของตัวควบคุม PIDA สำหรับมอเตอร์แบบแยกตัวกระตุ้น	55
4.7 ไดอะแกรมของระบบควบคุมแบบสัญญาณไม่ต่อเนื่อง.....	57
4.8 ไดอะแกรมจำลองการควบคุมแบบป้อนไปข้างหน้า.....	57
4.9 ผลการวิเคราะห์เสถียรภาพของระบบเวลาไม่ต่อเนื่อง.....	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
6.6 ไคอะแกรมของระบบควบคุมมอเตอร์โดยใช้หน่วยประมวลผลสัญญาณเชิงเลข.....	89
6.7 เมื่อปรับความเร็วที่ 2250 rpm และลดความเร็วลงอย่างฉับพลันไปเป็น 1000 rpm	89
6.8 เมื่อความเร็วที่ 2250 rpm และต่อ โทลคขนาด 1.2 Nm	90
6.9 เมื่อความเร็วที่ 2250 rpm เปลี่ยนความเร็วอย่างฉับพลันไปเป็น 1000 rpm และต่อ โทลค 1.2 N-m	90
6.10 การวัดผลตอบสนองของเวลาขณะเข้าสู่ในสภาวะคงตัว.....	91
6.11 ผลตอบสนองของเวลาในสภาวะคงตัวเมื่อเพิ่มเกนที่ K	91
6.12 ผลตอบสนองของเวลาในสภาวะคงตัวเมื่อกลับทิศทางการหมุน.....	92
6.13 ผลการเขียนแบบที่เอาท์พุตต่อสัญญาณรบกวนค่าต่างๆ	93
6.14 เครื่องต้นแบบที่สร้างขึ้น	93
6.15 วงจรนับ 32 บิต วงจรตรวจจับทิศทางการหมุน และวงจรอินเทอร์เฟสโมดูล.....	94
6.16 ภาพด้านหน้าขณะลงกล่อง.....	94
6.17 ภาพระบบโดยรวม	94
7.1 โครงสร้างของระบบควบคุม PIDA ในวงปิดแบบทั่วไปที่จัดใหม่.....	97
7.2 ไคอะแกรมโดยรวมของระบบควบคุม PIDA โรบัสต์ สำหรับมอเตอร์เหนี่ยวนำ.....	98
7.3 การควบคุมมอเตอร์เหนี่ยวนำกระแสสลับด้วยตัวควบคุม PIDA โรบัสต์ โดยการชดเชยแบบ ป้อนไปข้างหน้า	99
7.4 โบลในวงรอบปิดของตัวควบคุม PIDA โรบัสต์	101
7.5 เส้นคอนทัวร์ในควิสต์ ในวงรอบปิด.....	101
7.6 ผลการตอบสนองของฟังก์ชันขั้นหนึ่งหน่วย เมื่อเพิ่มค่าอัตราขยายเกนที่ $K = 1, 20, 50, 100$ 1000 และ 10000 ตามลำดับ	102
7.7 ผลการวิเคราะห์เสถียรภาพโดยใช้รูดโกลด์.....	103
7.8 เปรียบเทียบผลการตอบสนองของฟังก์ชันขั้นหนึ่งหน่วยของตัวควบคุม PIDA กับตัวควบคุม PIDA โรบัสต์	103
7.9 เปรียบเทียบผลการตอบสนองของเอาท์พุตต่อสัญญาณรบกวนเมื่อเพิ่มค่าเกนที่ K ของ ตัวควบคุม PIDA โรบัสต์	104
1(ก) วงจรนับ 32 บิต และวงจรตรวจจับทิศทางการหมุน	107
2(ก) วงจรอินเทอร์เฟสโมดูล 32 บิต กับ TMS321C31	108
3(ก) โมดูลหน่วยประมวลผลสัญญาณเชิงเลข	109
4(ก) วงจรแหล่งจ่ายไฟสำหรับโมดูล TMS320C31	110

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
5(ก) วงจรภายในของโมดูล TMS320C31	111
6(ก) วงจรแปลงสัญญาณ A/D และ D/A ขนาด 14 บิต ของ TLC32040	112
7(ก) วงจรพอร์ที่เชื่อมต่อกับไมโครคอมพิวเตอร์ทางพอร์ที่ขนานของ TMS320C31	113
8(ก) พอร์ที่เชื่อมต่อภายนอกของโมดูล TMS320C31	114
1(ข) ผลตอบสนองของตัวควบคุม PIDA ทั้งสามควบคุมกระบวนการอันดับสามชนิด Type 0	116
2(ข) ผลของตัวควบคุม PIDA ควบคุมกระบวนการชนิด Type 1 หลังจากเพิ่มเกนซ์ K	118
3(ข) ผลของตัวควบคุม PIDA ควบคุมกระบวนการชนิด Type 2 หลังจากเพิ่มเกนซ์ K	119



สารบัญตาราง

ตารางที่	หน้า
4.1 ค่าพารามิเตอร์ของมอเตอร์ที่ใช้จำลองแบบ.....	49
4.2 ค่าพารามิเตอร์ที่ได้จากการคำนวณ.....	50
4.3 แสดงผลของตัวควบคุม ที่มีค่า R และเกณฑ์ต่างกันสำหรับการจำลองแบบมอเตอร์กระแสตรง แบบแยกตัวกระตุ้น	54
5.2.1 ชื่อและหน้าที่ต่างๆ ของรีจิสเตอร์ใน CPU	65
5.2.2 แสดงหน้าที่ต่างๆ ของบัสควบคุมรีจิสเตอร์ปฐมภูมิ.....	70
5.2.3 แสดงหน้าที่ต่างๆ ของบัสควบคุมขยายรีจิสเตอร์.....	70
5.4.1 แสดงสถานะของขาสัญญาณ.....	80
6.1 ผลการควบคุมกระแสและแรงดันต่อความเร็วเมื่อจำกัดกระแสอินพุทขณะต่อโหลด.....	90
1(ข) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการอันดับสามชนิด type 0	116
2(ข) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการ อันดับสามชนิด type 1.....	117
3(ข) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการอันดับสามชนิด type 2	119

คำอธิบายสัญลักษณ์

e_u = แรงดันไฟฟ้าด้านกลับ

i = กระแสไฟฟ้า

i_u = กระแสไฟฟ้าไหลเข้าวงจรอาร์เมเจอร์

i_f = กระแสไฟฟ้าไหลเข้าวงจรสร้างสนามแม่เหล็ก

i_ϕ = กระแสไฟฟ้าแม่กนิโตซิงค์ หรือกระแสการทำแม่เหล็ก

n = ความเร็วรอบ

(t) = โดเมนทีในเชิงเวลา

T_d = กำลังบิดที่มอเตอร์สร้างขึ้น

T_L = กำลังบิดที่โหลดต้องการ

t_q = ค่าเวลาที่ใช้ในการหยุดทำงานของ SCR (Turn-off Time)

t_{SM} = ค่าเวลาคงที่ของวงจรกรองสัญญาณความถี่ต่ำผ่าน

u = แรงดันไฟฟ้าที่ขั้วอาร์เมเจอร์ของมอเตอร์

$v_c(t)$ = แรงดันไฟฟ้าควบคุม

v_f = แรงดันไฟฟ้าที่ป้อนเข้าวงจรสร้างสนามแม่เหล็ก

$v_{(n)}$ = แรงดันไฟฟ้าที่จุด n เปรียบเทียบกับจุด GND

R_u = ความต้านทานที่วงจรอาร์เมเจอร์

L_{aq} = ค่าความเหนี่ยวนำที่อาร์เมเจอร์

τ_u = ค่าเวลาคงตัวของไฟฟ้า

τ_m = ค่าเวลาคงตัวเชิงกล

e_{mo} = แรงเคลื่อนที่ตกคร่อมขณะไม่มีโหลด

K_m = ค่าคงที่ของมอเตอร์

ω_{mo} = ความเร็วเชิงมุมขณะไม่มีโหลด

T = แรงบิดที่เกิดจากสนามแม่เหล็ก

J = ความเฉื่อยของระบบ

T_L = แรงบิดของโหลดที่มีทิศทางตรงกันข้ามกับทิศทางหมุนของมอเตอร์

B = ค่าความหนืด

J_m = โมเมนของมอเตอร์

ξ = แคมป์ปิ้งของการหน่วง

s_1, s_2 = ค่ารากในครึ่งระนาบด้านซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา XIII องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายสัญลักษณ์ (ต่อ)

ω_n = ความเร็วเชิงมุมโดยธรรมชาติ

α = มุมกระตุ้นที่เฟกเตอร์แคมป์ปิ้ง

ω = ความเร็วเชิงมุม

ω_m = ความเร็วเชิงมุมของมอเตอร์

ω_d = ค่าตั้งความเร็วเชิงมุมที่ฟังก์ชันขั้นหนึ่งหน่วย

θ_m = ค่าความเร็วต่อตำแหน่งของมอเตอร์

θ_d = ค่าตั้งความเร็วต่อตำแหน่งของมอเตอร์ที่ฟังก์ชันขั้นหนึ่งหน่วย



บทที่ 1

บทนำ

ปัจจุบันการใช้งานระบบขับเคลื่อนมอเตอร์ไฟฟ้าแบบควบคุมความเร็วและตำแหน่งได้เข้ามามีบทบาทในโรงงานอุตสาหกรรมอย่างกว้างขวาง สามารถแยกประเภทตามชนิดของมอเตอร์ซึ่งใช้เป็นโรตอร์กำลังคือ ระบบที่ใช้มอเตอร์ไฟฟ้ากระแสสลับเป็นต้นกำลัง และระบบที่ใช้มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลัง ซึ่งในระบบแรกแม้ว่าราคาของมอเตอร์ และค่าใช้จ่ายในการบำรุงรักษาของมอเตอร์จะต่ำกว่าระบบหลังมากแต่อุปกรณ์สำหรับขับและควบคุมการทำงานของมอเตอร์ระบบแรกให้ทำงานได้แม่นยำ จะมีเทคนิคที่ยุ่ยยาก และราคาสูง ดังนั้นเมื่อมีความต้องการระบบขับเคลื่อนซึ่งมีความจำเป็นที่จะต้องมีความสมบัติในการทำงานที่แม่นยำสูง ระบบขับเคลื่อนที่ใช้มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังจึงได้รับความนิยมมากกว่าระบบแรก โดยเฉพาะมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น ทั้งนี้เพราะการควบคุมแรงดันไฟฟ้าในส่วนของวงจรรีเมเจอร์ และการควบคุมปริมาณเส้นแรงสนามแม่เหล็กภายในมอเตอร์สามารถแยกการควบคุมให้เป็นอิสระต่อกัน ซึ่งทำให้มอเตอร์ชนิดนี้ง่ายต่อการควบคุม โดยแต่เดิมนั้นจะใช้ตัวควบคุม PID ในการควบคุม ซึ่งสามารถสร้างขึ้นได้ง่ายสำหรับกระบวนการที่เป็นอันดับสอง ส่วนในกรณีระบบที่มีกระบวนการสูงขึ้นนั้นจะควบคุมได้ยาก เนื่องจากอันดับของกระบวนการ ที่มีจำนวนโพลมากกว่าจำนวนซีโรว์ และเพื่อต้องการออกแบบตัวควบคุมสำหรับกระบวนการที่มีอันดับสูงๆ จึงได้มีการพัฒนาที่จะนำเสนอการออกแบบตัวควบคุมให้มีเสถียรภาพและความแม่นยำสูง สำหรับควบคุมตำแหน่งการหมุนของมอเตอร์ เพื่อทำให้มีผลตอบสนองสภาวะของการเปลี่ยนแปลงตำแหน่งในขณะที่มอเตอร์ทำงานได้อย่างรวดเร็ว จึงได้นำเสนอวิธีการออกแบบตัวควบคุม PIDA ซึ่งให้ผลการตอบสนองในสภาวะชั่วคราว และสภาวะคงตัวมีความเที่ยงตรงสูง ในการใช้มอเตอร์เป็นต้นกำลังในโรงงานอุตสาหกรรม หรือใช้ขับเคลื่อนรถไฟ และแขนกลในอุตสาหกรรม ฯลฯ โดยเหตุที่เรามีความต้องการว่าวิธีการควบคุมจะต้องเป็นวิธีประหยัดมีราคาไม่แพง สามารถควบคุมอัตราเร็วของมอเตอร์ได้ต่อเนื่องแม่นยำ และยังสามารถเปลี่ยนความเร็วได้อย่างรวดเร็ว มีพิสัยของอัตราเร็วตามต้องการ และมีเสถียรภาพที่ดีมีความคงทน ในอดีตระบบควบคุมความเร็วรอบไฟฟ้ากระแสตรงอาจจะถูกสร้างขึ้นมาได้หลายวิธีและหลายรูปแบบ เช่น ระบบควบคุมความเร็วรอบโดยควบคุมปริมาณเส้นแรงสนามแม่เหล็กภายในมอเตอร์ ซึ่งระบบควบคุมความเร็วดังกล่าวนั้นขึ้นอยู่กับโหลด ทำให้มีการตอบสนองต่อการทำงานช้า มีขนาดใหญ่และราคาแพง ปัจจุบันอุปกรณ์อิเล็กทรอนิกส์กำลัง ได้รับการพัฒนามากขึ้นทำให้เกิดการพัฒนาจอร์อิเล็กทรอนิกส์กำลัง เพื่อนำไปใช้ควบคุมกำลังขนาดหลายๆ และเมื่อนำจอร์อิเล็กทรอนิกส์กำลังมาสร้างเป็นระบบควบคุมมอเตอร์ ก็จะทำให้การทำงานของระบบนี้สามารถตอบสนองต่อการทำงานได้เร็วกว่า ขนาดเล็ก และราคาถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 กล่าวนำ

พิจารณาตัวควบคุม PIDA (proportional-integral-derivative-acceleration) เป็นการนำเสนอสำหรับกระบวนการอันดับสาม [12] โดยที่ตัวควบคุม PIDA นี้ประกอบด้วย 3 โพล และ 3 ซีโร่ แต่อาจจะมี 2 โพลก็ได้ ในการออกแบบตัวควบคุมกระบวนการทำได้โดยการเพิ่ม ซีโร่ ที่ตัวควบคุม PID จะได้ตัวควบคุม PIDA กรณีที่เพิ่มซีโร่ เข้าไปที่ตัวควบคุม PID นั้นเป็นการเปลี่ยนตำแหน่งค่าราก รูดโลกซ์ของกระบวนการอันดับสาม ที่ครอบคลุมค่ารากที่มีมากกว่าโดยการกำจัดผลที่เกิดจากการที่ไม่ครอบคลุมค่าราก ในวิทยานิพนธ์ฉบับนี้ยังได้กล่าวถึงวิธีการวิเคราะห์ที่เป็นไปได้สำหรับการนำไปพิจารณาการออกแบบตัวควบคุม PIDA ที่ตำแหน่งของค่ารากที่หนึ่ง และค่ารากที่สอง ของกระบวนการ หากต้องการให้เส้นแกนค่าจริงอยู่ต่ำกว่าที่ครอบคลุมรากโดยห่างจากจุดกำเนิดเล็กน้อยในเส้นแกนจริงจะได้สมการคุณลักษณะใหม่ สมการคุณลักษณะที่ได้ใหม่นี้เป็นฟังก์ชันถ่ายโอนในวงปิด สำหรับตัวควบคุม PIDA และได้จากโครงสร้างของกระบวนการอีกด้วย สมการคุณลักษณะของตัวชดเชยและสมการที่สมบูรณ์ของระบบได้กล่าวไว้ใน บทที่ 4. ซึ่งสามารถแก้สมการโดยพีชคณิตเพื่อหาค่าอัตราขยายสำหรับตัวควบคุม PIDA ผลที่ได้จากการแก้สมการตามวิธีการของ PIDA จะทำให้ได้ผลตอบสนองตามเงื่อนไขของระบบควบคุม โดยที่ผลกระทบจากการรบกวนที่ตัวกระบวนการสามารถทำให้ลดน้อยลงได้โดยต่อ แคชเคตเกนซ์ K ที่เพิ่มขึ้น นอกจากนี้ในวิทยานิพนธ์ยังได้แสดงตัวอย่างการออกแบบตัวควบคุม PIDA สำหรับกระบวนการอันดับสามในลักษณะชนิดต่างๆ ที่มีความแตกต่างกันออกไปไว้ในภาคผนวก (ข) และเพื่อเป็นการนำเสนอวิธีการทดสอบตัวควบคุม PIDA สำหรับควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น และเนื่องจากอุปกรณ์อิเล็กทรอนิกส์กำลังในปัจจุบันสามารถให้กำลังขับเคลื่อนและแรงดันไฟฟ้าได้สูง จึงทำให้การควบคุมมอเตอร์ไฟฟ้ากระแสตรงที่ต้องการกำลังขับมากๆ ทำได้ไม่ยุ่งยากมากนัก ดังนั้นจึงได้นำตัวควบคุมดังกล่าวมาควบคุมอุปกรณ์อิเล็กทรอนิกส์กำลัง ที่จ่ายกำลังให้กับมอเตอร์ได้อย่างเหมาะสม และเพื่อนำมาพิจารณาควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้นเป็นกรณีศึกษา และเพื่อวิเคราะห์ระบบควบคุมจากการจำลองระบบควบคุมตำแหน่ง ด้วยตัวควบคุม PIDA ซึ่งจะได้กล่าวต่อไป

1.2 วัตถุประสงค์ของวิทยานิพนธ์

เพื่อใช้ในการควบคุมการทำงานของอุปกรณ์ควบคุมความเร็วและตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น โดยอาศัยความสัมพันธ์ของฟลักซ์แม่เหล็ก และแรงบิดของมอเตอร์โดยตรง ในการควบคุมด้วยวิธี PIDA นี้ตัวควบคุมมอเตอร์จะทำหน้าที่คอยตรวจสอบการเปลี่ยนแปลงของกระแส และแรงดันที่จ่ายให้กับมอเตอร์ในทุกๆ 30 ไมโครวินาที เพื่อควบคุมแรงดันไฟฟ้าที่วงจรอาร์เมเจอร์ ซึ่งโดยปกติแล้วมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นนั้นจะกำหนดให้กระแสและแรงดันของวงจรมอเตอร์นั้นตามพิกัดคงที่ ซึ่งแยกเป็นอิสระต่อกัน ทำให้การควบคุมง่าย

เอกสารนี้เป็นเอกสารทงสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้มาใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังผลให้แรงดันไฟฟ้าและการควบคุมปริมาณเส้นแรงแม่เหล็กที่จ่ายให้กับขดลวดสนามของมอเตอร์ได้ตามพิกัด ในขณะที่แรงบิดของภาระมีค่าสูงสุด ตัวควบคุม PIDA จะเร่งให้ผลตอบสนองต่อการควบคุมความเร็ว และตำแหน่งได้อย่างรวดเร็วสอดคล้องกับภาระ และเพื่อต้องการทำให้ผลตอบสนองในสภาวะชั่วคราว และในสภาวะคงตัวทำได้อย่างรวดเร็ว ทำให้สามารถควบคุมมอเตอร์ซึ่งต้องการผลตอบสนองที่เร็ว โดยให้ประสิทธิภาพในการทำงานเป็นเชิงเส้น ทั้งในสภาวะมีโหลด และขณะไม่มีโหลด นอกจากนี้ยังได้กล่าวถึงการประยุกต์ใช้งานตัวประมวลผลสัญญาณเชิง (DSP) เบอร์ TMS320C31 ซึ่งมีคำสั่งที่อำนวยความสะดวกในการทำงานทางคณิตศาสตร์ได้โดยตรง

1.3 หลักการใหม่ที่นำมาใช้ในวิทยานิพนธ์

เทคนิคที่นำมาใช้วิเคราะห์คือการประยุกต์ใช้งานตัวควบคุม PID และ กราฟรูตโลกัส ที่มีอยู่เดิม โดยการเพิ่มซีโรเข้าไปที่ตัวควบคุม PID อีกจำนวนหนึ่งจะทำให้ได้ตัวควบคุม PIDA ด้วยวิธี Dominant Roots มาใช้ในการออกแบบตัวควบคุมให้เหมาะสมกับแบบจำลองกระบวนการของมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น ทำให้เปอร์เซ็นต์โอเวอร์ชูทลดลง และยังทำให้เข้าสู่สภาวะคงตัวได้อย่างรวดเร็ว ซึ่งก็หมายถึงการทำให้องค์ประกอบแรงดันไฟฟ้าที่วงจรรีเมเจอร์ได้ตามการเปลี่ยนแปลงของโหลด ซึ่งใช้การคำนวณด้วยวิธีการตามอัลกอริทึม PIDA ที่ใช้หน่วยประมวลผลสัญญาณ TMS320C31 เป็นหน่วยคำนวณทางคณิตศาสตร์ ทำให้สามารถคำนวณหาค่าความเร็วรอบ และตำแหน่งของเพลามอเตอร์ตลอดช่วงความถี่ในการใช้งาน ได้อย่างถูกต้องและแม่นยำ

1.4 รายละเอียดในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์นี้ได้แบ่งออกเป็น 7 บท ซึ่งในแต่ละบทมีเนื้อหาดังนี้คือ

บทที่ 1 กล่าวถึงความจำเป็นของวิทยานิพนธ์ และอธิบายถึงวัตถุประสงค์ของวิทยานิพนธ์ รวมถึงหลักการใหม่ที่นำมาใช้ สุดท้ายได้กล่าวถึงรายละเอียดโดยรวมของแต่ละบท

บทที่ 2 อธิบายถึง การวิเคราะห์เครื่องจักรกลไฟฟ้ากระแสตรงแบบขดลวดสนามแยกตัวกระตุ้น โดยอธิบายถึง สภาวะชั่วคราว และสภาวะทางพลศาสตร์ของเครื่องจักรกลไฟฟ้ากระแสตรงแบบขดลวดสนามแยกกระตุ้น ท้ายกล่าวถึงแบบจำลองทางพลศาสตร์ และการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น โดยอธิบายทารนเฟอร์ฟังก์ชันของระบบที่นำมาศึกษา และผลกระทบต่อความเร็วรอบของมอเตอร์ชนิดดังกล่าว

บทที่ 3 อธิบายถึงหลักการของระบบควบคุมป้อนกลับแบบดั้งเดิม โดยอธิบายถึงความจำเป็นที่ต้องการระบบควบคุมแบบป้อนกลับ ทั้งยังมีการเปรียบเทียบระหว่างการควบคุมแบบเปิด และการควบคุมแบบป้อนกลับ รวมถึงการวิเคราะห์หาค่าความไวของระบบควบคุมซึ่งมีประโยชน์ต่อระบบควบคุมว่าความไวมีผลต่อการเปลี่ยนแปลงของค่าพารามิเตอร์ต่างๆ และยังได้กล่าวถึงชนิดของ

ระบบควบคุมพื้นฐาน ที่ใช้กับกระบวนที่มีอันดับหนึ่ง และอันดับสอง ทั้งยังกล่าวถึงการเลือกอัตราขยายของตัวควบคุม และเกณฑ์ครรชนีสมรรถนะของระบบควบคุม

บทที่ 4 อธิบายถึงการออกแบบตัวควบคุม PIDA ซึ่งเป็นหลักการใหม่ที่นำมาใช้ในวิทยานิพนธ์นี้ โดยกล่าวถึง การออกแบบตัวควบคุมสำหรับหลักกระบวนการที่มีอันดับสาม ตามทฤษฎีและหลักการของตัวควบคุม PIDA ขั้นตอนการออกแบบตัวควบคุม PIDA ดังนั้นจึงได้นำมาออกแบบระบบควบคุมมอเตอร์ดีซีโดยการจำลองแบบของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น ซึ่งมีกระบวนการเป็นอันดับสาม และได้ยกตัวอย่างในการออกแบบตัวควบคุม PIDA กับกระบวนการอันดับสาม โดยยกตัวอย่างกรณีที่กระบวนการเป็นชนิดที่หนึ่ง ชนิดที่สอง และชนิดที่สามไว้ในภาค กนวก(ข) ซึ่งได้เรียบเรียงใหม่จาก [12] และ [13] เพื่อให้เข้าใจถึงประโยชน์ที่จะได้รับจากการออกแบบตัวควบคุม PIDA กับกระบวนการอันดับสามดังกล่าว

บทที่ 5 อธิบายถึงการออกแบบระบบควบคุมดิจิทัลด้วยหน่วยประมวลผลเชิงสัญญาณ โดยกล่าวถึงนิยามของการประมวลผลเชิงเลขซึ่งเป็นการอธิบายทฤษฎีการสุ่มตัวอย่าง และบทนิยามการแปลงแซด และกล่าวถึงรายละเอียดสถาปัตยกรรมหน่วยประมวลผลสัญญาณเชิงเลข TMS320C31 ซึ่งอธิบายโครงสร้างภายในของ CPU และการจัดการ Memory รวมถึงการติดต่อภายนอกและระบบถ่ายโอนข้อมูลรวมถึงคำสั่งของ ไอซี TMS320C31 นอกจากนี้ยังได้กล่าวถึงเซ็นเซอร์และเอนโคเดอร์ที่ใช้ในระบบควบคุมมอเตอร์ และการทำงานของอินทรีเมนต์เอนโคเดอร์ นอกจากนี้ได้กล่าวถึง แอสแซมเบลอร์ และดีบั๊กเกอร์ ที่เป็นตัวช่วยในการเขียนโปรแกรม สุดท้ายกล่าวถึงโครงสร้างทางฮาร์ดแวร์ที่สร้างขึ้น ซึ่งได้อธิบายหลักการการทำงานของแต่ละวงจร รวมถึงไดอะแกรมการออกแบบซอฟต์แวร์

บทที่ 6 อธิบายถึงการทดลอง และผลการทดลอง โดยทดลองควบคุมความเร็วขณะโหลดเปลี่ยนแปลงว่าตัวควบคุมที่สร้างขึ้นสามารถควบคุมได้ตามเป้าหมายหรือไม่ และทดลองผลการควบคุมกระแสที่จ่ายให้กับมอเตอร์โดยการกำหนดกระแสอ้างอิง และผลการทดลองในการควบคุมมอเตอร์ โดยการกำหนดความเร็วอ้างอิงตามที่ต้องการ สุดท้ายเป็นการเปรียบเทียบผลการทดลองกับการเทียบแบบจากการจำลองระบบเป็นเชิงเลขขณะต่อโหลดที่ค่าต่างๆ เพื่อวิเคราะห์ว่าผลที่ได้จากการออกแบบระบบควบคุมมอเตอร์นั้นมีความเหมาะสมมากน้อยเพียงใด

บทที่ 7 เป็นการกล่าวโดยสรุปผล และข้อเสนอแนะ ของวิทยานิพนธ์ฉบับนี้ และได้ทำการจำลองการควบคุมมอเตอร์เหนี่ยวนำสามเฟส โดยใช้การออกแบบตัวควบคุม PIDA เป็นแบบโรบัสต์ด้วย ทำยสุดของวิทยานิพนธ์ฉบับนี้ได้แสดงรายละเอียดของวงจรที่ได้ออกแบบ และวงจรของหน่วยประมวลผลเชิงเลขที่นำมาประยุกต์ใช้ และตัวอย่างการออกแบบตัวควบคุม PIDA กับกระบวนการอันดับสามชนิดต่างๆ รวมถึงโปรแกรมที่ใช้ในวิทยานิพนธ์ และงานวิจัยในระหว่างการศึกษาปริญญาโทที่ได้รับการตีพิมพ์เผยแพร่ ทางวิชาการสาขาวิศวกรรมไฟฟ้า ไว้ในตอนท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

วิเคราะห์เครื่องกลไฟฟ้ากระแสตรงแบบขดลวดสนามแยกกระตุ้น

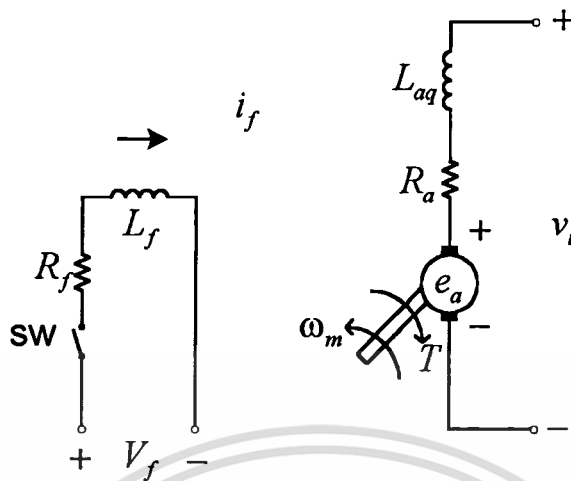
2.1 บทนำ

โดยทั่วไปดีซีมอเตอร์สามารถจำแนกออกเป็นสามแบบคือแบบกระตุ้นอนุกรม แบบกระตุ้นขนาน และแบบแยกตัวกระตุ้น แต่อย่างไรก็ตาม คุณลักษณะของมอเตอร์สองแบบแรกไม่เป็นเชิงเส้นดังนั้นในการนำไปประยุกต์ใช้งานในระบบควบคุมนั้นจึงนิยมใช้แบบที่สามดังในหนังสืออ้างอิงที่ [5] ซึ่งในการวิจัยนี้ได้นำมามอเตอร์กระแสตรงแบบแยกตัวกระตุ้น ซึ่งเครื่องจักรกลไฟฟ้ากระแสตรงสามารถควบคุมได้ง่ายและการทำงานของส่วนใหญ่ จะเป็นการควบคุมความเร็วเนื่องจากสามารถควบคุมได้ในช่วงกว้าง นอกจากนี้ยังประยุกต์ในงานควบคุมตำแหน่ง เช่น ระบบขับเคลื่อนเซอร์โว (Servo drives) และเพื่อให้ง่ายต่อการวิเคราะห์และลดความยุ่งยาก การพิจารณาจะกำหนดองค์ประกอบของระบบดังนี้

1. ไม่คิดผลการอิ่มตัวทางแม่เหล็ก (Magnetic saturation) โดยสมมติว่าค่าความเหนี่ยวนำไม่ขึ้นอยู่กับกระแส
2. สมมติว่าแรงเคลื่อนแม่เหล็กเนื่องจากขดลวดสนามและเนื่องจากอาร์เมเจอร์เกิดขึ้นตามแนวแกน d และแนวแกน q ด้วยเหตุผลดังกล่าวนี้จึงไม่เกิดค่าความเหนี่ยวนำร่วมระหว่างวงจรถ่านและวงจรรอาร์เมเจอร์ และที่สำคัญจากเหตุผลดังกล่าวจะไม่ทำให้ผลของความเป็นสนามแม่เหล็กลดลง (demagnetizing effect) ที่เกิดจากสนามแม่เหล็กสะท้อนจากอาร์เมเจอร์ (armature reaction)

ในส่วนแรกจะเป็นการพิจารณาสถานะทรานเซียนของเครื่องกำเนิดไฟฟ้ากระแสตรง เพื่อศึกษาถึงเหตุผลที่จะนำไปวิเคราะห์พฤติกรรมทางไดนามิกส์ของมอเตอร์ไฟฟ้ากระแสตรงโดยจะพิจารณา และวิเคราะห์จากสมการและทรานเฟอร์ฟังก์ชัน ที่เป็นความสัมพันธ์ระหว่างตัวแปรทางด้านเอาต์พุตต่ออินพุตที่ได้จากผลการตอบสนองของระบบซึ่งเกิดจากอินพุตมีการเปลี่ยนแปลง ในกรณีนี้จะต้องให้ขดสนามสนามของมอเตอร์มีค่าคงที่เพื่อให้การทำงานเป็นเชิงเส้นและเพื่อนำมาพิจารณาถึงวิธีการออกแบบการควบคุมความเร็วและแรงบิดให้มีความเที่ยงตรงสูง ดังนั้นเมื่อทำการออกแบบตัวควบคุม PIDA จะทำให้ง่ายต่อการควบคุมความเร็ว และการควบคุมตำแหน่งของมอเตอร์ซึ่งจะได้กล่าวในตอนต่อไป

2.2 เครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกกระตุ้น



ภาพที่ 2.1 แสดงโครงสร้างของเครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกกระตุ้น

โครงสร้างของเครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกกระตุ้นแสดงดังภาพที่ 2.1 ซึ่งค่าความเหนี่ยวนำของอาร์มเจอร์แสดงตามแนวแกน q (L_{aq}) เนื่องจากแรงเครื่องแม่เหล็กของอาร์มเจอร์เกิดขึ้นตามแนวแกน q สำหรับสมการพื้นฐานของเครื่องจักรกลไฟฟ้ากระแสตรงกำหนดดังนี้ คือ

$$e_a = K_a \Phi \omega_m \quad (2.1)$$

$$T = K_a \Phi i_a \quad (2.2)$$

สมมุติว่าสนามแม่เหล็กเปลี่ยนแปลงเป็นเชิงเส้น $K_a \Phi = K_f i_f$ ดังนั้นสมการ (2.1) และ (2.2) ดังนั้นเขียนใหม่ได้ดังนี้ คือ

$$e_a = K_f i_f \omega_m \quad (2.3)$$

$$T = K_f i_f i_a \quad (2.4)$$

2.2.1 พิจารณาทรานเซียนในวงจรสนาม

เมื่อพิจารณาทรานเซียนที่เกิดขึ้นในเครื่องกำเนิดไฟฟ้าแบบขดลวดสนามแยกกระตุ้น ประการแรกที่จะศึกษาคือผลลัพธ์จากการเปลี่ยนแปลงเนื่องจากการกระตุ้นแรงดันให้กับวงจรสนาม สำหรับวงจรอาร์มเจอร์เปิดวงจร และเครื่องกำเนิดหมุนด้วยความเร็วคงที่ ω_m จากภาพที่ 2.1 สมการแรงดันของวงจรสนามหลังจาก สวิตช์ปิด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_f = R_f I_f + L_f \frac{di_f}{dt} \quad (2.5)$$

ดังนั้นผลการแปลงลาปลาซของสมการ (2.5) เมื่อกำหนดเงื่อนไขของสภาวะต้นเป็นศูนย์ คือ

$$\begin{aligned} V_f(s) &= R_f I_f(s) + L_f s I_f(s) \\ &= I_f(s) (R_f + s L_f) \end{aligned} \quad (2.6)$$

สำหรับความสัมพันธ์ของทรานเฟอร์ฟังก์ชันระหว่างกระแสต่อแรงดันของขดลวดสนามคือ

$$\frac{I_f(s)}{V_f(s)} = \frac{1}{R_f + s L_f} = \frac{1}{R_f (1 + s \tau_f)} \quad (2.7)$$

เมื่อ $\tau_f = \frac{L_f}{R_f}$ คือ ค่าเวลาคงตัวของวงจรถนอม

สำหรับแรงดันที่สร้างขึ้นในวงจรถนอมเมเจอร์จากสมการ(2.3) คือ

$$e_s = K_f i_f \omega_m = K_g \omega_m \quad (2.8)$$

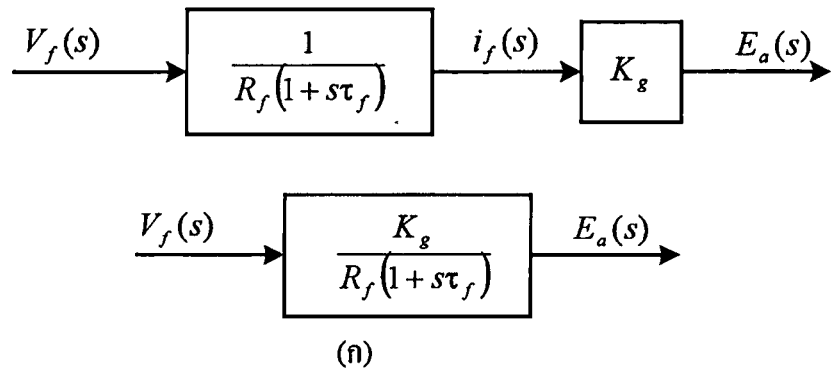
เมื่อ $K_g = K_f \omega_m$ คือความชันของส่วนที่เกิดการเปลี่ยนแปลงแบบเชิงเส้น เส้นโค้งการทำแม่เหล็ก ที่ความเร็ว ω_m แสดงได้จากการพล็อตค่า e_s กับ i_f สำหรับการแปลงลาปลาซของสมการ (2.8) คือ

$$E_s(s) = K_g I_f(s) \quad (2.9)$$

จากสมการ (2.7) และ (2.9) ทรานเฟอร์ฟังก์ชันที่แสดงความสัมพันธ์ระหว่างแรงดันอาร์เมเจอร์ต่อแรงดันวงจรถนอมคือ

$$\frac{E_s(s)}{V_f(s)} = \frac{E_s(s)}{I_f(s)} \cdot \frac{I_f(s)}{V_f(s)} = \frac{K_g}{R_f (1 + s \tau_f)} \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.2 ทรานเซียนในวงจรมอเตอร์ (ก) บล็อกไดอะแกรม (ข) ผลการตอบสนอง

จากสมการ (2.7) , (2.9) และ (2.10) สามารถแสดงโดยบล็อกไดอะแกรมภาพที่ 2.2 การตอบสนองในโดเมนเวลาที่สัมพันธ์กับทรานเฟอร์ฟังก์ชันของสมการ (2.10) (เมื่อ V_f เปลี่ยนแปลงเป็นขั้น ๆ)

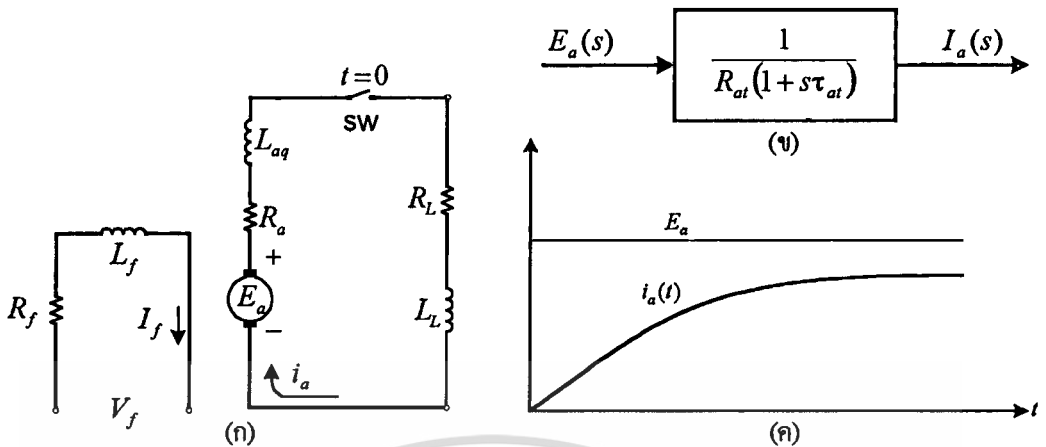
$$e_a(t) = \frac{K_g V_f}{R_f} (1 - e^{-t/\tau_f}) \quad (2.11)$$

$$= E_a (1 - e^{-t/\tau_f}) \quad (2.12)$$

ถ้า $E_a = e_a(\infty) = K_g V_f / R_f = K_g I_f$ คือแรงดันที่เกิดขึ้นในสภาวะคงตัวและ $I_f = \frac{V_f}{R_f}$ คือกระแส

สแตที่ป้อนขดลวดเข้าสนามในสภาวะคงตัว ผลการตอบสนองดังภาพที่ 2.2(ข) เป็นผลการตอบสนองของระบบอันดับหนึ่ง เทียบกับค่าเวลาคงตัว ซึ่งค่าเวลาคงตัวของวงจรมอเตอร์ τ_f มีค่ามากและเปลี่ยนแปลงในช่วง 0.1 - 2 วินาที

2.2.2 พิจารณาทรานเซียนในวงจรรีเมเจอร์



ภาพที่ 2.3 ทรานเซียน และสภาวะทางไดนามิกส์ของวงจรรีเมเจอร์เมื่อต่อโหลด

เมื่อพิจารณาทรานเซียนที่เกิดขึ้นในวงจรรีเมเจอร์ เมื่อต่อโหลดเข้ากับระบบดังภาพที่ 2.3 ซึ่งประกอบด้วยความต้านทาน R_L และค่าความเหนี่ยวนำ L_L ซึ่งต่ออยู่กับขั้วอาร์เมเจอร์ เมื่อสวิตช์ถูกปิดที่เวลา $t = 0$ ถ้าสมมติว่าอาร์เมเจอร์หมุนด้วยความเร็วคงที่และกระแสของขดลวดสนามคงที่ด้วย ดังนั้นหลังจากสวิตช์ปิดสมการแรงดัน E_a กำหนดได้ตามสมการ (2.13)

$$E_a = R_a i_a + L_{aq} \frac{di_a}{dt} + R_L i_a + L_L \frac{di_a}{dt} \quad (2.13)$$

และเมื่อจัดสมการที่ (2.13) ใหม่จะได้ดังนี้

$$E_a = (R_a + R_L) i_a + (L_{aq} + L_L) \frac{di_a}{dt} \quad (2.14)$$

เมื่อ $R_{at} = R_a + R_L$ คือ ผลรวมของค่าความต้านทานในวงจรรีเมเจอร์

$L_{at} = L_{aq} + L_L$ คือ ผลรวมของค่าความเหนี่ยวนำในวงจรรีเมเจอร์

$$E_a = R_{at} i_a + L_{at} \frac{di_a}{dt} \quad (2.15)$$

เมื่อแปลงลาปลาซ ของสมการ 2.15 จะได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$E_s(s) = R_{st}I_s(s) + L_{st}sI_s(s) \quad (2.16)$$

คังนั้ทรานเฟอ์ฟังกั้ฉั้ คือ

$$\frac{I_s(s)}{E_s(s)} = \frac{1}{R_{st}(1+s\tau_{st})} \quad (2.17)$$

โดยที่ $\tau_{st} = \frac{L_{st}}{R_{st}}$ คือ ค่ำเวลาคังตัวของวงจรรำเมเจอร์

ส่วนบล็อกไคอะแกรมแสดงทรานเฟอ์ฟังกั้ฉั้ของสมการ (2.17) แสดงคังภาพที่ 2.3(ข) และ ผลการคอบสนองในโดเมนเวลา ค่ำหนดตามสมการ (2.18)

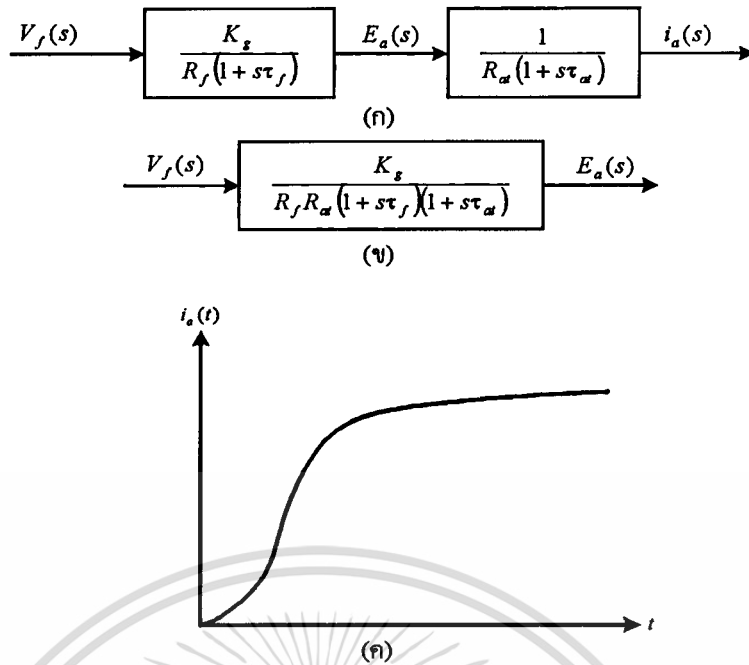
$$i_s(t) = \frac{E_s}{R_{st}} (1 - e^{-t/\tau_{st}}) \quad (2.18)$$

และผลการคอบสนองสมการ (2.18) แสดงคังภาพที่ 2.3(ค) ฉั้เป็นผลคอบสนองอ้นคั้ที่หนั้ เทียบค่ำเวลาคังตัว τ_{st} ตามปกติแล้วนั้ τ_{st} มีค่ำค่ำ คังนั้กระแสรำเมเจอร์ฉั้เพิ่มฉั้ได้อั้ยั้ รวดเร็ว

จากสมการ (2.10) และ (2.17) ผลรวมของทรานเฟอ์ฟังกั้ฉั้ที่สัมพันธ์กับกระแสรำเมเจอร์ต่อ แรงคั้ขคลวดสนามคือ

$$\frac{I_s(s)}{V_f(s)} = \frac{I_s(s)}{E_s(s)} \cdot \frac{E_s(s)}{V_f(s)} = \frac{K_g}{R_f R_{st} (1+s\tau_f)(1+s\tau_{st})} \quad (2.19)$$

ความสัมพันธ์ของสมการ (2.19) แสดงได้โดยบล็อกไคอะแกรมภาพที่ 2.4(ข)



ภาพที่ 2.4 ผลรวมความสัมพันธ์ของกระแสต่อแรงดันขดลวดสนาม

และสำหรับการเปลี่ยนแปลงของแรงดันแบบสเตป (Step change of voltage) ในวงจรสนามกำหนดตาม สมการ (2.20)

$$V_f(s) = \frac{V_f}{s} \quad (2.20)$$

และจากสมการ (2.19) และ (2.20) กระแสอาร์เมเจอร์หาได้ตามสมการ (2.21)

$$I_a(s) = \frac{K_g V_f}{R_f R_a s (1+s\tau_f)(1+s\tau_a)} \quad (2.21)$$

หรือ

$$I_a(s) = \frac{K_g V_f}{R_f R_a \tau_f \tau_a s (1+s\tau_f)(1+s\tau_a)} \quad (2.22)$$

กำหนดให้ $A = K_g V_f / R_f R_a \tau_f \tau_a$ ดังนั้นสมการ (2.22) เขียนใหม่ได้ดังนี้

$$I_a(s) = \frac{A}{s(s+1/\tau_f)(s+1/\tau_a)} \quad (2.23)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (2.23) แก่สมการหาค่า $i_s(t)$ ได้ดังนี้ คือ

$$I_s(s) = \frac{A_1}{s} + \frac{A_2}{s+1/\tau_f} + \frac{A_3}{s+1/\tau_{st}} \quad (2.24)$$

$$\text{เมื่อ } A_1 = \left. \frac{A}{(s+1/\tau_f)(s+1/\tau_{st})} \right|_{s=0} = A\tau_f\tau_{st}$$

$$A_2 = \left. \frac{A}{s(s+1/\tau_{st})} \right|_{s=-1/\tau_f}$$

$$A_3 = \left. \frac{A}{s(s+1/\tau_f)} \right|_{s=-1/\tau_{st}}$$

ดังนั้นผลการตอบสนองของ i_s ในโมเมนเวลา คือ

$$i_s(t) = A_1 + A_2 e^{-t/\tau_f} + A_3 e^{-t/\tau_{st}} \quad (2.25)$$

หมายเหตุ A_1 แสดงค่ากระแสอาร์เมเจอร์ในสภาวะคงตัว เมื่อ

$A_1 = i_s(\infty) = (K_g V_f) / (R_f R_{st}) = K_g I_f / R_{st} = E_g / R_{st}$ และภาพที่ 2.4 (ข) แสดงการตอบสนองของกระแส i_a สำหรับการเปลี่ยนแปลงแรงดันแบบสเตปในวงจรรีเลย์แบบ Step of change

2.3 แบบจำลองทางพลศาสตร์ และการควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น

มอเตอร์ไฟฟ้ากระแสตรงถูกนำมาประยุกต์ใช้งานกันอย่างกว้างขวาง เนื่องจากการควบคุมความเร็วและแรงบิดสามารถควบคุมได้ในช่วงกว้าง สำหรับวิธีการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนามแยกกระตุ้นเป็นการควบคุมการกระตุ้นขดลวดสนามให้คงที่ โดยที่ความเร็วสามารถควบคุมด้วยการปรับแรงดันที่ขั้ว ซึ่งการศึกษาผลที่เกิดจากสภาวะทรานเซียนในวงจรรีเลย์และทรานเซียนทางกลในระบบขับเคลื่อนทางกลโดยพิจารณาออเตอร์ ซึ่งในการออกแบบระบบควบคุมความเร็วรอบของมอเตอร์นั้น ถึงแม้ว่าจะไม่จำเป็นต้องศึกษาถึงการออกแบบและการสร้างตัวมอเตอร์โดยละเอียด แต่ก็ยังคงจำเป็นจะต้องศึกษาถึงคุณสมบัติที่สำคัญบางอย่างได้แก่ สมการความเร็วรอบแรงบิด T_d, T_L, e_s, P_I และ $\% \eta$ ของมอเตอร์ ซึ่งจากค่าตัวแปรเหล่านี้สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

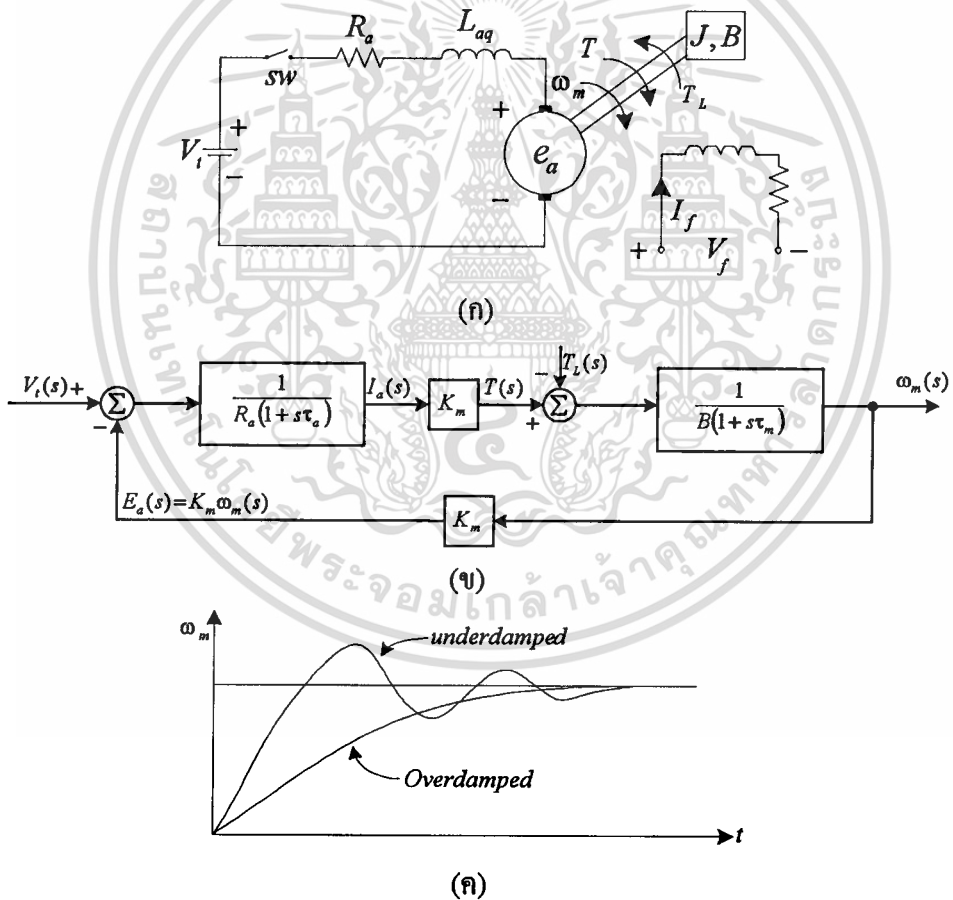
อธิบายถึงวิธีการปรับความเร็วรอบมอเตอร์

ระบบมอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนามแยกตัวกระตุ้นแสดงดังภาพที่ 2.5(ก) ถ้าสมมติว่าการเปลี่ยนแปลงของสนามแม่เหล็กเป็นแบบเชิงเส้น (Magnetic linearity) และสมการพื้นฐานของมอเตอร์กำหนดดังนี้ คือ

$$e_a = K_f i_f \omega_m = K_m \omega_m \quad (2.26)$$

$$T = K_f i_f i_a = K_m i_a \quad (2.27)$$

เมื่อ $K_m = K_f i_f$ คือ ค่าคงที่ ซึ่งเท่ากับอัตราส่วนของ e_a / ω_m แรงดันอาร์เมเจอร์ e_a จะสัมพันธ์กับกระแสในขดลวดสนาม i_f ที่ความเร็ว ω_m



ภาพที่ 2.5 ระบบมอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนามแยกตัวกระตุ้น

สำหรับผลการแปลงลาปลาซของสมการ (2.26) และ (2.27) กำหนดได้ดังนี้

$$E_a = K_m \omega_m (s) \quad (2.28)$$

$$T(s) = K_m I_a(s) \quad (2.29)$$

ในภาพที่ 2.5(ก) ถ้าสวิตช์ปิดที่เวลา $t = 0$ ดังนั้นหลังจากสวิตช์ปิดไปแล้วแรงดันที่ขั้ว V_t ตามกฎของ KVL กำหนดได้ดังนี้ คือ

$$V_t = e_a + R_a i_a + L_{aq} \frac{di_a}{dt} \quad (2.30)$$

และจากสมการ (2.26) และ (2.30)

$$V_t = K_m \omega_m + R_a i_a + L_{aq} \frac{di_a}{dt} \quad (2.31)$$

ดังนั้นผลการแปลงลาปลาซของสมการ (2.31) จากเงื่อนไขขณะเริ่มต้นเป็นศูนย์จะได้ $V_t(s)$ คือ

$$V_t(s) = K_m \omega_m(s) + R_a i_a(s) + L_{aq} s I_a(s) \quad (2.32)$$

จัดสมการ (2.32) ใหม่ได้ดังนี้ คือ

$$V_t(s) = K_m \omega_m(s) + I_a(s) R_a (1 + s\tau_a) \quad (2.33)$$

เมื่อ $\tau_a = L_{aq} / R_a$ คือ ค่าเวลาคงตัวของอาร์เมเจอร์

สำหรับสมการทางไดนามิกส์ของระบบทางกลกำหนดได้ดังนี้

$$T = K_m i_a = J \frac{d\omega_m}{dt} + B\omega_m + T_L \quad (2.34)$$

เมื่อ J คือ ผลรวมความเฉื่อยของเนื่องจากโพลและโรเตอร์

B คือ ค่าคงที่ของแรงเสียดทานเนื่องจากความฝืด ของโพลและมอเตอร์

T_L คือ แรงบิดทางกลของโพล

สำหรับในเทอม $B\omega_m$ แทนแรงบิดสูญเสียเนื่องจากการหมุนของระบบการแปลงลาปลาซ ของสมการ (2.34) นั่นคือ

$$T(s) = K_m I_a(s) = Js\omega_m(s) + B\omega_m(s) + T_L(s) \quad (2.35)$$

จากสมการ (2.35) และ (2.28) สมการความเร็ว (ω_m) กำหนดได้ดังนี้ คือ

$$\omega_m(s) = \frac{T(s) - T_L(s)}{B(1 + sJ/B)} = \frac{K_m I_a(s) - T_L(s)}{B(1 + s\tau_m)} \quad (2.36)$$

เมื่อ $\tau_m = J/B$ คือ ค่าเวลาคงตัวทางกลของระบบ และจากความสัมพันธ์ของสมการที่ (2.29) และ (2.33) กระแสอาร์เมเจอร์หาได้ตามสมการ (2.37)

$$I_a(s) = \frac{V_t(s) - E_a(s)}{R_a(1 + s\tau_a)} = \frac{V_t(s) - K_m \omega_m(s)}{R_a(1 + s\tau_a)} \quad (2.37)$$

สำหรับบล็อกไดอะแกรมของสมการ (2.36) และ (2.37) แสดงดังภาพที่ 2.5(ข) และถ้าพิจารณาแยกเป็นกรณีทำได้ดังนี้ คือ

1. แรงบิดของโหลดเป็นสัดส่วนกับความเร็ว

$$\begin{aligned} T_L &\propto \omega_m \\ &= B_L \omega_m \end{aligned} \quad (2.38)$$

ถ้ากำหนดให้ J คือผลรวมของความเฉื่อยทั้งระบบซึ่งเขียนได้ดังนี้ คือ

$$J = J_{\text{motor}} + J_{\text{load}} \quad (2.38.1)$$

จากสมการ (2.35) สามารถเขียนได้ดังนี้ คือ

$$K_m I_a(s) = Js\omega_m(s) + B_m \omega_m(s) + B_L \omega_m(s) \quad (2.39)$$

เมื่อกำหนดให้ $B = (B_m + B_L)$ ดังนี้

$$K_m I_a(s) = Js\omega_m(s) + B\omega_m(s) \quad (2.39.1)$$

เมื่อใส่โหลดให้กับระบบขึ้นจะทำให้แรงเสียดทานเนื่องจากความฝืดของระบบทางกลเพิ่มขึ้นและจากสมการ (2.33) และ (2.39.1) แรงดันที่ขั้วเขียนใหม่ได้ดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_t(s) = K_m \omega_m(s) + \frac{BR_a}{K_m} (1+s\tau_m)(1+s\tau_a) \omega_m(s) \quad (2.40)$$

ดังนั้นอัตราส่วนของความเร็วต่อแรงดันที่ขั้วหาได้ตามสมการ (2.40.1) คือ

$$\frac{\omega_m(s)}{V_t(s)} = \frac{1}{K_m + (BR_a/K_m)(1+s\tau_m)(1+s\tau_a)} \quad (2.40.1)$$

การตอบสนองของความเร็วกำหนดโดยการเปลี่ยนแปลงแรงดันที่ขั้ว V_t ซึ่งเป็นการตอบสนองของระบบอันดับสอง เนื่องจากมีค่าเวลาคงตัวสองค่าคือ τ_m และ τ_a การพิจารณาผลตอบสนองว่าเป็นแบบถ่วงน้อย หรือถ่วงมาก ขึ้นอยู่กับค่าเวลาคงตัวเหล่านั้นรวมทั้งค่าพารามิเตอร์อื่นๆ ด้วย เช่น K_m , B , และ R_a เป็นต้น ซึ่งการตอบสนองทั้งสองกรณีแสดงดังภาพที่ 2.5(ค)

2. ถ้า $L_{a'} = 0$ นั่นคือไม่คิดผลของค่าความเหนี่ยวนำและค่าเวลาคงตัว $\tau_a = 0$ ทรานเฟอร์ฟังก์ชันในสมการ (2.40.1) เขียนใหม่ได้ตามสมการ (2.41)

$$\frac{\omega_m(s)}{V_t(s)} = \frac{1}{K_m + (BR_a/K_m)(1+s\tau_m)} \quad (2.41)$$

หรือ

$$\frac{\omega_m(s)}{V_t(s)} = \left(\frac{K_m}{K_m^2 + R_a B} \right) \left(\frac{1}{1+s\tau'_m} \right) \quad (2.42)$$

เมื่อ

$$\tau'_m = \frac{R_a B}{K_m^2 + R_a B} \tau_m < \tau_m \quad (2.43)$$

3. ถ้า $B = 0$ สมการ (2.39.1) เขียนใหม่ได้ดังนี้ คือ

$$K_m I_a(s) = Js \omega_m(s) \quad (2.44)$$

และจากสมการ (2.33) และ (2.44) จะได้สมการแรงดันที่ขั้ว คือ

$$V_t(s) = K_m \omega_m(s) + \frac{Js \omega_m(s) R_a (1+s\tau_a)}{K_m}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

$$\frac{\omega_m(s)}{V_i(s)} = \frac{1}{K_m + (J R_a / K_m) s (1 + s \tau_a)} \quad (2.45)$$

4. กรณีแหล่งจ่ายถูกปลดออก ถ้าแหล่งจ่ายถูกปลดออกอย่างรวดเร็ว นั่นคือ สวิตซ์ดังภาพที่ 2.5(ก) เปิดออกที่เวลา $t = 0$ สมการทางไดนามิกส์ สำหรับระบบทางกล คือ

$$T = K_m i_a = J \frac{d\omega_m}{dt} + B \omega_m = 0 \quad (2.46)$$

ดังนั้น

$$B \omega_m = -J \frac{d\omega_m}{dt} \quad (2.47)$$

และผลการแปลงลาปลาซ ของสมการ (2.47) จะได้เป็น

$$B \omega_m(s) = -J [s \omega_m(s) - \omega_{m0}] \quad (2.48)$$

จากสมการ (2.48) เมื่อ ω_{m0} คือ ความเร็วเริ่มต้น

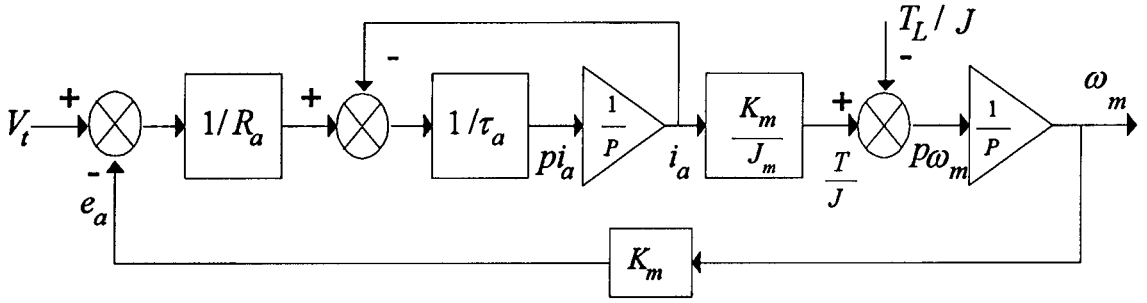
ดังนั้นความเร็ว ω_m เขียนได้ตามสมการ (2.49) คือ

$$\omega_m(s) = \frac{J \omega_{m0}}{B + sJ} = \frac{\omega_{m0}}{(s + B/J)} \quad (2.49)$$

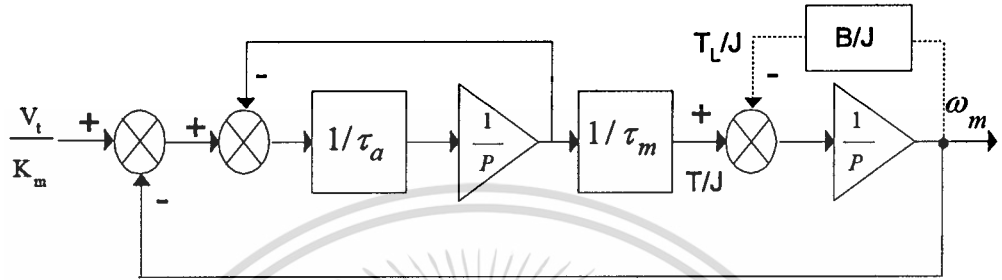
สำหรับผลตอบสนองของความเร็วในโดเมนเวลา คือ

$$\omega_m(t) = \omega_{m0} e^{-t/\tau_m} \quad (2.50)$$

และจากสมการ (2.26)-(2.34) รวมถึง 4 กรณีต่างๆ ดังกล่าวสามารถเขียนไดอะแกรมทางพลศาสตร์ของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น ได้ใหม่เทอมของพหุคูณดิฟ ดังแสดงความสัมพันธ์ของตัวแปรต่างๆ และสมการทางคณิตศาสตร์ได้ใหม่ ดังบล็อคดีอะแกรมภาพที่ 2.6



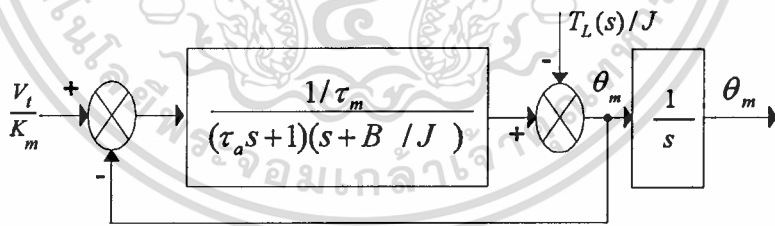
(ก)



(ข)

ภาพที่ 2.6 แบบจำลองทางพลศาสตร์ของมอเตอร์แบบแยกตัวกระตุ้น

จากภาพที่ 2.6 สามารถเขียนไดอะแกรมทรานเฟอร์ฟังก์ชันทางคณิตศาสตร์ในเทอมของการแปลงเชิงซ้อนของทางพลศาสตร์ในภาพที่ 2.6 ใหม่เป็นดังภาพที่ 2.7 และพิจารณาการควบคุมตำแหน่งของระบบมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นได้ดังภาพที่ 2.7



ภาพที่ 2.7 บล็อก ไดอะแกรมของระบบมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น

ดังนั้นสมการทรานเฟอร์ฟังก์ชันเอาต์พุตต่ออินพุตจะเป็น

$$\frac{\theta_m}{V_t/K_m} = \frac{1}{\tau_m(\tau_a s + 1)(s + B/J) + 1} \times \frac{1}{s} \tag{2.51}$$

หรือทำให้อยู่ในภาพของตัวระบวนการเป็นชนิดที่1 จะได้เป็นดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\theta_m}{V_t / K_m} = \frac{1}{\tau_m s (\tau_a s + 1) (s + B/J) + 1} \quad (2.52)$$

และจากสมการที่ (2.52) สามารถหาค่าสมการคุณลักษณะ จะได้เป็น

$$s \left[\left(s + \frac{1}{\tau_m} \right) \left(s + \frac{B}{J} \right) + \frac{1}{\tau_m \tau_a} \right] = 0 \quad (2.53)$$

และจะได้

$$s \left(s^2 \left(\frac{1}{\tau_a} + \frac{B}{J} \right) s + \frac{1}{\tau_a} \left(\frac{1}{\tau_m} + \frac{B}{J} \right) \right) = 0 \quad (2.54)$$

เทียบกับรูปแบบมาตรฐานสมการอันดับสองจะได้เป็น

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.55)$$

เมื่อแทนค่าที่เหมาะสมและใช้การแปลงลาปลาซ จะได้ทรานเฟอร์ฟังก์ชันระหว่างระยะแทนที่ของมอเตอร์กับแรงดันอินพุตจะได้เป็น

$$G(s) = \frac{\theta_m}{V_t / K_m} = \frac{\frac{1}{\tau_a} \left(\frac{1}{\tau_m} + \frac{B}{J} \right)}{s \left(s^2 \left(\frac{1}{\tau_a} + \frac{B}{J} \right) s + \frac{1}{\tau_a} \left(\frac{1}{\tau_m} + \frac{B}{J} \right) \right)} \quad (2.56)$$

ด้วยเหตุที่ดีซีมอเตอร์แบบควบคุมอาร์เมเจอร์ จะมีวงรอบป้อนกลับในตัวเองซึ่งเกิดจากแรงดันไฟฟ้าต้านกลับ แรงดันไฟฟ้าต้านกลับนี้จะเป็นสัดส่วนกับค่าลบของความเร็วของมอเตอร์ดังสมการ (2.51)-(2.52) ดังนั้นผลของแรงดันไฟฟ้าต้านกลับจึงมีความหมายเป็นค่าความเสียดทานทางไฟฟ้าซึ่งจะปรับปรุงเสถียรภาพของมอเตอร์ให้ดีขึ้น จะเห็นว่าค่าคงที่ของแรงดันไฟฟ้าต้านกลับ K_m เป็นเทอมที่ถูกแทนเพิ่มให้กับเทอมค่าความต้านทานและสัมประสิทธิ์ค่าความเสียดทาน โดยที่ค่าคงที่ของแรงดันไฟฟ้าต้านกลับ K_m จะมีความสัมพันธ์กับค่าคงที่ค่าหนึ่ง และกำลังงานเชิงกลที่เกิดขึ้นในวงจรรอาร์เมเจอร์ของมอเตอร์หาได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P = \frac{e_a(t)j_a(t)}{746} \quad \text{hp} \quad (2.57)$$

ดังนั้นสามารถแสดง ความถี่ธรรมชาติที่ไม่มีการหน่วงของ ω_n จาก

$$\omega_n = \sqrt{\frac{1}{\tau_a} \left(\frac{1}{\tau_m} + \frac{B}{J} \right)} \quad (2.58)$$

และแฟคเตอร์การหน่วงของ α จะได้เป็นดังนี้

$$\alpha = \frac{1}{2} \left(\frac{1}{\tau_a} + \frac{B}{J} \right) \quad (2.59)$$

และเมื่อพิจารณาความสัมพันธ์ของแฟคเตอร์การหน่วง α กับอัตราการหน่วงของ ζ จะเป็นดังนี้คือ

$$\zeta = \frac{\alpha}{\omega_n} \quad (2.60)$$

โดยค่ารากในครึ่งระนาบด้านซ้ายมือหาได้จากสมการ (2.61) คือ

$$s_1, s_2 = -\zeta\omega_n \pm \omega_n \sqrt{\zeta^2 - 1} = -\zeta\omega_n \pm \omega_n \sqrt{1 - \zeta^2} \quad (2.61)$$

จากสมการ (2.61) เมื่อกำหนดให้ $\zeta > 1$ ในเทอมของเอ็คโพเนนเชียล ด้วยจำนวนเต็มค่าลบ และ $\zeta < 1$ คือค่ารากอันดับที่สอง ที่มีการหน่วงเป็นแบบชายโนโซลคอลล์ โดยโหลดทางกลนั้นมีผลกระทบต่อทางเดินของรากล้น้อยมาก ดังนั้นค่า ω_n และ α จึงไม่มีผลกระทบต่อความเร็วในสภาวะคงตัว ดังนั้นสามารถเขียนสมการความถี่ธรรมชาติที่ไม่มีการหน่วงได้ใหม่เป็น

$$\omega_n = \sqrt{\frac{1}{\tau_a \tau_m}} \quad (2.62)$$

ซึ่งแฟคเตอร์การหน่วงของ α จะได้ว่า

$$\alpha = \frac{1}{2\tau_a} \quad (2.63)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

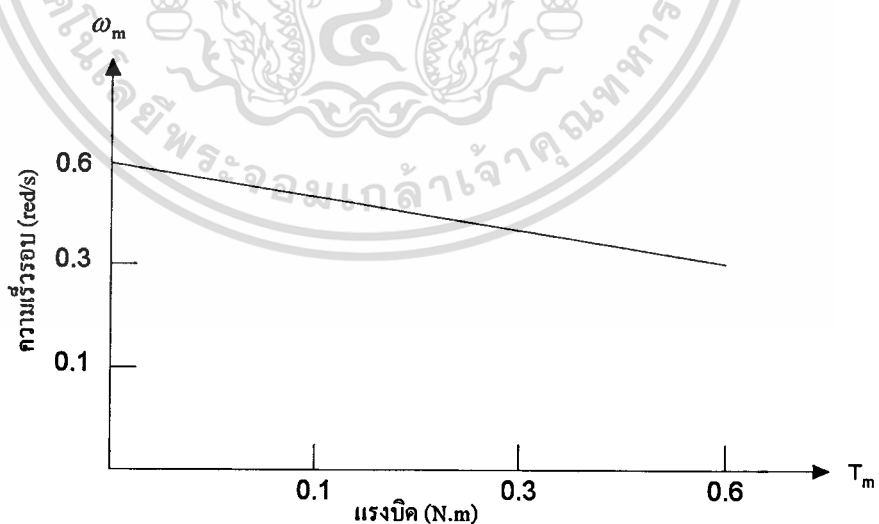
และอัตราส่วนการหน่วงของ ζ จะได้เป็น

$$\zeta = \frac{1}{2} \sqrt{\frac{r_m}{r}} \quad (2.64)$$

ดังนั้นการแก้มการอันดับที่สองของระบบนี้จึงเป็นแบบมีเงื่อนไขตามสภาวะทางเดินของราก และจากการทดลองกำหนดให้มอเตอร์ใช้กำลังงานสูงสุด และมีการชดเชยในตัวมอเตอร์แล้วด้วยแรงดันด้านกลับค่าลบ หรือเรียกว่าวงรอบของการควบคุมความเร็วภายในตัวมอเตอร์

2.4 ผลกระทบต่อความเร็วรอบของมอเตอร์

เมื่อกลับไปพิจารณาไดอะแกรมคังภาพที่ 2.1 สังเกตได้ว่าแรงดันไฟฟ้าที่ขั้วของอาร์เมเจอร์ V_a ซึ่งเป็นปริมาณคงที่ ถ้าแรงบิดเกิดการเปลี่ยนแปลงเนื่องจากการเพิ่มขึ้นหรือลดลงของโหลด (ทำให้กระแสในอาร์เมเจอร์ i_a เปลี่ยนแปลง) จะมีผลทำให้ความเร็วรอบ และสมการที่ (2.36) เปลี่ยนแปลงไปด้วย ซึ่งความสัมพันธ์ของความเร็วรอบกับแรงบิดของมอเตอร์ในกรณีนี้แสดงได้คังภาพที่ 2.8 และจากภาพที่ 2.8 สรุปได้ว่ามอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้น เมื่อแรงดันไฟฟ้าที่ขั้วของอาร์เมเจอร์มีปริมาณคงที่แล้ว เมื่อมีโหลดเพิ่มขึ้นก็จะทำให้ความเร็วรอบลดลง ดังนั้นจึงจำเป็นต้องมีระบบควบคุมเพื่อควบคุมความเร็วรอบ และแรงบิดของมอเตอร์ ไม่ให้มีการเปลี่ยนแปลงไปเนื่องจากโหลดเพิ่มขึ้นหรือลดลง



ภาพที่ 2.8 แสดงความสัมพันธ์ความเร็วรอบ และกำลังบิด เมื่อแรงดันไฟฟ้าที่ขั้วของวงจรอาร์เมเจอร์มีค่าคงที่

บทที่ 3

หลักการของระบบควบคุมป้อนกลับ

ระบบควบคุมแบบอัตโนมัติ (Automatic Controller) บางครั้งจะเรียกว่าการควบคุมป้อนกลับ (Feedback control) ซึ่งจะทำหน้าที่เป็นตัวควบคุมการทำงานของระบบพลศาสตร์ที่เราจะควบคุม โดยระบบควบคุมแบบอัตโนมัตินี้จะทำการเปรียบเทียบสัญญาณที่วัดค่าตัวแปรสถานะหรือ state variables จริงจากระบบด้วยตัวตรวจรู้ (sensor) แล้วนำมาเปรียบ เทียบกับค่าที่ต้องการพิจารณาการเปลี่ยนแปลงที่เกิดขึ้นและสร้างสัญญาณควบคุมซึ่งจะสามารถลดการเปลี่ยนแปลงนี้ลงไปได้ให้มีค่าเป็นศูนย์หรือเป็นค่าตัวเลขที่มีขนาดน้อยๆ การกระทำของระบบควบคุมอัตโนมัติที่สร้างสัญญาณควบคุม (control signal) นี้เราเรียกว่ากิริยาควบคุม (Control Action)

ในบทนี้เราจะมาพิจารณาถึงการใช้งานของกิริยาควบคุม (Control Action) ชนิดต่างๆ ที่ใช้ใน ระบบควบคุมในโรงงานอุตสาหกรรม นอกจากนั้นก็จะศึกษาผลกระทบของกิริยาแบบสัดส่วน (Proportional action, P) กิริยาแบบอนุพันธ์ (derivative action, D) และกิริยาแบบอินทิกรัล (integral action, I) กับสมรรถนะของการทำงานของระบบที่เราจะทำการควบคุม การศึกษาถึงประโยชน์ในการใช้ระบบควบคุมแบบป้อนกลับเพื่อมาลดผลกระทบที่เกิดขึ้นเนื่องจากการเปลี่ยนแปลงของค่า พารามิเตอร์ต่างๆ ในระบบ ซึ่งจะได้กล่าวต่อไป

3.1 ความจำเป็นที่ต้องการระบบควบคุมแบบป้อนกลับ

ระบบควบคุมความเร็วของมอเตอร์กระแสตรงสลับดังได้กล่าวมาแล้วในบทที่ 2. ถ้าสมมุติให้ ค่าเหนี่ยวนำของขดลวดในตัวมอเตอร์นั้นมีค่าน้อย เมื่อเทียบกับค่าความต้านทานของขดลวด สม การการเคลื่อนที่สำหรับใช้ในการควบคุมควบคุมความเร็วของมอเตอร์กระแสตรงแบบการควบคุม อาร์เมเจอร์ (Armature control) ซึ่งได้กล่าวมาแล้วในบทต้นนั้น และเขียนสมการใหม่ได้ดังนี้

$$v_a(t) - K_m \omega_m = i_a R_a \quad (3.1)$$

$$i_a = \frac{T}{k_i} R_a \quad (3.2)$$

$$T = L \frac{d\omega_m}{dt} + c\omega_m \quad (3.3)$$

เมื่อ k_i เป็นเกณฑ์ของอินทิกรัลในรูปกระแส และ L เป็นกระแสที่ไหลในวงจรรออาร์เมเจอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำมาใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะได้ว่า

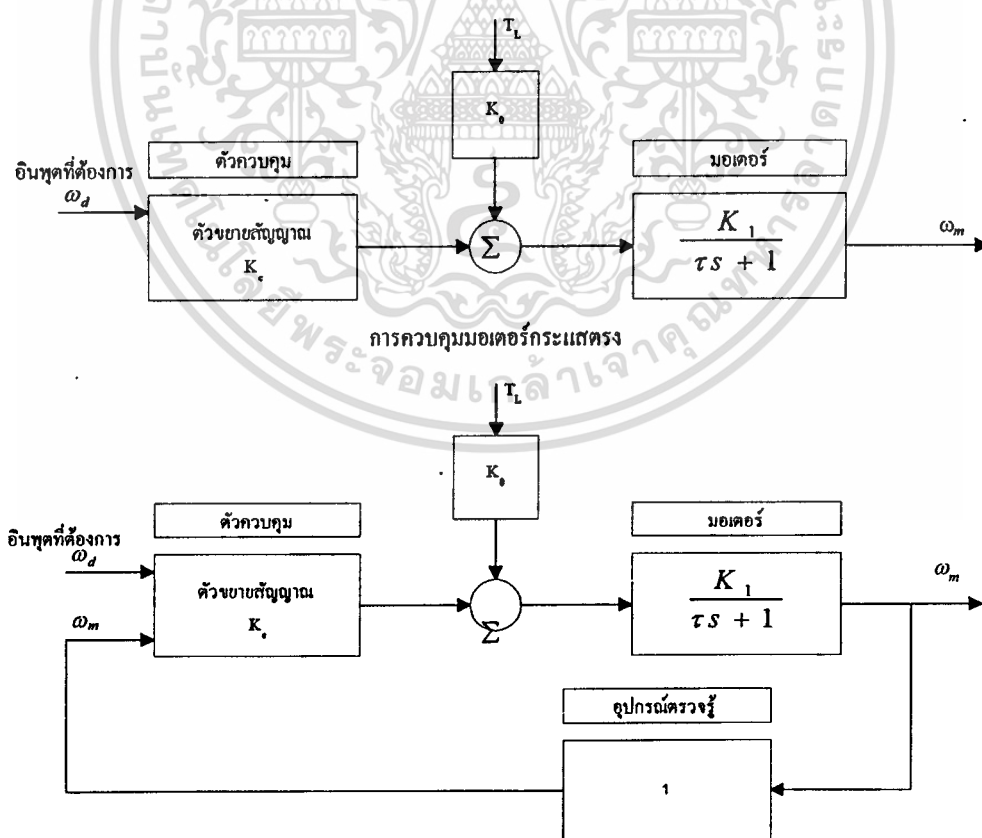
$$I \frac{d\omega_m(t)}{dt} + \left(\frac{k_i k_m}{R_s} + c \right) \omega_m(t) = \frac{k_i}{R_s} V_i(t) \quad (3.4)$$

$$\text{และถ้ากำหนดให้ } \tau = \frac{IR_s}{k_i k_m + cR_s} \text{ และ } K_1 = \frac{k_i}{k_i k_m + cR_s} \quad (3.5)$$

$$\text{ก็จะได้ว่า } \tau \frac{d\omega_m}{dt} + \omega_m = K_1 V_i \quad (3.6)$$

และในรูปของสมการฟังก์ชันถ่ายโอน (Transfer function) ก็จะได้ว่า

$$\frac{\omega_m(s)}{V_i(s)} = \frac{K_1}{\tau s + 1} \quad (3.7)$$



ภาพที่ 3.1 การควบคุมการทำงานของมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าโหลดของมอเตอร์นั้นเปลี่ยนแปลงเราอาจจะถือได้นั้นเป็นสัญญาณรบกวน ได้ และเมื่อรวมเอาระบบควบคุมแบบป้อนกลับเข้าด้วยกันแล้ว แผนภาพบล็อกของทั้งระบบมอเตอร์และระบบควบคุมสามารถแสดงได้ดังภาพที่ 3.1 โดยที่ T_L เป็นแรงบิดของโหลด $H(s)$ เป็นสมการฟังก์ชันถ่ายโอนของเครื่องมือตรวจรู้ (sensor) ในที่นี้ให้มีค่าเท่ากับ 1

3.2 เปรียบเทียบการควบคุมแบบเปิด และการควบคุมแบบป้อนกลับ

สำหรับการควบคุมแบบเปิด (open loop) นั้นระบบควบคุม โดยที่ค่าอัตราขยาย K_c คือค่ากำลังขยายที่ขยายสัญญาณ (amplifier) เพื่อใช้ขับมอเตอร์ ดังนั้นค่าแรงดันไฟฟ้า (voltage) ที่ใช้ในการขับเคลื่อนของมอเตอร์คือ

$$e_a = K_c \omega_d \quad (3.8)$$

ถ้ากำหนดให้ค่า K_c มีค่าเท่ากับ $1/K_1$ จะเห็นว่าในกรณีที่ไม่มีแรงบิดโหลด (load torque, $T_L = 0$) และคำสั่งความเร็ว ω_d เป็นฟังก์ชันขั้นหนึ่งหน่วย ค่าความเร็วเชิงมุมในสภาวะคงตัวของมอเตอร์จะมีค่าดังนี้

$$\begin{aligned} \omega_{ss} &= \lim_{s \rightarrow 0} s \frac{K_c}{Js + B} \times \frac{1}{s} \\ &= 1 = \omega_d \end{aligned} \quad (3.9)$$

ซึ่งแสดงว่าเมื่อปล่อยให้มอเตอร์หมุนไปจนถึงสภาวะอยู่ด้วยความเร็วของมอเตอร์จะเท่ากับความเร็วที่สั่งให้ตัวขยายสัญญาณสำหรับระบบควบคุมแบบป้อนกลับค่าแรงดันไฟฟ้าที่ตัวขยายสัญญาณสามารถเขียนได้ดังนี้

$$e_a = K(\omega_d - \omega_m) \quad (3.10)$$

โดยที่ค่าอัตราขยายเกน $K = \text{controller gain}$ ที่ตั้งปรับที่ตัวขยายสัญญาณเพื่อให้ได้แรงดันไฟฟ้าที่ใช้ควบคุมมอเตอร์ ดังนั้นความสัมพันธ์ระหว่างสัญญาณเอาต์พุต (output, ω_m) และสัญญาณคำสั่งหรือสัญญาณอินพุต (input, ω_d) ของระบบสามารถเขียนได้ดังนี้

$$\frac{\omega_m}{\omega_d} = \frac{KK_1}{\tau s + 1 + KK_1} \quad (3.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ ω_d เป็นฟังก์ชันขั้นหนึ่งหน่วย สามารถหาความเร็วเชิงมุมในสภาวะอยู่ตัว ได้ดังนี้คือ

$$\omega_{ss} = \lim_{s \rightarrow 0} s \frac{KK_1}{\tau s + 1 + KK_1} \times \frac{1}{s} \quad (3.12)$$

$$= \frac{KK_1}{1 + KK_1} \quad (3.13)$$

ถ้าเรากำหนดให้ค่าอัตราขยายเกนซ์ K มีค่ามากจะเห็นว่าค่า $\frac{KK_1}{1 + KK_1}$ จะมีค่าเข้าใกล้ 1 หรือ นั่นคือ ω_{ss} มีค่าเข้าใกล้ค่าความเร็วเชิงมุมที่ตั้งหรือ ω_d จะเห็นว่า ในกรณีของการควบคุมแบบเปิด (open loop) นั้นความเร็วเชิงมุมที่ในสภาวะอยู่ตัว (steady state) $\omega_{ss} = \omega_d$ แต่ในกรณีของการควบคุมแบบป้อนกลับ (closed - loop) นั้นค่าความเร็วในการหมุนของมอเตอร์ ω_m เพียงแต่มีค่าใกล้ ω_d เฉพาะในกรณีที่ค่าอัตราขยาย K ตั้งปรับไว้ที่ตัวควบคุมนั้นมีค่าสูงมากเท่านั้น และถ้าเราพิจารณาในกรณีที่ถ้าค่า K_1 สมการของมอเตอร์นั้นเราไม่สามารถหาค่าได้อย่างถูกต้องแม่นยำ กล่าวคือค่าที่ได้ อาจจะเป็น $K_1 + \delta K_1$

ในกรณีของการควบคุมแบบเปิด จะได้ว่า

$$\frac{\omega_m(s)}{\omega_d(s)} = \frac{\frac{1}{K}(K_1 + \delta K_1)}{\tau s + 1} \quad (3.14)$$

ความเร็วเชิงมุมในสภาวะอยู่ตัว (steady state) ในกรณีที่ $\omega_d = \text{unit step} = \frac{1}{s}$ จะได้ว่า

$$\omega_m = \lim_{s \rightarrow 0} \frac{\frac{1}{K}(K_1 + \delta K_1)}{\tau s + 1} \times \frac{1}{s} \quad (3.15)$$

$$= \frac{(K_1 + \delta K_1)}{K_1}$$

$$= 1 + \frac{\delta K_1}{K_1}$$

ดังนั้น

$$\delta \omega_m = \frac{\delta K_1}{K_1} \omega_d \quad (3.16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าคิดเป็นเปอร์เซ็นต์จะเขียนได้ว่า

$$\frac{\delta\omega_m}{\omega_m} = \frac{\delta K_1}{K_1} \times 100\% \quad (3.17)$$

ดังนั้นถ้า δK_1 มีค่า 10% ค่า $\delta\omega_m$ ซึ่งเป็นค่าผิดพลาดของความเร็วก็จะมีค่า 10 % เช่นเดียวกัน สำหรับในกรณีของการควบคุมแบบป้อนกลับ (feedback control) จะได้ว่า

$$\frac{\omega_m}{\omega_d} = \frac{K(K_1 + \delta K_1)}{(s+1) + K(K_1 + \delta K_1)} \quad (3.18)$$

และถ้ากำหนดให้ ω_d เป็นฟังก์ชันขั้นหนึ่งหน่วย นั่นคือ $\omega_d = \frac{1}{s}$ ดังนั้นค่าความเร็วเชิงมุมในสภาวะอยู่ตัวของความเร็วเชิงมุม ω_m สามารถเขียนได้ดังนี้

$$\begin{aligned} \omega_{ss} &= \lim_{s \rightarrow 0} s \frac{K(K_1 + \delta K_1)}{(s+1) + K(K_1 + \delta K_1)} \times \frac{1}{s} \\ &= \frac{K(K_1 + \delta K_1)}{1 + K(K_1 + \delta K_1)} \\ &\cong \frac{K(K_1 + \delta K_1)}{K(K_1 + \delta K_1)} \quad (K \gg 1) \\ &\cong 1 \end{aligned} \quad (3.19)$$

จะเห็นว่าถ้า δK_1 มีค่า 10% และ K หรือค่าอัตราขยายของตัวควบคุม (controller gain) มีค่าสูง ω_{ss} จะมีค่าเข้าใกล้ ω_d ดังนั้นจึงเห็นว่าเมื่อค่าพารามิเตอร์ของระบบมีค่าไม่แน่นอน ระบบควบคุมแบบป้อนกลับ(feedback)นั้นจะมีคุณสมบัติในการปรับปรุงดีกว่าระบบควบคุมที่เป็นแบบเปิด

3.3 ความไว

การวิเคราะห์หาค่าความไว (Sensitivity) ของระบบควบคุมมีประโยชน์ในการวิเคราะห์ว่าระบบที่เราจะควบคุมนั้นมีความไวต่อการเปลี่ยนแปลงของค่าพารามิเตอร์ต่าง ๆ ในระบบอย่างไรระบบที่มีความไวต่อการเปลี่ยนแปลงของค่าพารามิเตอร์ของระบบมาก เมื่อเวลาควบคุมต้องมีความระมัดระวังการเปลี่ยนแปลงค่าพารามิเตอร์นั้นด้วย โดยปกติแล้วการออกแบบระบบควบคุมต้องเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้ระบบควบคุมของเรานั้นมีความไวน้อยที่สุดเมื่อค่าพารามิเตอร์ของระบบเปลี่ยนแปลงไป การคำนวณหาความไว ของระบบนั้นสามารถหาได้ดังนี้คือ

$$\frac{\omega_m}{\omega_s} = \frac{K(K_1 + \delta K_1)}{1 + K(K_1 + \delta K_1)} \quad (3.20)$$

$$= \frac{\frac{KK_1}{1+KK_1} + \frac{K\delta K_1}{1+KK_1}}{1 + \frac{K\delta K_1}{1+KK_1}}$$

เนื่องจาก $\frac{1}{1 + \frac{K\delta K_1}{1+KK_1}} = 1 - \frac{K\delta K_1}{1+KK_1} ; \left(\frac{1}{(1+x)} = 1-x \right)$

เพราะฉะนั้น

$$\omega_m = \left\{ \frac{KK_1}{1+KK_1} + \frac{K\delta K_1}{1+KK_1} \right\} \left\{ \frac{1+K\delta K_1}{1+KK_1} \right\} \omega_s \quad (3.21)$$

$$\frac{\omega_m}{\omega_s} = \left\{ \frac{KK_1}{1+KK_1} + \frac{K\delta K_1}{1+KK_1} \frac{(KK_1) + K\delta K_1}{(1+KK_1)^2} \right\} \quad (3.22)$$

$$= \left\{ \frac{KK_1}{1+KK_1} + \frac{KK_1\delta K_1}{(1+KK_1)^2 K_1} \right\} \quad (3.23)$$

ดังนั้น

$$\delta \omega_m = \frac{KK_1\delta K_1}{(1+KK_1)^2 K_1} \quad (3.24)$$

ค่า $\omega_m = \frac{KK_1\omega_s}{1+KK_1}$ คือค่าความเร็วที่ไม่มีกรรบกรวน (unperturbed speed) ดังนั้นจะได้ว่า

$$\delta \omega_m = \frac{1}{1+KK_1} \frac{\delta K_1}{K_1} \quad (3.25)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่าค่า δK_1 จะเปลี่ยนแปลง 10 % แต่ $\delta \omega_m$ มีค่าเข้าใกล้ศูนย์ในกรณีที่ค่า KK_1 มีค่ามาก ค่า $\frac{\delta \omega_m / \omega_m}{\delta K_1 / K_1}$ นี้เรียกว่าค่าความไวของระบบ

จากการวิเคราะห์และพิจารณาผลที่ผ่านมา นั้น เราสมมุติว่า T_L หรือแรงบิด โหลด มีค่าเป็นศูนย์ ถ้า T_L มีค่าไม่เป็นศูนย์ ในกรณีของการควบคุมแบบเปิด เราจะได้ว่า

$$\omega_m = K_1(K_0 \omega_d + K_0 T_L) \frac{1}{\tau s + 1} \quad (3.26ก)$$

สำหรับค่าในสภาวะอยู่ตัว และ $\omega_d(s) = \omega_d / s$, $T_L(s) = T_L / s$ จะได้ว่า

$$\omega_m = K_1(K_0 \omega_d + K_0 T_L) \quad (3.26ข)$$

และถ้า $K_c = \frac{1}{K_1}$ แล้วจะได้ว่า

$$\omega_m = \omega_d + K_1 K_0 T_L \quad (3.27ก)$$

หรือ

$$\delta \omega_m = K_1 K_0 T_L \quad (3.27ข)$$

เพราะฉะนั้นค่าความเร็วที่ผิดพลาดจะเป็นสัดส่วนโดยตรงกับค่าโหลดที่มารบกวน (disturbing load) ในกรณีของการควบคุมแบบป้อนกลับจะได้

$$\omega_m = K_1 \left(\frac{K \omega_d}{(\tau s + B) + K_1 K \omega_d} + \frac{K_0 T_L}{(J s + B) + K_1 K T_L} \right) \quad (3.28ก)$$

สำหรับในสภาวะอยู่ตัว (steady-state) และเมื่อ $\omega_d(s) = \omega_d / s$, $T_L(s) = T_L / s$ จะได้ว่า

$$\omega_m = \left\{ \frac{KK_1}{1 + KK_1} \right\} \omega_d + \left\{ \frac{K_1 K_0}{1 + KK_1} \right\} T_L \quad (3.28ข)$$

ถ้าสามารถออกแบบให้ $K_1 K \gg 1$ และ $K_1 K \gg K_1 K_0$ แล้ว จะเห็นว่าค่าความผิดพลาดของความเร็วเนื่องจากมีโหลดเปลี่ยนแปลง (load change) จะมีค่าไม่มากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ชนิดของระบบควบคุมพื้นฐาน

อุปกรณ์ควบคุมจะถูกออกแบบไว้เพื่อให้สร้างสัญญาณควบคุมไปควบคุมระบบเมื่อมีสัญญาณผิดพลาด (error signal) เกิดขึ้น หลักการหรือวิธีการควบคุมนั้นเราเรียกว่ากฎของการควบคุม หรือ กฎการควบคุม (control law) หรือกิริยาควบคุม (control action) เมื่อสัญญาณผิดพลาด มีค่าไม่เป็น ศูนย์อาจจะเป็นผลเนื่องมาจากเกิดการเปลี่ยนแปลงของระบบ (system variable) ให้ใกล้เคียงกับ ค่าที่เราต้องการ (desired value) ซึ่งอาจจะสรุปหน้าที่ของตัวควบคุมพอสังเขปดังนี้

- ลดค่าผิดพลาดในสภาวะอยู่ตัว (minimize the steady state error)

- ลดค่าเวลาเข้าที่ (minimize the settling time)

- เพื่อให้การตอบสนองในในสภาวะชั่วคราวมีคุณลักษณะตามที่กำหนดไว้ เช่น ต้องการทำให้ค่า โอเวอร์ชูตมีค่าน้อยที่สุด

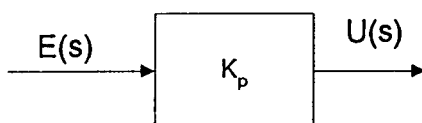
ในทางปฏิบัติแล้วการกำหนดลักษณะเฉพาะสำหรับออกแบบระบบควบคุมมีรายละเอียดมากกว่า นี้เช่น เราอาจจะต้องกำหนดความถี่แถบ (bandwidth) ที่ทำให้ระบบมีความปลอดภัยจากการที่ ระบบจะไม่มีเสถียรภาพหรือไม่มีความสมดุล เราไม่สามารถรู้ได้แน่ชัดเลยเกี่ยวกับค่าพารามิเตอร์ ของระบบเรานั้นว่ามีค่าแน่นอนเท่าไร ระบบควบคุมบางระบบสามารถออกแบบให้มีความไว มาก น้อยกว่าระบบควบคุมระบบอื่น ดังนั้นการออกแบบบางครั้งอาจจะต้องครอบคลุมถึงความไว ของ ระบบควบคุมต่อการเปลี่ยนแปลงของพารามิเตอร์ของระบบด้วย

ในบทนี้จะกล่าวถึงระบบควบคุมที่ใช้มากในอุตสาหกรรมซึ่งพอจะแบ่งได้ดังนี้คือ

- 1) ตัวควบคุมเชิงสัดส่วน (Proportional controllers, P-control)
- 2) ตัวควบคุมแบบอินทิกรัล (Integral controllers, I-control)
- 3) ตัวควบคุมเชิงสัดส่วนบวกอินทิกรัล (Proportion-plus-integral controllers, PI-control)
- 4) ตัวควบคุมเชิงสัดส่วนบวกอนุพันธ์ (Proportional-plus-derivative controllers, PD-control)
- 5) ตัวควบคุมเชิงสัดส่วนบวกอนุพันธ์บวกอินทิกรัล (Proportional-plus- integral-plus- derivative controllers, PID-control)

3.4.1 (ตัวควบคุมเชิงสัดส่วน)

การควบคุมระบบแบบป้อนกลับโดยใช้ตัวควบคุมเชิงสัดส่วนนั้น สัญญาณควบคุม (u) จะเป็นสัดส่วนโดยตรงกับค่าสัญญาณความผิดพลาด (e) ที่เกิดจากผลต่างระหว่างค่าสัญญาณอ้างอิง กับสัญญาณเอาต์พุตของระบบที่ทำการควบคุมนั้นดังแสดงในแผนภาพบล็อกข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่างเอาต์พุตจากตัวควบคุมและสัญญาณผิดพลาดที่ส่งเข้าไปในระบบสามารถเขียนได้ดังนี้

$$u(t) = K_p e(t) \quad (3.29)$$

ถ้าเขียนอยู่ในรูปแบบผลการแปลงลาปลาซ จะได้ว่า

$$\frac{U(s)}{E(s)} = K_p \quad (3.30)$$

เมื่อค่า K_p จะเป็นค่าอัตราขยายของตัวควบคุมนี้หรือเรียกว่าเกนสัดส่วน

การใช้ตัวควบคุมเชิงสัดส่วน (proportional control) กับการควบคุมตำแหน่งของการเปิดปิดวาล์วอาจจะทำได้ดังนี้

$$y(t) = K_p e(t) + x \quad (3.31)$$

โดยที่ $y(t)$ คือตำแหน่งที่ต้องการเปิดปิดวาล์ว และ x เป็นตำแหน่งที่เราเปิดปิดวาล์วให้อยู่ในตำแหน่งที่ใกล้เคียงกับตำแหน่งที่ต้องการหรือพยายามทำให้ค่าสัญญาณผิดพลาดเป็นศูนย์ ซึ่งตรงตำแหน่งนี้จะเรียกว่าการปรับตั้งใหม่ด้วยมือ (manual reset) เพราะฉะนั้นถ้าตำแหน่งเปลี่ยนไปเป็นระยะ $e(t)$ แล้ว ตัวควบคุมก็จะทำหน้าที่ปรับเพื่อให้ $y(t)$ มีค่าเท่ากับ $x(t)$ การควบคุมแบบนี้เปอร์เซ็นต์การเปลี่ยนแปลงของสัญญาณผิดพลาดที่ใช้ในการปรับตำแหน่งของวาล์วให้เต็มสเกลจะเรียกว่าแถบเชิงสัดส่วน (proportional band) ซึ่งสามารถเขียนเป็นความสัมพันธ์กับเกนได้ดังนี้

$$K_p = \frac{100}{band\%} \quad (3.32)$$

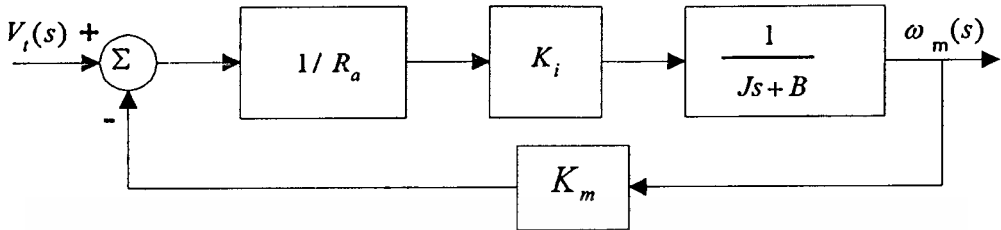
3.4.1.1 (การควบคุมโดยใช้ตัวควบคุมเชิงสัดส่วนกับระบบอันดับที่หนึ่ง)

ในหัวข้อนี้จะมาศึกษาถึงพฤติกรรมของการควบคุมโดยใช้ตัวควบคุมเชิงสัดส่วนกับระบบที่มีอันดับของสมการเท่ากับหนึ่งจากที่ได้ทำการศึกษาระบบพลศาสตร์ของการควบคุมการทำงานของมอเตอร์กระแสตรง เราจะนำมาใช้ในที่นี้ ซึ่งจะเป็นการควบคุมความเร็วในการหมุนของมอเตอร์ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\omega_m(s)}{V_i(s) - K_m \omega_m(s)} = \frac{K_i / R_a}{Js + B} \quad (3.33)$$

ซึ่งสามารถเขียนเป็นแผนภาพบล็อกได้ดังนี้



ภาพที่ 3.2 โค้ดแกรมของการควบคุมความเร็วของมอเตอร์กระแสตรง

จากแผนภาพบล็อกนี้เราสามารถหาสมการฟังก์ชันถ่ายโอนของระบบมอเตอร์ได้ดังนี้

$$\frac{\omega_m(s)}{V_i(s)} = \frac{K_i / R_a}{Js + B + \frac{K_i K_m}{R_a}} \quad (3.34)$$

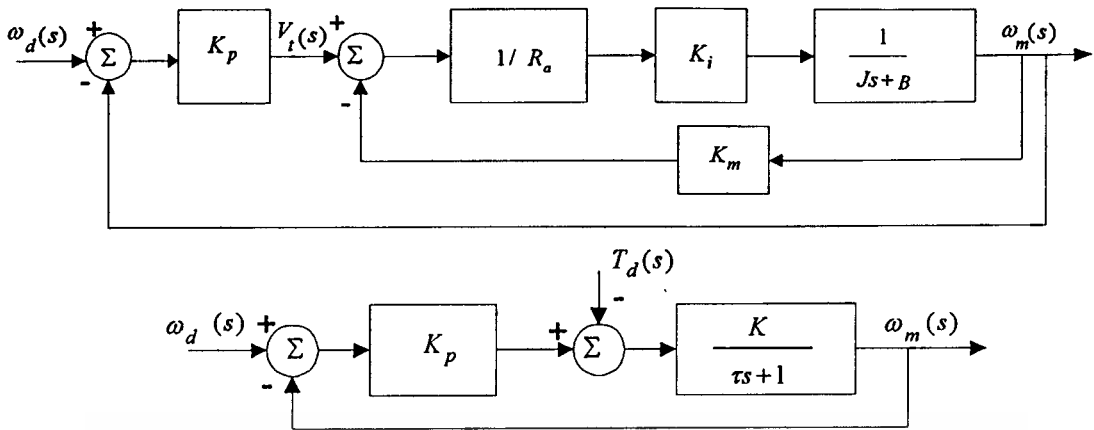
หรือ

$$\frac{\omega_m(s)}{V_i(s)} = \frac{K}{\tau s + 1} \quad (3.35)$$

โดยที่

$$K = \frac{K_i}{BR_a + K_i K_m} \quad \text{หรือ} \quad \tau = \frac{JR_a}{BR_a + K_i K_m} \quad (3.37)$$

ดังนั้นถ้าเราใช้ตัวควบคุมเชิงสัดส่วนมาควบคุมความเร็วของการหมุนของมอเตอร์เราสามารถเขียนเป็นแผนภาพบล็อกได้ดังแสดงในภาพที่ 3.3 ดังนี้



ภาพที่ 3.3 การควบคุมความเร็วในการหมุนของมอเตอร์โดยใช้ตัวควบคุมเชิงสัดส่วน

สมการฟังก์ชันถ่ายโอนสำหรับการควบคุมความเร็วการหมุนของมอเตอร์โดยใช้ตัวควบคุมเชิงสัดส่วนสามารถเขียนได้ดังนี้คือ

$$\frac{\omega_m(s)}{\omega_d(s)} = \frac{KK_p}{\tau s + 1 + KK_p} \quad (3.38)$$

$$\frac{\omega_m(s)}{T_d(s)} = \frac{K}{\tau s + 1 + KK_p} \quad (3.39)$$

เราสามารถวิเคราะห์ผลที่เกิดจากการเปลี่ยนแปลงความเร็วที่ต้องการโดยใช้สัญญาณเป็นฟังก์ชันหรือ $\omega_d(s) = \frac{1}{s}$ ดังนั้นเราจะได้ว่า

$$\omega_o(s) = \frac{KK_p}{\tau s + 1 + KK_p} \times \frac{1}{s} \quad (3.40)$$

ความเร็วในสภาวะอยู่ตัวสามารถหาได้ดังนี้คือ

$$\omega_{ss} = \lim_{s \rightarrow 0} s \frac{KK_p}{\tau s + 1 + KK_p} \times \frac{1}{s} = \frac{KK_p}{1 + KK_p} < 1 \quad (3.41)$$

ดังนั้นค่าความเร็วในสภาวะอยู่ตัวจะน้อยกว่าหนึ่ง ถ้าค่าอัตราขยายของตัวควบคุมเชิงสัดส่วน K_p มีค่ามาก ความเร็วในการหมุนของมอเตอร์ก็จะมีค่าเข้าใกล้ 1 มากขึ้น สำหรับระบบควบคุมแบบป้อนกลับในตัวอย่างนี้ค่าเวลาคงตัวจะมีค่าเท่ากับ
ไม่ทำการแก้ไขใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\tau_{cl} = \frac{\tau}{1+KK_p} \quad (3.42)$$

ถ้ามีการเปลี่ยนแปลงโพลตรงกันหันซึ่งเราอาจจะจำลองการเปลี่ยนแปลงนี้ด้วยอินพุตหนึ่งหน่วย (unit-step input) ดังนั้นสามารถวิเคราะห์ค่าความเร็วในสภาวะอยู่ตัวอันเนื่องมาจากการเปลี่ยนแปลงโพลแบบตรงกันหันนั้นได้ดังสมการนี้คือ

$$\omega_m(s) = -\frac{K}{zs+1+KK_p} \times \frac{1}{s} \quad (3.43)$$

และ

$$\omega_{ss} = \lim_{s \rightarrow 0} -s \frac{K}{zs+1+KK_p} \times \frac{1}{s} = -\frac{K}{1+KK_p} \quad (3.44)$$

จะเห็นว่าเราสามารถลดผลกระทบต่อความเร็วในสภาวะอยู่ตัวอันเนื่องมาจากการเปลี่ยนแปลงโพลได้โดยการใช้ค่าอัตราขยายของตัวควบคุมเชิงสัดส่วนที่มีค่าสูง

ดังนั้นจะเห็นว่า การใช้ตัวควบคุมเชิงสัดส่วนนี้จะมีข้อเสียก็ตรงที่ว่าค่าอัตราขยายของตัวควบคุมนั้นจะต้องมีค่าสูง การควบคุมแบบนี้จึงเหมาะสำหรับระบบที่ยอมให้เราตั้งค่าอัตราขยายได้สูง ข้อดีของการควบคุมเชิงสัดส่วนนั้นก็คือค่าสัญญาณควบคุมจะตอบสนองกับค่าสัญญาณผิดพลาดทันทีทันใดเลขซึ่งจะมีประโยชน์ในกรณีที่เราต้องการการตอบสนองอย่างรวดเร็วเป็นต้น ในกรณีที่ค่าคงตัวเวลาของระบบมีขนาดเล็กเกินไปสำหรับการควบคุมแบบเปิด/ปิด (on/off) นั้น ถ้าใช้การควบคุมเชิงสัดส่วนนี้ก็จะสามารถบรรลุเป้าหมายได้

3.4.1.2 (การชดเชยแบบป้อนไปข้างหน้า)

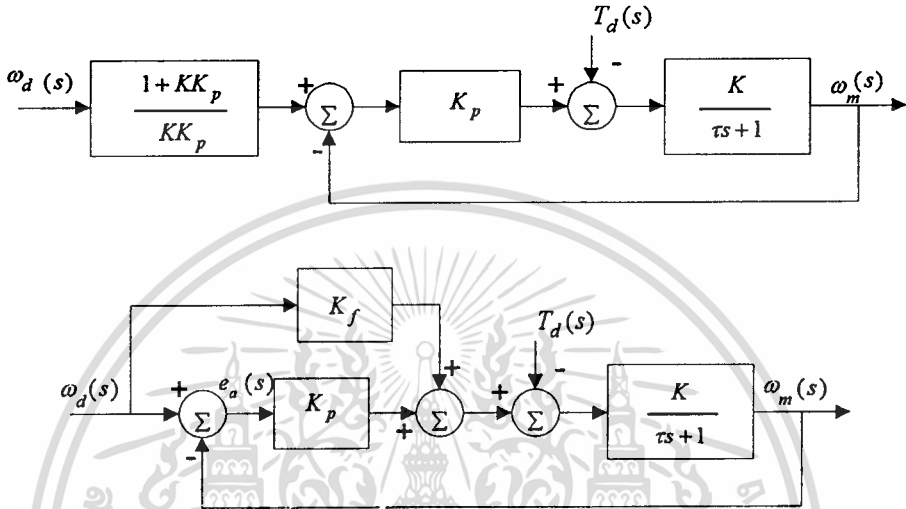
จากสมการที่ 3.38 และ 3.41 สามารถหาค่าความผิดพลาดของความเร็วในสภาวะอยู่ตัวได้ดังนี้

$$e_{ss} = \omega_m - \omega_{ss} = \frac{1}{1+KK_p} \quad (3.45)$$

การออกแบบระบบควบคุมนี้เราสามารถออกแบบเพื่อทำให้ความผิดพลาดในสภาวะอยู่ตัวเป็นศูนย์ได้โดยการเพิ่มส่วนที่เป็นการควบคุมที่ป้อนสัญญาณไปข้างหน้า (feedforward) เข้าไปในระบบหรืออีกในทำนองหนึ่งก็คือเราจะลดผลกระทบจากการเพิ่มค่าความเร็วอ้างอิงให้สูงขึ้นกว่าความเร็วที่ต้องการ โดยจะให้สูงกว่าเป็น $\frac{1+KK_p}{KK_p}$ เท่าของความเร็วที่ต้องการ ดังแสดงในภาพที่ 3.4 วิธีการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นพื้นฐานของหลักการที่เรียกว่าการชดเชยอินพุตซึ่งจะใช้ในการออกแบบระบบควบคุมที่มีการปรับอัตราโหม้ติเพื่อให้สัญญาณผิดพลาดในสภาวะอยู่ตัวมีค่าเป็นศูนย์ แต่วิธีการนี้มีข้อเสีย คือว่าเราจำเป็นต้องรู้ก่อนที่จะทำการป้อนสัญญาณไปข้างหน้าว่าขนาดของสัญญาณหรือค่าอัตราขยายในส่วนที่ไปข้างหน้ามีค่าเท่าไร นั่นก็คือเราจะต้องรู้ขนาดของสัญญาณผิดพลาดในสภาวะคงตัวก่อนเพื่อนำมาคำนวณหาอัตราขยายในส่วนที่ป้อนไปข้างหน้า (feedforward gain)



ภาพที่ 3.4 การควบคุมแบบป้อนไปข้างหน้า

สมการฟังก์ชันถ่ายโอนสามารถเขียนได้ดังนี้คือ

$$\frac{\omega_m(s)}{\omega_d(s)} = \frac{K(K_p + K_f)}{\tau s + 1 + KK_p} \quad (3.46)$$

เพราะฉะนั้นถ้าเราเลือกอัตราขยายของส่วนที่ป้อนไปข้างหน้ามีค่าเท่ากับ $1/K$ เราสามารถแสดงได้ว่าค่าความเร็วผิดพลาดในสภาวะอยู่ตัวจะมีค่าเป็นศูนย์ตามต้องการ ซึ่งจะเห็นว่าค่าคงตัว K นี้จะขึ้นอยู่กับค่าพารามิเตอร์ของระบบมอเตอร์ด้วยซึ่งส่วนมากแล้วการจะหาค่า K ให้ถูกต้องแม่นยำนั้นเป็นไปได้ค่อนข้างยาก ดังนั้นการทำการควบคุมที่มีส่วนป้อนไปข้างหน้าในทางปฏิบัติแล้วยังคงไม่สามารถทำให้ค่าความเร็วผิดพลาดในสภาวะอยู่ตัวเป็นศูนย์ได้อย่างสมบูรณ์

3.4.1.3 (ตัวควบคุมเชิงสัดส่วนเมื่อสัญญาณอินพุตเป็นฟังก์ชันแรมป์)

การใช้สัญญาณอินพุตเป็นสัญญาณฟังก์ชันแรมป์ $\left(\frac{A}{s^2}\right)$ นี้จะช่วยให้เราตรวจสอบระบบ

ได้ว่าระบบควบคุมที่เราออกแบบนั้นมีความสามารถในการติดตามสัญญาณคำสั่งที่มีการเปลี่ยนแปลง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลงแบบคงที่ได้อย่างไร ถึงแม้ว่าสัญญาณอินพุตที่เป็นฟังก์ชันแรมปีนี้จะมีค่าเพิ่มขึ้นตลอดเวลาก็ตามแต่ ในการวิเคราะห์ของเรานั้นสนใจเฉพาะช่วงเวลาที่การตอบสนองของระบบขึ้นกับเวลาหรือในช่วงโพลด์ชั่วคราว (transient response) เท่านั้น จากแผนภาพ บล็อกข้างต้นเราสามารถสรุปได้ว่า

$$\frac{e_s(s)}{\omega_d(s)} = \frac{\tau s + 1 - KK_f}{\tau s + 1 + KK_p} \quad (3.47)$$

ดังนั้นถ้า $\omega_m(s) = \frac{A}{s}$ เราสามารถหาความเร็วผิดพลาดในสภาวะอยู่ตัวได้ดังนี้

$$e_{ss} = \begin{cases} \infty & K_f \neq \frac{1}{K} \\ \frac{A\tau}{1+KK_p} & K_f = \frac{1}{K} \end{cases} \quad (3.48)$$

จากข้างต้นถึงแม้ว่าค่าความผิดพลาดในสภาวะอยู่ตัวจะมีค่าเป็นอนันต์เมื่อค่า $K_f \neq \frac{1}{K}$ แต่ในทางปฏิบัติแล้วในการควบคุมความเร็วของมอเตอร์นั้นจะมีด้วยกันสามช่วง คือช่วงความเร่ง ช่วงความเร็วคงที่ และช่วงความหน่วงคงที่ จะเห็นว่าในช่วงความเร่งนั้นเราจะเร่งด้วยความเร่งคงที่ในขณะที่เดียวกันความเร็วก็จะเพิ่มขึ้นเป็นแบบสัญญาณฟังก์ชันแรมปี ซึ่งจะเพิ่มขึ้นจนถึงเวลาหนึ่งสมมติว่าเป็น t_1 จากเวลาเริ่มต้นเคลื่อนที่ ดังนั้นเมื่อเวลา $t > t_1$ แล้วความเร็วจะคงที่หรือความเร่งเป็นศูนย์นั่นเอง ฉะนั้นเราสามารถหาสัญญาณผิดพลาด ที่เวลา t_1 นี้ได้ สมมติว่ามีขนาด $|e_{ss}(t_1)|$ เราสามารถทำให้ $|e_{ss}(t_1)| < a$ ได้โดยการใช้การป้อนสัญญาณไปข้างหน้า และเลือกค่าอัตราขยาย K_p ที่เหมาะสม แต่อย่างไรก็ตามวิธีนี้จำเป็นต้องมีการทดลองหาแบบลองผิดลองถูก (trial-and-error) คือทำโดยทดลองเปลี่ยนค่าไปเรื่อยๆ หรือบางครั้งอาจจะสามารถคำนวณหาโดยวิธีการพีชคณิต ถ้าเรารู้ค่าอัตราขยาย K ของระบบแม่นยำแล้ว วิธีที่สะดวกที่สุดก็คือเลือกค่า $K_f = \frac{1}{K}$ แล้วเลือกค่า K_p ได้จากสมการ

$$e_{ss} = \frac{A\tau}{1+KK_p} = a \quad (3.49)$$

3.4.1.4 (การควบคุมเชิงสัดส่วนกับระบบอันดับสอง)

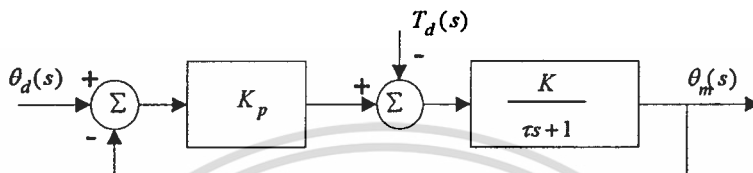
ถ้าต้องการควบคุมตำแหน่งของการหมุนของมอเตอร์ระบบพลศาสตร์ของมอเตอร์จะกลายเป็นระบบอันดับสอง สมการฟังก์ชันถ่ายโอน สามารถเขียนได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\theta_m(s)}{\theta_d(s)} = \frac{KK_p}{\tau s^2 + s + KK_p} \quad (3.50)$$

และ

$$\frac{\theta_m(s)}{T_d(s)} = -\frac{K}{\tau s^2 + s + KK_p} \quad (3.51)$$



ภาพที่ 3.5 การควบคุมตำแหน่งของการหมุนทางพลศาสตร์ของมอเตอร์กระแสตรง

ถ้ากำหนดให้ $\theta_d(s) = \frac{1}{s}$ สามารถคำนวณหาค่าในสภาวะอยู่ตัวได้ดังนี้

$$\theta_{ss} = \frac{KK_p}{KK_p} = 1 \quad (3.52)$$

และถ้ามีโหลดเปลี่ยนแปลงเป็นกระทันหันค่าคือ $T_d = \frac{1}{s}$ เราจะได้ว่าตำแหน่งในสภาวะอยู่ตัวมีค่าเท่ากับ

$$\theta_{ssT_d} = \frac{1}{K_p} \quad (3.53)$$

ในตัวอย่างนี้จะเห็นว่าค่าผิดพลาดในสภาวะอยู่ตัวนั้น (โดยไม่คำนึงถึงผลจากโหลดเปลี่ยนแปลง) จะมีค่าเป็นศูนย์สำหรับระบบ $G(s) = \frac{K}{s(\tau s + 1)}$ ซึ่งเป็นระบบชนิด type 1 แต่ถ้าระบบเป็น

$G(s) = \frac{1}{Js^2 + cs + K}$ ซึ่งเป็นระบบชนิด type 0 แล้วค่าผิดพลาดหรือออฟเซต (offset) จะไม่เป็นศูนย์

ฟังก์ชันถ่ายโอนของระบบควบคุมแบบป้อนกลับในตัวอย่างนี้เราสามารถหาค่าอัตราส่วนการหน่วง (ξ) ของระบบควบคุมแบบป้อนกลับได้ดังนี้

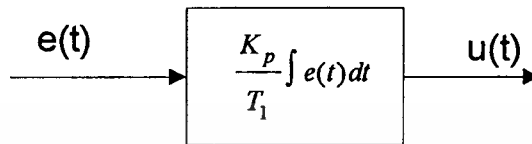
$$\xi = \frac{1}{2\sqrt{\tau KK_p}} \quad (3.54)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราสามารถออกแบบให้การตอบสนองในช่วงสภาวะชั่วคราว ของระบบควบคุมมีค่าอัตราส่วน การหน่วงตามต้องการได้โดยการเลือกค่า K_p ที่เหมาะสม

3.4.2 (การควบคุมแบบอินทิกรัล)

การควบคุมแบบอินทิกรัล (Integral control) นั้นมีรูปแบบของสมการดังนี้คือ



$$u(t) = \frac{K_p}{T_1} \int e(t) dt \quad (3.55)$$

หรือในรูปแบบของสมการฟังก์ชันถ่ายโอน จะได้ว่า

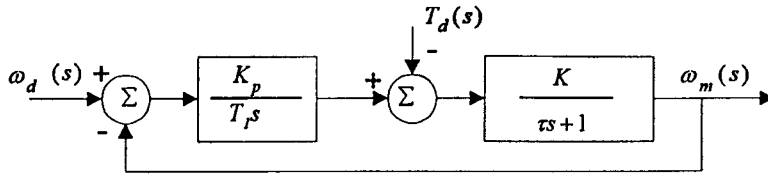
$$\frac{U(s)}{E(s)} = \frac{K_p}{T_1 s} \quad (3.56)$$

โดยที่ $T_1 =$ integral time or reset time

$$\frac{1}{T_1} = \text{reset rate}$$

จากรูปแบบของการควบคุมแบบอินทิกรัล นี้จะเห็นว่าสัญญาณควบคุม $u(t)$ อาจจะมีค่าค่อนข้างมากก็ได้โดยไม่ลดลง ทั้งๆ ที่สัญญาณผิดพลาด $e(t)$ มีค่าเป็นศูนย์ในภายหลังแล้วก็ตาม ทั้งนี้ก็เพราะว่าสัญญาณควบคุมในกรณีของการควบคุมแบบอินทิกรัลขึ้นอยู่กับค่าในอดีต (past value) ไม่เหมือนกับการควบคุมเชิงสัดส่วน ซึ่งจะขึ้นอยู่กับค่าปัจจุบัน ประโยชน์ของการควบคุมแบบอินทิกรัล ก็เพื่อต้องการลดค่าความผิดพลาดในสภาวะอยู่ตัว ในขณะที่เดียวกันค่าความมีเสถียรภาพของระบบก็จะลดน้อยลงด้วย ระบบควบคุมความเร็วมอเตอร์พร้อมการควบคุมแบบอินทิกรัลแสดงไว้ในภาพที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 ระบบควบคุมแบบป้อนกลับโดยใช้ตัวควบคุมแบบอินทิกรัล

ซึ่งสามารถเขียนสมการฟังก์ชันถ่ายโอน ได้ดังนี้คือ

$$\frac{\omega_m(s)}{\omega_d(s)} = \frac{KK_p}{\tau T_i s^2 + T_i s + KK_p} \quad (3.57)$$

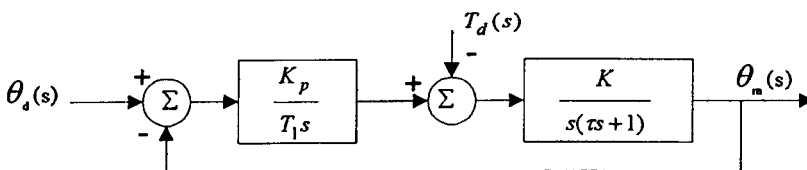
และจะได้เป็น

$$\frac{\omega_d(s)}{T_d(s)} = \frac{Ks}{\tau T_i s^2 + T_i s + KK_p} \quad (3.58)$$

จากสมการฟังก์ชันถ่ายโอนนี้เราสามารถแสดงได้ว่า ω_{ss} จะมีค่าเท่ากับ 1 เมื่อ $\omega_d = \frac{1}{s}$ และเมื่อมี โหลดเปลี่ยนแปลงทันทีหรือ $T_i = \frac{1}{s}$ ค่าความเร็วผิดพลาดในสภาวะอยู่ตัวหรือค่าที่เบี่ยงเบนไปเนื่องจากการเปลี่ยนแปลงโหลดนี้ก็สามารถถูกปรับให้กลับมาเป็นศูนย์ได้ ข้อสังเกตข้อหนึ่งของระบบควบคุมแบบนี้ก็คือจะเห็นว่าการ ควบคุมแบบอินทิกรัลนี้จะเป็นการเพิ่มอันดับให้กับระบบของเรา เช่น ในกรณีนี้ระบบเป็นอันดับที่หนึ่งและเมื่อใช้การควบคุมแบบอินทิกรัลแล้วระบบป้อนกลับจะเป็นอันดับที่สอง

3.4.2.1 (การควบคุมแบบอินทิกรัล กับระบบอันดับที่สอง)

จากภาพที่ 3.7 แสดงการควบคุมตำแหน่งการหมุนของมอเตอร์ดังได้กล่าวมาแล้วแต่ ในที่นี้จะใช้การควบคุมแบบอินทิกรัล



ภาพที่ 3.7 การควบคุมมอเตอร์กระแสตรงด้วยตัวควบคุมแบบอินทิกรัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสามารถหาสมการฟังก์ชันถ่ายโอน ได้ดังนี้คือ

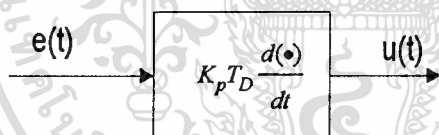
$$\frac{\theta_m(s)}{\theta_d(s)} = \frac{KK_p}{T_1 T_2 s^3 + T_1 s^2 + KK_p} \quad (3.59)$$

และจะได้

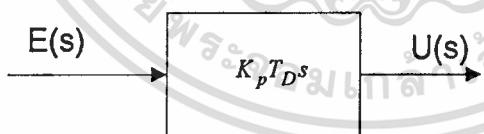
$$\frac{\theta_m(s)}{T_d(s)} = \frac{K}{T_1 T_2 s^3 + T_1 s^2 + KK_p} \quad (3.60)$$

จะเห็นว่าสมการฟังก์ชันถ่ายโอนของระบบควบคุมตำแหน่งการหมุนของมอเตอร์กระแสตรงนี้ ไม่สมดุล ซึ่งตรวจสอบได้โดยการใช้เกณฑ์เสถียรภาพของเรทซ์ดังได้กล่าวมาแล้ว ดังนั้นจะเห็นว่า ถึงแม้การควบคุมแบบอินทิกรัลนี้จะช่วยให้ค่าความผิดพลาดในสภาวะอยู่ตัวเป็นศูนย์ก็ตาม แต่ตัวอินทิกรัลนี้จะทำให้เสถียรภาพของระบบของเราลดลง ดังนั้นจึงไม่นิยมใช้เดี่ยวๆ แต่จะใช้ร่วมกับตัวควบคุมแบบอื่นดังจะกล่าวต่อไป

3.4.3 (การควบคุมแบบอนุพันธ์)



หรือ



$$\frac{U(s)}{E(s)} = K_p T_D s \quad (3.61)$$

โดยที่ค่า T_D = ช่วงเวลาอนุพันธ์

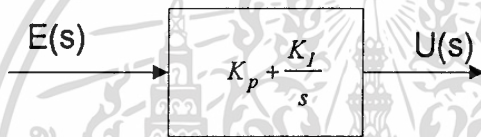
การควบคุมเชิงอนุพันธ์ (derivative control) นี้ส่วนมากแล้วจะใช้ร่วมกับตัวควบคุมตัวอื่นเช่น ใช้ร่วมกับการควบคุมเชิงสัดส่วน ซึ่งจะเรียกว่าตัวควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์หรือพีดี (PD-control) ถ้าตัวควบคุมเชิงสัดส่วนใช้ร่วมกับอินทิกรัลจะเรียกว่าตัวควบคุมแบบสัดส่วนบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ แบบอินทิกรัลหรือพีไอ (PI-control) และถ้าใช้ตัวควบคุม แบบอนุพันธ์ ร่วมกันกับตัวควบคุม แบบสัดส่วนและตัวควบคุมแบบอินทิกรัลจะเรียกว่าตัวควบคุมแบบพีไอดี (PID-control)

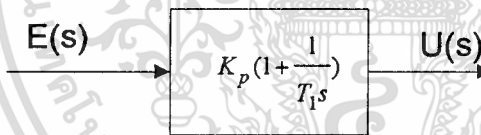
ตัวควบคุมแบบอนุพันธ์นี้จะช่วยให้เราปรับค่าความหน่วง (damping) ในระบบที่จะควบคุมให้มีค่ามากขึ้น นั่นก็คือทำให้ระบบมีเสถียรภาพเพิ่มขึ้น แต่โดยทั่วไปแล้วตัวควบคุมเชิงอนุพันธ์นี้จะไม่ทำให้ค่าความผิดพลาดในสภาวะอยู่ตัวมีค่าเป็นศูนย์ได้ ในขณะเดียวกันจะเห็นว่าสัญญาณเอาต์พุตที่ออกจากตัวควบคุมเชิงอนุพันธ์นี้เป็นสัญญาณที่เกิดจากการหาอนุพันธ์ของสัญญาณผิดพลาด $\frac{d(e(t))}{dt}$ ดังนั้นถ้าสัญญาณผิดพลาดนี้มีสัญญาณรบกวนมาก สัญญาณเอาต์พุตที่ได้จากตัวควบคุมเชิงอนุพันธ์นี้จะกระเพื่อม (fluctuate) ค่อนข้างมาก (เนื่องจากค่าความชัน (slope) เปลี่ยนแปลงค่อนข้างมาก) ซึ่งจะทำให้ระบบควบคุมอาจไม่มีเสถียรภาพได้

3.4.4 (การควบคุมแบบสัดส่วนบวกกับอินทิกรัล)



$$\frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} \quad (3.62)$$

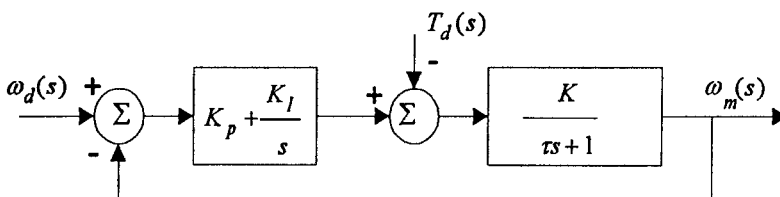
หรือ



$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_I s}\right) \quad (3.63)$$

3.4.4.1 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลกับระบบอันดับหนึ่ง)

ในที่นี้เราจะมาพิจารณาระบบควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัล (PI-control) กับการควบคุมความเร็วของมอเตอร์ดังแสดงไว้ในแผนภาพบล็อกข้างล่างนี้



ภาพที่ 3.8 การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนภาพบล็อกเราสามารถเขียนสมการฟังก์ชันถ่ายโอนได้ดังนี้คือ

$$\frac{\omega_m(s)}{\omega_d(s)} = \frac{K(K_p s + K_i)}{\tau s^2 + (1 + KK_p)s + KK_i} \tag{3.64}$$

และจะได้เป็น

$$\frac{\omega_m(s)}{\tau_d(s)} = \frac{Ks}{\tau s^2 + (1 + KK_p)s + KK_i} \tag{3.65}$$

จากสมการฟังก์ชันถ่ายโอนนี้เราสามารถแสดงว่า ω_{ss} จะมีค่าเท่ากับ 1 เมื่อ $\omega_d = \frac{1}{s}$ และเมื่อมีโพลด์เปลี่ยนแปลงทันทีหรือ $\tau_d = \frac{1}{s}$ ค่าความเร็วผิดพลาดหรือที่เบี่ยงเบนไปเนื่องจากการเปลี่ยนแปลงโพลด์นี้ก็สามารถถูกปรับให้กลับมามีค่าเป็นศูนย์ได้ ซึ่งเป็นผลเนื่องมาจากตัวอินทิกรัลและดังที่กล่าวมาแล้วจะเห็นว่าการควบคุมแบบอินทิกรัลนี้จะเป็นการเพิ่มอันดับให้กับระบบของเราเช่นในกรณีนี้ระบบเป็นอันดับหนึ่งและเมื่อใช้การควบคุมแบบอินทิกรัลแล้วระบบป้อนกลับจะมีอันดับเป็นสอง ดังนั้นค่าสัดส่วนการหน่วง (damping ratio) สามารถเขียนได้ดังนี้

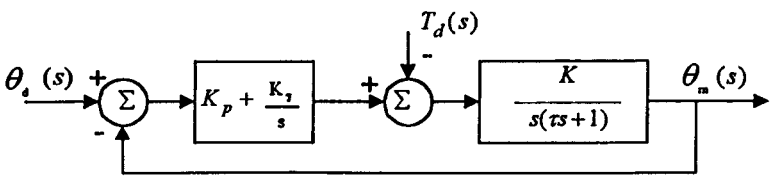
$$\xi = \frac{1 + KK_p}{2\sqrt{KK_i}\tau} \tag{3.66}$$

ค่าเวลาคงที่ของระบบควบคุมแบบป้อนกลับนี้เป็นกรณีหน่วงต่ำกว่าวิกฤต สามารถหาได้ดังนี้

$$\tau_{cl} = \frac{2\tau}{1 + KK_p} \tag{3.67}$$

3.4.4.2 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลกับระบบอันดับสอง)

การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัล (PI-control) นั้นจะทำให้ระบบควบคุมของเรามีอันดับเป็นสาม ดังภาพที่ 3.9



ภาพที่ 3.9 การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PI

เราสามารถเขียนสมการฟังก์ชันถ่ายโอนได้ดังนี้คือ

$$\frac{\theta_m(s)}{\theta_d(s)} = \frac{K(K_p + K_i s)}{\tau s^3 + s^2 + KK_p s + KK_i} \quad (3.68)$$

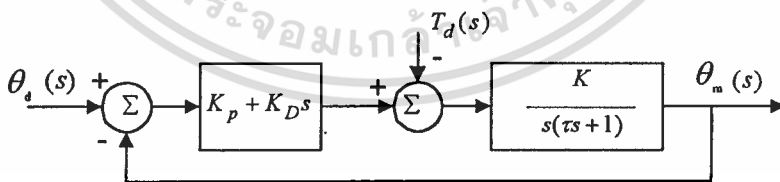
หรือจัดใหม่ได้เป็น

$$\frac{\theta_m(s)}{T_d(s)} = \frac{Ks}{\tau s^3 + s^2 + KK_p s + KK_i} \quad (3.69)$$

ซึ่งจะเห็นว่าค่าความผิดพลาดในสภาวะอยู่ตัวจะมีค่าเป็นศูนย์เช่นเดียวกันเนื่องจากตัวอินทิกรัลและรวมถึงการเปลี่ยนแปลงตำแหน่งของการหมุนที่อาจจะเกิดขึ้นเนื่องจากโหลดเปลี่ยนแปลงก็สามารถปรับเข้าสู่ค่าที่ต้องการได้เช่นกัน ถ้าอาศัยเกณฑ์เสถียรภาพของเราที่มาตรวจสอบความเสถียรภาพของระบบ จะเห็นว่าการออกแบบนั้นจะต้องทำให้ค่า $KK_p - \tau KK_i > 0$ ระบบควบคุมจึงจะมีเสถียรภาพได้ ซึ่งทำได้โดยให้ค่าอัตราขยายของส่วนที่เป็นตัวควบคุมเชิงสัดส่วน (K_p) มีค่ามากไว้ก่อน

3.4.4.3 (การควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์ กับระบบอันดับสอง)

การควบคุมแบบสัดส่วนบวกกับแบบอนุพันธ์ นี้ไม่นิยมใช้ควบคุมระบบที่มีอันดับหนึ่ง การควบคุมตำแหน่งการหมุนของมอเตอร์โดยใช้การควบคุมแบบสัดส่วนบวกกับอนุพันธ์นั้น แสดงได้ดังแผนภาพบล็อกข้างล่างดังนี้



ภาพที่ 3.10 การควบคุมตำแหน่งการหมุนของมอเตอร์โดยใช้การควบคุมแบบ PD

สามารถเขียนสมการฟังก์ชันถ่ายโอนได้ดังนี้

$$\frac{\theta_m(s)}{\theta_d(s)} = \frac{K(K_p + K_D s)}{\tau s^2 + (1 + KK_D)s + KK_p} \quad (3.70)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และอินพุตต่อสัญญาณรบกวนจะได้ว่า

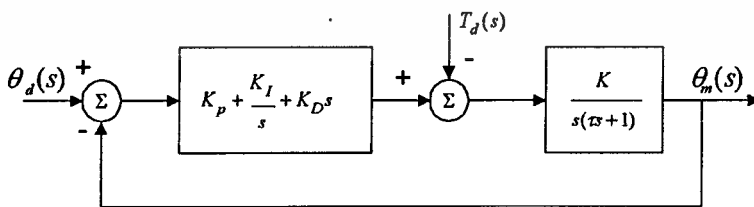
$$\frac{\theta(s)}{T_d(s)} = -\frac{Ks}{\tau s^2 + (1 + KK_p)s + KK_p} \quad (3.71)$$

ค่าความเร็วในสภาวะอยู่ตัวเนื่องมาจากสัญญาณอินพุตเป็นฟังก์ชันขั้นหนึ่งหน่วยนั้นจะมีค่า เป็น 1 ซึ่งก็หมายความว่าค่าผิดพลาดของตำแหน่งการหมุนในสภาวะคงตัว มีค่าเป็นศูนย์ ส่วนค่าผิดพลาดของตำแหน่งในสภาวะอยู่ตัวเนื่องจากมีโพลเปลี่ยนแปลงกระตุ้นหันทันนั้นจะมีค่าน้อยลง เมื่อค่าอัตราขยายของตัวควบคุมเชิงสัดส่วนมีดีมาก ซึ่งเป็นลักษณะเช่นเดียวกับการควบคุมที่ใช้ตัวควบคุมแบบสัดส่วนอย่างเฉียว และจะพบว่าระบบจะมีเสถียรภาพถ้าอัตราขยาย K_p และ K_D มีค่าเป็นบวก ค่าอัตราส่วนการหน่วงสามารถหาได้ดังนี้คือ

$$\xi = \frac{1 + KK_p}{2\sqrt{\tau KK_p}} \quad (3.72)$$

3.4.4.4 (การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัลบวกกับแบบอนุพันธ์กับระบบอันดับสอง)

การควบคุมแบบสัดส่วนบวกกับแบบอินทิกรัล (PI-control) กับระบบอันดับสองข้างต้นจะเห็นว่าเรายังไม่สามารถบรรลุการออกแบบที่สมบูรณ์ เนื่องจากว่าค่าคงตัวเวลา (τ) ของระบบมีค่ามาก ระบบอาจจะไม่มีเสถียรภาพก็ได้ในหัวข้อนี้เราจะเพิ่มตัวควบคุมเชิงอนุพันธ์เข้าไปในการควบคุมตำแหน่งการหมุนของมอเตอร์ ซึ่งแสดงไว้ในแผนภาพบล็อกข้างล่างนี้



ภาพที่ 3.11 การควบคุมตำแหน่งการหมุนของมอเตอร์ด้วยตัวควบคุม PID

ซึ่งเราสามารถเขียนสมการฟังก์ชันถ่ายโอนได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{\theta_o(s)}{\theta_m(s)} = \frac{K(K_D s^2 + K_p s + K_i)}{\tau s^3 + (1 + KK_D)s^2 + KK_p s + KK_i} \quad (3.73)$$

$$\frac{T_o(s)}{\theta_o(s)} = \frac{Ks}{\tau s^3 + (1 + KK_D)s^2 + KK_p s + KK_i} \quad (3.74)$$

ซึ่งค่าผิดพลาดในสภาวะคงตัว (steady state error) อันเนื่องมาจากสัญญาณอินพุตที่เป็นฟังก์ชันขั้นหนึ่งหน่วยจะเป็นศูนย์เนื่องมาจากมีตัวควบคุมแบบอินทิกรัล ระบบนี้จะมีเสถียรภาพเมื่อค่าอัตราขยาย K_p , K_i และ K_D มีค่าเป็นบวกและ

$$(1 + KK_D)KK_p - \tau KK_i > 0 \quad (3.75)$$

ซึ่งค่าอัตราขยาย K_D ที่อยู่ในระบบควบคุมนี้จะช่วยทำให้สมการ (3.75) ข้างต้นเป็นไปได้

3.5 การเลือกค่าอัตราขยายของตัวควบคุม

ในหัวข้อข้างต้นนั้นเราได้เรียนรู้รูปแบบของตัวควบคุมพื้นฐาน ในการออกแบบนั้นเราต้องเลือกรูปแบบของตัวควบคุมที่เหมาะสมกับระบบของเรา การออกแบบตัวควบคุมจะสมบูรณ์ได้เราจำเป็นต้องต้องเลือกค่าอัตราขยายต่างๆ ของตัวควบคุมนั้น การเลือกค่าอัตราขยายหรือการคำนวณหาอัตราขยายนั้นก็ขึ้นอยู่กับคุณสมบัติเฉพาะของการตอบสนองในช่วงภาวะชั่วคราว ตัวอย่างเช่น สำหรับตัวควบคุมแบบพีไอดี (PID-control) นั้นเราอาจจะกำหนดค่าคุณสมบัติเฉพาะของการตอบสนองในช่วงภาวะชั่วคราวนี้คือ ค่าอัตราส่วนการหน่วง, ค่าคงตัวเวลาที่ช้าที่สุด (dominant time constant) และค่าสัญญาณผิดพลาดในสภาวะอยู่ตัว (steady state error) ในทางปฏิบัตินั้นการกำหนดคุณสมบัติเฉพาะจะมีรายละเอียดมากกว่านี้ เช่นอาจจะต้องคำนึงถึง ช่วงเวลาขึ้น (rise time), เวลาเข้าที่ (settling time), และค่าโอเวอร์ชูทสูงสุด (maximum overshoot) ด้วย นอกจากนี้สมรรถนะการทำงานนั้นอาจจะบอกอยู่ในรูปแบบของความถี่ก็ได้เช่น ช่วงความถี่การทำงาน (bandwidth), ความถี่เรโซแนนซ์ (resonant frequency) และค่าแอมป์พลิจูดสูงสุด (peak amplitude) เป็นต้น ซึ่งอาจจะสรุปลักษณะของคุณสมบัติเฉพาะต่างๆ เหล่านี้ที่เกี่ยวข้องกับการทำงานของระบบควบคุมเป็นสิ่งที่เราจะต้องคำนึงถึงในการออกแบบตัวควบคุมได้ดังนี้

1) คุณสมบัติที่ตำแหน่งสมดุล (Equilibrium specification)

- ความมีเสถียรภาพ (stability)
- ความผิดพลาดในสภาวะอยู่ตัว (steady state error)

2) คุณสมบัติช่วงการตอบสนองในภาวะชั่วคราว (transient specification)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

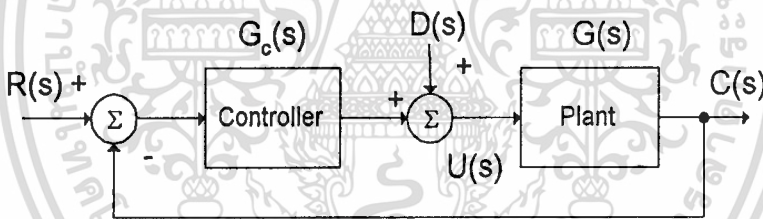
- ความเร็วในการตอบสนอง (speed of response)
 - รูปแบบของการตอบสนองซึ่งอาจจะบอกในรูปแบบของค่าการหน่วง(degree of damping)
- 3) คุณสมบัติที่บอกความไวของระบบ (sensitivity specification)
- ความไวของระบบที่ขึ้นอยู่กับ การเปลี่ยนแปลงค่าพารามิเตอร์ของระบบ (sensitivity to parameter variation)
 - ความไวของระบบที่ขึ้นอยู่กับ การหาค่าพารามิเตอร์ของระบบ ไม่ละเอียด (sensitivity to model inaccuracies)
 - การสามารถในการกำจัดสัญญาณรบกวน (Noise rejection) ซึ่งอาจจะบอกเป็นช่วงความถี่ (band width)
- 4) ผลอันเนื่องมาจากความไม่เป็นเชิงเส้นของระบบ (nonlinear effects)
- ความมีเสถียรภาพ (stability)
 - ความสามารถของอุปกรณ์ควบคุม (final control element capabilities)

นอกเหนือจากการกำหนดคุณสมบัติเฉพาะของการตอบสนองช่วงภาวะชั่วคราวแล้ว ในบางครั้ง เราไม่สามารถโมเดลแบบจำลองทางคณิตศาสตร์ของระบบที่เราจะควบคุมได้แม่นยำ ในวิธีข้างต้น อาจจะมี ความยากลำบากบ้างในการที่จะคำนวณออกแบบค่าอัตราขยายต่างๆ เพื่อให้ระบบควบคุมของเรามีการตอบสนองอย่างที่เราต้องการเนื่องจากระบบควบคุมบางชนิดมีค่าอยู่หลายค่าที่ต้องการ การปรับเพื่อให้การทำงานเป็นไปตามความต้องการ ในกรณีนี้การออกแบบอาจจะกระทำได้โดยการทดสอบตัวควบคุมมาตรฐานเชิงอุตสาหกรรม (standard industrial controller) กับระบบจริงและนำผลลัพธ์ที่วัดจากการทำงานจริงมาเพื่อปรับ (tuning) ตัวควบคุมเพื่อให้ระบบทำงานได้ตามต้องการ แนวทางหรือวิธีการในการปรับตัวควบคุมโดยใช้ผลลัพธ์ที่วัดจากการทำงานจริง แต่อย่างไรก็ตามวิธีหลังนี้บางครั้งอาจจะต้องสิ้นเปลืองค่าใช้จ่ายค่อนข้างมาก เพราะต้องการสร้างระบบพร้อมตัวควบคุมขึ้นมาก่อนจึงจะทำการทดสอบได้ และการทดสอบตัวควบคุมกับการทำงานจริงเพื่อปรับค่าอัตราขยายในตัวควบคุมนั้นส่วนมากจะใช้เวลาและสิ้นเปลืองค่าใช้จ่ายค่อนข้างสูง

บทที่ 4

การควบคุมแบบ Proportional, Integral, Derivative, Acceleration

การออกแบบตัวควบคุม PIDA ได้รับการตีพิมพ์เดือนธันวาคม ปี 1996 โดย C.Dorf [12,13] ซึ่งคิดค้นขึ้นมา เพื่อนำไปออกแบบการควบคุมกระบวนการที่มีอันดับสามโดยใช้เทคนิค และวิธีการของ PID ซึ่งตัวควบคุม PID [11,15] ถูกใช้กันอย่างกว้างขวางในโรงงานอุตสาหกรรม เพราะสามารถสร้าง ขึ้นได้ง่ายสำหรับชนิดของกระบวนการที่เป็นอันดับสอง. ในกรณีเดียวกันกับระบบที่มีแบบจำลองของกระบวนการเป็นอันดับสาม. อันที่จริงระบบควบคุมการเคลื่อนที่ที่มีลักษณะเช่นเดียวกันกับการควบคุมตำแหน่งการหมุนของมอเตอร์ให้ถูกต้อง ซึ่งมีแบบจำลองของกระบวนการเป็นอันดับสาม โดยจะไม่เหมือนกับที่เคยมีใน [10,15] ซึ่งเป็นอีกทางเลือกหนึ่งที่ทำให้ทราบว่าแท้จริงแล้วเป็นเรื่องที่ยากมากๆ ที่จะออกแบบตัวควบคุม PID สำหรับกระบวนการที่เป็นอันดับสาม เนื่องจากอันดับของกระบวนการที่มีจำนวนโพลมากกว่าจำนวนของ ซีโร่ ซึ่งถ้าหากพิจารณาตัวควบคุม PID [10,14] ทำให้เกิดทางเดินของรากเคลื่อนเข้าใกล้จุดกำเนิดดังเอกสารที่ [11] ดังนั้นตัวควบคุมสำหรับกระบวนการอันดับสามจึงมีข้อกำหนดที่จะต้องการควบคุมเป็นโรบัสต์



ภาพที่ 4.1 โครงสร้างของระบบควบคุมในวงปิดแบบทั่วไป

4.1 การออกแบบตัวควบคุม PIDA ด้วยวิธีพิจารณาค่ารากหลัก Dominant Roots

ในหัวข้อต่อไปนี้จะเป็นการนำเสนอวิธีการออกแบบตัวควบคุม PIDA ซึ่งเป็นการเพิ่มซีโร่ เข้าด้วยกันกับตัวควบคุม PID. แนวความคิดการออกแบบตัวควบคุม PIDA เป็นการประยุกต์ใช้งานของกราฟรูตโพลใน s -plane ดังแสดงในภาพที่ 4.2 เมื่อโพลของกระบวนการที่มากที่สุดไม่ได้อยู่ที่ตรงจุดกำเนิดแต่มีค่าน้อยกว่า $-\zeta\omega_n$ ดังนั้นจะแทนตำแหน่งของโพลทั้งสองที่ให้ครอบคลุมโดยนำตัวที่หนึ่งวางที่ตำแหน่งด้านซ้ายโพล ของกระบวนการที่มากที่สุด และอีกตัวหนึ่งวางให้ไกลออกจากจุดกำเนิดใน s -plane ที่มีผลในการลดจำนวนโพล ทำให้มีผลกระทบต่อระบบทั้งหมด โดยตำแหน่งใหม่ของโพลดังกล่าวจะเข้ามามีผลกระทบต่อระบบแทน ซึ่งสามารถปรับแต่งตำแหน่งด้วยการใช้การออกแบบตัวควบคุม PIDA นอกจากนั้นผลกระทบดังกล่าวจะปรากฏมากขึ้นกับกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะโดยวิธีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดที่หนึ่ง และมากที่สุดกับกระบวนการชนิดที่สอง เนื่องมาจากโพลที่จุดกำเนิด

4.2 ทฤษฎีและหลักการของ PIDA

พิจารณาโครงสร้างที่แสดงการควบคุมในภาพที่ 1. เมื่อมีสัญญาณป้อนกลับเป็นหนึ่งหน่วย (unity feedback) ตัวชดเชย PIDA กำหนดให้เป็นดังนี้

$$G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{(s+d)} + \frac{K_A s^2}{(s+d)(s+e)} \quad (4.1)$$

$$= K \frac{(s+a)(s+b)(s+z)}{s(s+d)(s+e)} \quad a, b, z \ll d, e \quad (4.2)$$

เมื่อ a, b, z และ d, e เป็นซีโร และ โพลของตัวควบคุม PIDA ตามลำดับ โดยเหตุที่ว่า $a, b, z \ll d, e$. พิจารณาที่โพล d, e มีค่าเล็กน้อยมากจึงไม่นำมาคิด ซึ่งจุดมุ่งหมายหลักของการออกแบบตัวควบคุม PIDA นั้นเป็นการคำนวณหาค่าตัวควบคุมที่สามารถเลือกค่าของตัวควบคุมที่จะสามารถแสดงสภาวะชั่วคราวได้ตามต้องการ ดังนั้นฟังก์ชันถ่ายโอน $T(s)$ ในวงปิด จะเป็น

$$T(s) = \frac{G(s)G_c(s)}{1+G(s)G_c(s)} \quad (4.3)$$

เมื่อ $1+G(s)G_c(s) = 0$ เป็นสมการคุณลักษณะ ระบบควบคุมที่แสดงในภาพที่ 4.1 ซึ่งมักจะกำหนดโดยผลตอบสนองชั่วคราวด้วยการเพิ่มค่าพารามิเตอร์ต่างๆ เช่น เปอร์เซ็นโอเวอร์ชูท (P.O), เวลาเข้าที่ (T_s) และ เวลาคายอด (T_p) ดังนั้นจึงกำหนดค่าให้มีคุณสมบัติดังต่อไปนี้

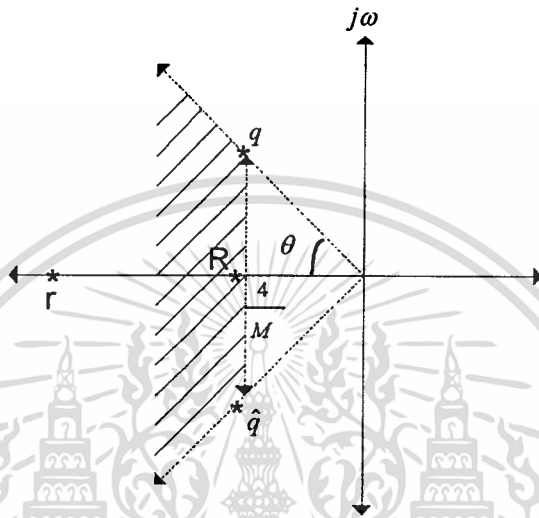
$$P.O \leq L \quad (4.4)$$

$$T_s \leq M \quad (4.5)$$

$$\text{ค่าสูงสุด} \left| \frac{\text{ผลตอบสนองเอาต์พุต}}{\text{สัญญาณรบกวน}} \right| = \max \frac{|C(t)|}{|D(t)|} < W \quad (4.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อค่าของ L, M และ W จะถูกเลือกโดยผู้ออกแบบตัวควบคุม ที่ทำให้ค่าผิดพลาดในสถานะอยู่ตัว $err = 0$ ด้วยตัวควบคุม PIDA ที่เหมาะสมกับอินพุตที่เป็นขั้น (step) ซึ่งคำนวณค่าได้อย่างรวดเร็วและแม่นยำ โดยค่า L และ M เป็นการกำหนดตำแหน่งรากที่ใกล้จุดกำเนิดที่สุดใน s -plane ดังนั้นจุดมุ่งหมายของบทความนี้จึงเป็นแนวทางในการวิเคราะห์วิธีใหม่ของการออกแบบตัวควบคุม PIDA สำหรับกระบวนการอันดับสาม ซึ่งให้คุณสมบัติเฉพาะได้ตามต้องการ



ภาพที่ 4.2 โลกัสมี่ครอบคลุมรากที่ยอมรับได้และตำแหน่งค่าราก ในวงปิดที่ตำแหน่ง (*) ใน s -plane

4.3 ขั้นตอนการออกแบบตัวควบคุม PIDA

ขั้นตอนที่ 1. กำหนด $\xi\omega_n$ ของตำแหน่งรากที่ครอบคลุมจากเวลาเข้าที่ (T_s) ตามสมการ

$$T_s = \frac{4}{\xi\omega_n} \quad \text{โดยที่ตำแหน่งรากครอบคลุมนั้นจะให้ } s=q \text{ และ } q^* \text{ เมื่อ } q = -\xi\omega_n + j\omega_n\sqrt{1-\xi^2}$$

ขั้นตอนที่ 2. กำหนดค่า ζ ที่ตำแหน่งของเปอร์เซ็นต์โอเวอร์ชูท (P.O) นั่นคือ

$$\zeta = \frac{\left(\ln \frac{L}{100}\right)^2}{\pi^2 + \left(\ln \frac{L}{100}\right)^2}$$

ขั้นตอนที่ 3. เลือกค่ารากในแกนจริงให้เท่ากับค่าจริงของรากที่ต้องการให้ครอบคลุมโดยกำหนดดังนี้ คือ $R = \text{Re}\{\text{dominant roots}\} \leq -\xi\omega_n$ และเพื่อที่จะลดผลที่เกิดจากรากในวงปิดที่ไม่ครอบคลุม ดังนั้นจึงกำหนดเลือกค่า R โดยอยู่ทางด้านซ้ายของโพลที่มากที่สุด ในวงเปิดของกระบวนการซึ่งไม่อยู่ที่จุดกำเนิดในครึ่งระนาบด้านซ้ายของ s -plane

ขั้นตอนที่ 4. เลือกตำแหน่งรากที่มากที่สุดบนแกนจริงที่มีค่า r โดยที่กำหนดให้ $r \ll -\xi\omega_n$

ขั้นตอนที่ 5. เขียนสมการคุณลักษณะ $1+G(s)G_c(s)=0$ และกำหนดให้ $(s+r)(s+R)(s+q)(s+q^*)=0$

ขั้นตอนที่ 6. คำนวณหาสมการคุณลักษณะจากขั้นตอนที่ 5.

ขั้นตอนที่ 7. แก่สมการจากขั้นตอนที่ 6. เพื่อหาค่าพารามิเตอร์ a , b , z และ K แล้วทำให้อยู่ในรูปฟังก์ชันถ่ายโอนในวงรอบปิดคั้งสมการ ที่ (2) ทั้ง 4. สมการ

ขั้นตอนที่ 8. พล็อตกราฟแสดงผลตอบสนอง ดังนี้

ก. สำหรับที่ $c(t)$ ให้ $r(t) = \text{unity step}$

ข. สำหรับ $c(t)$ ให้ $r(t) = 0$ และ $d(t) = \text{unity step}$

ถ้าเปอร์เซ็นโอเวอร์ชูท (P.O) ไม่เป็นที่พอใจก็ให้เพิ่มค่าเกน K ที่ต่อ cascade ดังในสมการที่ (4.1)

4.4 การจำลองแบบระบบมอเตอร์กระแสตรงชนิดแยกตัวกระตุ้นด้วยตัวควบคุม PIDA

จากการออกแบบกับแบบจำลองทางพลศาสตร์ของมอเตอร์ดีซี ซึ่งปัจจุบันมอเตอร์ชนิดนี้ได้ถูกพัฒนาขึ้นอย่างมากเพื่อใช้ในการงานอุตสาหกรรมและได้มีการผลิตมอเตอร์ที่มีแม่เหล็กถาวรซึ่งมีความเข้มสูงและตัวหมุนมีความเฉื่อยต่ำมาก ดังนั้นมอเตอร์แบบนี้จึงนิยมใช้ในการงานอุตสาหกรรมตลอดจนเครื่องมือทางการแพทย์ และอุปกรณ์ทางเม็คคาโทนิคส์ ซึ่งมอเตอร์ที่ใช้ในการจำลองแบบมีค่าพารามิเตอร์ดังนี้ ตารางที่ 4.1

ตารางที่ 4.1 ค่าพารามิเตอร์ของมอเตอร์ที่ใช้จำลองแบบ

Nameplate Separatly DC Motor	
0.75 kw, 1700 r/min, 240V	
R_a	0.0144 Ω
L_m	0.0011 H
K_m	1.27 V.s/rad
J	1.828 Kg.m ²

ดังนั้นเมื่อนำกลับไปพิจารณาให้อยู่ในรูปทรานเฟอร์ฟังก์ชันของแบบจำลองดังภาพที่ 2.8 ในบทที่ 2 ซึ่งค่า ω_n และ α ในสภาวะคงตัวหาได้จากสมการ (2.63), (2.64) และ (2.65) ดังกล่าวมาแล้วข้างต้นนั้น และสามารถคำนวณหาค่าพารามิเตอร์จากตารางที่ 4.2

จากค่าพารามิเตอร์ที่คำนวณได้จากตารางที่ 4.2 เมื่อแทนค่าลงในสมการของ $G(s)$ ของมอเตอร์หรือตามสมการที่ (2.56) จะได้ทรานเฟอร์ฟังก์ชันในวงปิดของแบบจำลองทางคณิตศาสตร์สำหรับระบบมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น ได้ดังนี้

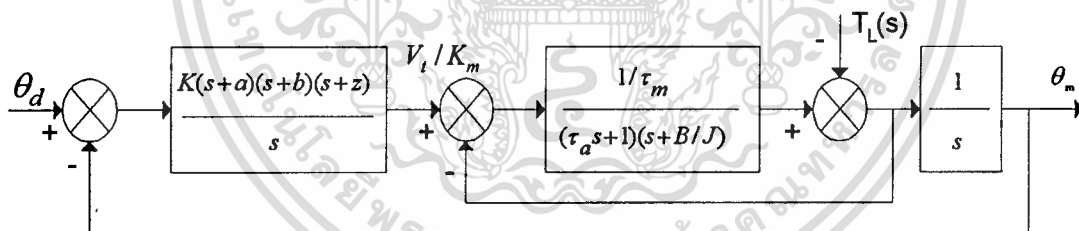
$$G(s) = \frac{209.731}{s(s^2 + 13.71s + 209.731)} \quad (4.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าพารามิเตอร์ที่ได้จากการคำนวณ

separatly excited DC motor 1750 r/min = 183 rad/s	
Rated torque T_m	$= \frac{1750}{183} = 409 \text{ N.m}$
$B = \frac{T_m}{\omega_n}$	$= 2.23 \text{ N.m.s/red}$
$J = \frac{2(1.82)}{0.738}$	$= 3.64 \text{ kg.m}$
$\frac{B}{J}$	$= 0.61$
$\tau_a = \frac{0.0022}{0.0288}$	$= 0.0764 \text{ s}$
$\frac{1}{\tau_a}$	$= 13.1 \text{ s}$
$\tau_m = 3.64 \left(\frac{0.0288}{1.27^2} \right)$	$= 0.065 \text{ s}$
$\frac{1}{\tau_m}$	$= 15.4 \text{ s}$

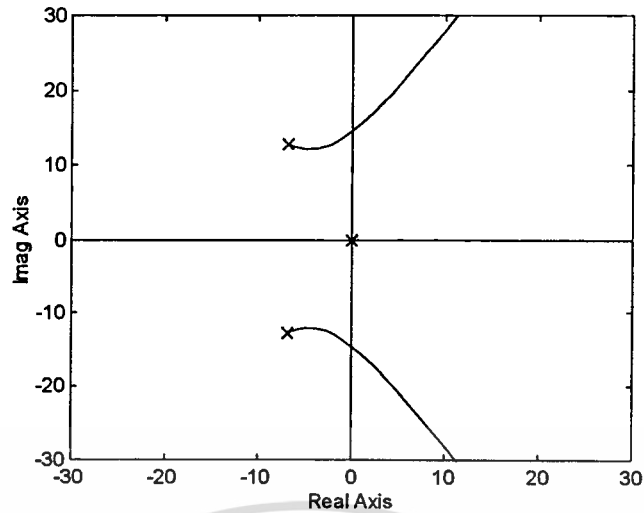
พิจารณาขั้นตอนการหาค่าตัวควบคุม PIDA ตามขั้นตอนที่ 1, 2, 3 และ 4 และสามารถ แสดง ไดอะแกรมแบบจำลองของระบบควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นได้ดังภาพที่ 4.3



ภาพที่ 4.3 ระบบควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นด้วยตัวควบคุม PIDA

จากการทดลองนั้นมีจุดประสงค์ที่ต้องการให้ $P.O. \leq 5\%$, $T_s \leq 2 \text{ sec}$ และ $\frac{|c(t)|}{|D(t)|} \leq 0.05$

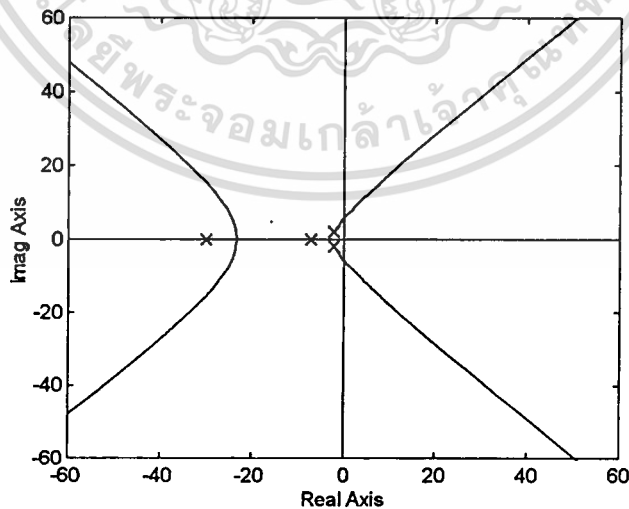
กำหนดให้ $\xi_{\omega_n} \leq -2$ และ $\xi \geq 0.707$ โดยตำแหน่งที่ครอบคลุุมรบกวนห่างจากจุดกำเนิด ดังนั้นจะต้องเลือกค่าให้ครอบคลุมค่ารากที่ $-2.1 \pm 2j$ และ จากสมการ (4.6) เมื่อพล็อตกราฟรูตโกล์ส ดังภาพที่ 4.4 จะเห็นว่าแบบจำลองของระบบไม่เสถียร และสามารถทำให้กลับมามีเสถียรภาพได้ โดยสร้างสมการคุณลักษณะแบบจำลองของมอเตอร์ขึ้นมาใหม่ด้วยวิธีการของ PIDA โดยที่ค่าโพลอยู่ที่ $0, -6.8550 \pm 12.7569j$ ซึ่งเป็นค่าเชิงซ้อนทั้งคู่ ดังภาพที่ 4.4



ภาพที่ 4.4 รูดโลกัสจากค่าพารามิเตอร์ทางพลศาสตร์ของระบบมอเตอร์

ดังนั้นเมื่อเลือกค่า R จะต้องให้มากกว่า $-6.8550 \pm 12.7569j$ และเลือกค่า R ของสมการ (4.7) ใหม่เป็น $R = 7$ ส่วนตำแหน่งที่อยู่ทางด้านซ้ายมือของรูดโลกัส ที่มีโพลมากที่สุดคือ $r = -30$ ซึ่งจะได้สมการคุณลักษณะใหม่ทำให้กลับมามีเสถียรภาพ จะได้สมการคุณลักษณะดังสมการที่ (4.8) ขั้นตอนที่ 5 เขียนสมการคุณลักษณะที่ได้ใหม่จะเป็นดังนี้

$$1+G(s)G_c(s) = (s+7)(s+30)(s+2.1 \pm 2j) \quad (4.8)$$



ภาพที่ 4.5 รูดโลกัส ที่ได้จากการแก้สมการคุณลักษณะตามวิธีการของ PIDA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการพอร์ทรีดโกลด์ ดังภาพที่ 4.5 จะเห็นได้ว่าครึ่งระนาบด้านซ้ายมือของ s-plane มีพื้นที่ในการควบคุมมากขึ้น และระบบยังคงเสถียร

ขั้นตอนที่ 6 พิจารณาสมการคุณลักษณะของสมการ (4.8) ใหม่จะได้เป็นดังนี้

$$s^4 + (13.7+K)s^3 + (209.73+K(a+b+z))s^2 + (K(a+b)z+Kab)s + Kabz = s^4 + 41.2s^3 + 373.81s^2 + 1193.17s + 1766.1 \quad (4.9)$$

ขั้นตอนที่ 7 จะได้สมการทั้งสี่ จากการเทียบสัมประสิทธิ์ของสมการดังนี้

$$13.71 + K = 41.2$$

$$K(a+b+z) + 209.73 = 373.81$$

$$K(a+b)z + Kab = 1193.17$$

$$Kabz = 1766.1$$

แก้สมการทั้งสี่สมการเพื่อหาค่า a, b, z และ K ของสมการ (4.9) โดยที่ค่า a, b, z หาได้จากโปรแกรม matlab ที่มีฟังก์ชัน root([]) จะได้ค่า a, b = $2.0902 \pm 5.6178j$, z = 1.788, K = 27.49 และสมการของตัวควบคุม $G_{c1}(s)$ เป็นดังนี้

$$G_{c1}(s) = 27.49 \frac{(s^2 + 4.1804s + 35.9286(s+1.788))}{s} \quad (4.10)$$

พิจารณาค่าตัวควบคุมใหม่เป็น $G_{c2}(s)$ โดยเปลี่ยนค่า R = 13 ทำตามขั้นตอนดังกล่าวจะได้สมการทั้งสี่สมการดังนี้

$$s^4 + (13.7+K)s^3 + (209.73+K(a+b+z))s^2 + (K(a+b)z+Kab)s + Kabz = s^4 + 47.2s^3 + 579s^2 + 1999.6s + 3279.9 \quad (4.11)$$

เทียบสมการทั้งสี่สมการจะได้เป็น

$$13.71 + K = 47.2$$

$$K(a+b+z) + 209.73 = 579$$

$$K(a+b)z + Kab = 1999.6$$

$$Kabz = 3279.9$$

แก้สมการทั้งสี่สมการเพื่อหาค่า a, b, z และ K โดยที่ค่า a, b, z หาได้จากโปรแกรม matlab และจะได้ค่า a, b = $0.5525 \pm 1.2514j$, z = 2.585, K = 33.49 ดังนั้นสมการของตัวควบคุม $G_{c2}(s)$ จะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_{c2}(s) = 33.49 \frac{(s^2 + 8.4412s + 37.8874)(s + 2.585)}{s} \quad (4.12)$$

ในการทำงานเดียวกันที่ตัวควบคุม $G_{c3}(s)$ เมื่อ $R = 19$ จะได้สมการคุณลักษณะจะได้เป็นดังนี้

$$\begin{aligned} s^4 + (13.7 + K)s^3 + (209.73 + K(a+b+z))s^2 + (K(a+b)z + Kab)s + Kabz \\ = s^4 + 53.2s^3 + 784.2s^2 + 2806.1s + 4793.7 \end{aligned} \quad (4.13)$$

เทียบสมการทั้งสี่สมการจะได้เป็น

$$13.71 + K = 53.2$$

$$K(a+b+z) + 209.73 = 784.2$$

$$K(a+b)z + Kab = 2806.1$$

$$Kabz = 4793.7$$

แก้สมการทั้งสี่สมการเพื่อหาค่า a, b, z และ K โดยที่ค่า a, b, z หาได้จากโปรแกรม matlab และจะได้ค่า $a, b = 4.0531 \pm 1.5553j$, $z = 6.441$, $K = 39.49$ ดังนั้นสมการของตัวควบคุม $G_{c3}(s)$ จะเป็น

$$G_{c3}(s) = 39.49 \frac{(s^2 + 8.1062 + 18.8466j)(s + 6.441)}{s} \quad (4.14)$$

นำสมการที่ (4.10), (4.12) และ (4.14) ทำให้อยู่ในรูปแบบของฟังก์ชันถ่ายโอนในวงปิดของ $T(s)$ ดังสมการที่ (4.3) จะได้สมการของฟังก์ชันถ่ายโอนในวงปิดของตัวควบคุม $T_1(s), T_2(s)$ และ $T_3(s)$ เป็นดังนี้

เมื่อกำหนดฟังก์ชันในวงปิดที่ $R = 7$, $K = 27.49$

$$T_1(s) = \frac{5765.4777s^3 + 34416.4413s^2 + 250230.3731s + 370413.486}{s^4 + 5779.1877s^3 + 34626.1713s^2 + 250230.3731s + 370413.486} \quad (4.15)$$

ที่ $R = 13$, $K = 33.49$

$$T_2(s) = \frac{7023.8577s^3 + 77446.3679s^2 + 419380.4914s + 687910.2893}{s^4 + 7037.5677s^3 + 77656.0979s^2 + 419380.4914s + 687910.2893} \quad (4.16)$$

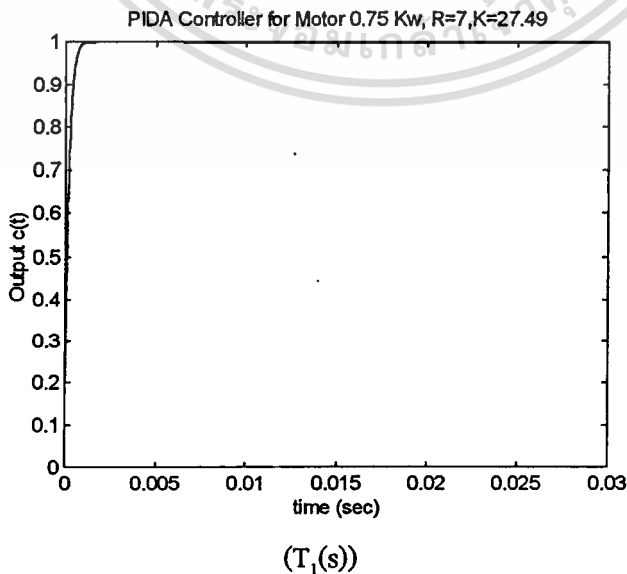
และที่ $R = 19$, $K = 39.49$

$$T_3(s) = \frac{8282.2377s^3 + 120483.3624s^2 + 588524.1919s + 1005389.098}{s^4 + 8295.9477s^3 + 120693.0924s^2 + 588524.1919s + 1005389.098} \quad (4.17)$$

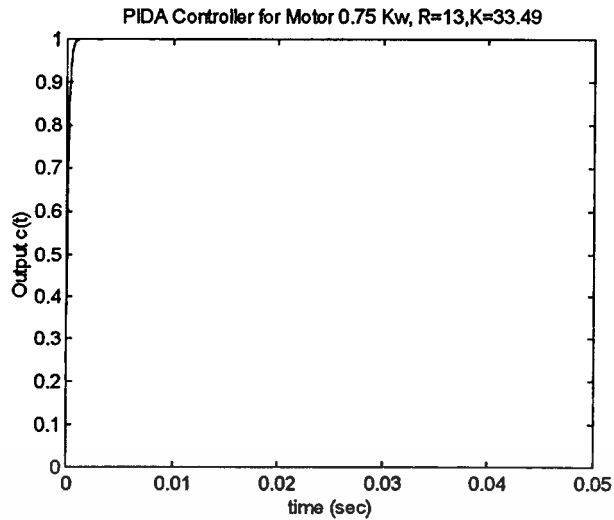
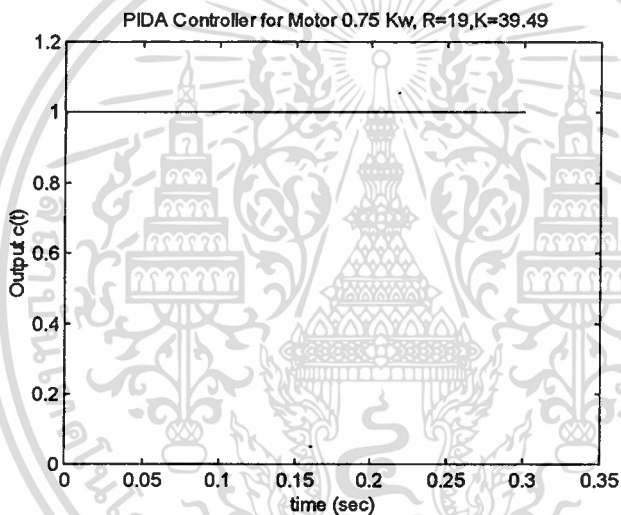
นำสมการ (4.15),(4.16) และ (4.17) ไปทำการทดสอบเพื่อดูผลตอบสนองของตัวควบคุมทั้งสาม ดังแสดงในภาพที่ 4.6 และบันทึกค่าต่างๆของการเปลี่ยนแปลงค่า R ลงในตารางที่ 4.3 จะเห็นได้ว่า ผลตอบสนองของตัวควบคุมทั้งหมดมีคุณสมบัติเป็นที่น่าพอใจ และตัวควบคุม $G_{c2}(s)$ จะให้ผลตอบสนองดีที่สุด นั่นก็หมายความว่าเมื่อเพิ่มค่า R หรือค่ารากให้ห่างจากจุด origin ของ s-plane จะทำให้มีผลตอบสนองในสภาวะคงตัวและค่าชั่วขณะที่เร็วขึ้น

ตารางที่ 4.3 แสดงผลของตัวควบคุมที่มีค่า R และ K ต่างกัน สำหรับการจำลองแบบ มอเตอร์กระแสตรงแบบแยกตัวกระตุ้น

controller	G_{c1}	G_{c2}	G_{c3}	Specification
R	7	13	19	--
K	27.49	33.49	39.49	--
P.O%	-0.138	-0.041	0.024	$\leq 5\%$
T_p (sec)	0.0022	0.0018	0.0018	≤ 1.0
T_s (sec)	0.0011	0.0001	0.00015	≤ 2.0
$\max \frac{C(t)}{D(t)}$	0.0002	0.0023	0.00024	0 or < 0.05



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(T₂(s))(T₃(s))

ภาพที่ 4.6 ผลตอบสนองของตัวควบคุม PIDA สำหรับมอเตอร์แบบแยกตัวกระตุ้น

4.4.1 (พิจารณาวงจร Zero Order Hold (ZOH))

วงจร Zero Order Hold เนื่องจากการควบคุมเป็นระบบดิจิทัล ดังนั้นจึงพิจารณาให้อุปกรณ์ระบบควบคุมมอเตอร์อยู่ในรูปของ Z-Transform จากการทดลองจะนำสมการของกระบวนการที่ (4.12) และสมการของตัวควบคุมที่ (4.16) มาทำการแปลงย้ายให้อยู่ในรูป ZOH และจากทฤษฎีการแปลงแซด (Z-Transform) กำหนดให้เวลาในการสุ่มเท่ากับ $T = 100$ มิลลิวินาที และ $s = \frac{2(z-1)}{T(z+1)}$ ซึ่งจะทำได้ง่ายต่อการสร้างระบบควบคุมสัญญาณที่ไม่ต่อเนื่อง ที่ยังคงมีเสถียรภาพ

เช่นเดียวกันกับระบบสัญญาณต่อเนื่อง โดยใช้การแปลงลาปลาซในโดเมนเวลา จาก s ไปเป็น z

เมื่อ $z = e^{st}$ คือการหาผลตอบสนองในโดเมนเวลาของระบบเชิงเลข ดังนั้นเขียนใหม่ได้เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$z = e^{sT} \approx \frac{1 - \frac{sT}{2}}{1 - \frac{sT}{2}} \quad (4.18)$$

ดังนั้นเมื่อออกแบบตัวควบคุมที่เป็นเชิงเลขโดยแปลงจากระบบสัญญาณที่ต่อเนื่อง กำหนดให้

$$G_c(z) = G_c(s) \quad (4.19)$$

และหาค่าเกณฑ์ของตัวควบคุมได้โดยการแปลงไบลิเนียร์ [7] หรือการอินทิเกรตหาพื้นที่ใต้สี่เหลี่ยมคางหมู ซึ่งสมการเอาท์พุทต่ออินพุทของวงจรถิงเลข จะเป็นดังนี้

$$\frac{y(z)}{u(z)} = \frac{2(z-1)}{T(z+1)} = \frac{2(1-z^{-1})}{T(1+z^{-1})} \quad (4.20)$$

เมื่อ z^{-1} เป็นดีเลย์ในโดเมนเวลา $z^{-1}u_k = u_{k-1}$ ดังนั้นสมการอินพุทของวงจรถิงเลข ZOH จะเป็น

$$u_k = u_{k-1} + \frac{T}{2}(y_k + y_{k-1}) \quad (4.21)$$

ซึ่งฟังก์ชันถ่ายโอนของระบบเวลาต่อเนื่องจะเป็นดังดังนี้

$$G_c(s) = \frac{\prod_{i=1}^m (s + t_i)}{\prod_{i=1}^n (s + s_i)} \quad (4.22)$$

โดยที่โพล และซีโร คือ $s = s_i$, $s = t_i$ ตามลำดับ และเทียบอันดับได้จาก $r = n - m > 0$ ดังนั้นเมื่อทำการแปลงไบลิเนียร์เป็น $G_c(z)$ จะได้ว่า

$$G_c(z) = \frac{\prod_{i=1}^m \left[\frac{2(z-1)}{T(z+1)} + t_i \right]}{\prod_{i=1}^n \left[\frac{2(z-1)}{T(z+1)} + s_i \right]} \quad (4.23)$$

ดังนั้นเขียนใหม่จะได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_c(z) = \left[\frac{2}{T} (z+1) \right]^r \frac{\prod_{i=1}^m [(z-1) + (z+1)t_i T/2]}{\prod_{i=1}^n [(z-1) + (z+1)s_i T/2]} \quad (4.25)$$

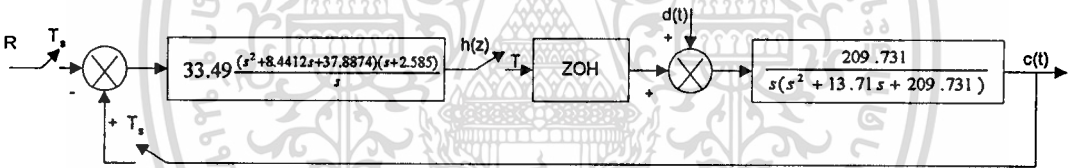
หรือ

$$G_c(z) = \left[\frac{2}{T} (z+1) \right]^r \frac{\prod_{i=1}^m [(1+t_i T/2)z - (1-t_i T/2)]}{\prod_{i=1}^n [(1+s_i T/2)z - (1-s_i T/2)]} \quad (4.26)$$

ดังนั้นเมื่อพิจารณาเสถียรภาพให้ดูตำแหน่งของโพลในสมการที่ (4.25) ซึ่งจะเห็นว่าระบบจะเสถียรภาพได้ก็ต่อเมื่อโพลของฟังก์ถ่ายโอน วางตัวอยู่ในวงกลมหนึ่งหน่วยของ z-plane โดยที่

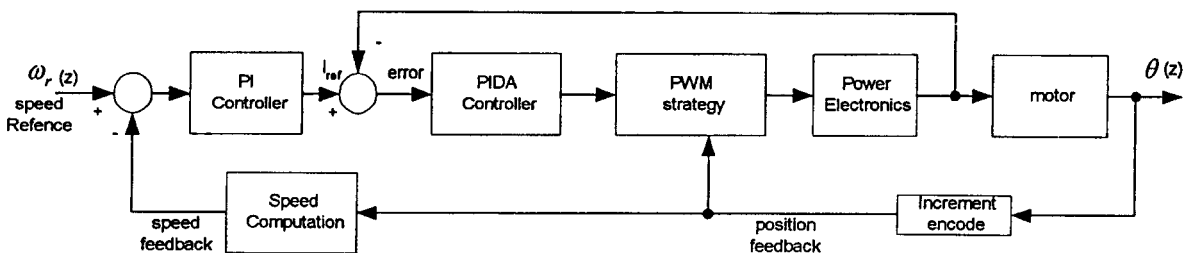
$$z = \frac{1+sT/2}{1-sT/2} \quad (4.27)$$

ดังนั้นเขียนไคอะแกรมการควบคุมจากภาพที่ 4.3 ให้เป็นระบบควบคุมแบบสัญญาณไม่ต่อเนื่องได้ดังภาพที่ 4.7



ภาพที่ 4.7 ไคอะแกรมของระบบควบคุมแบบสัญญาณไม่ต่อเนื่อง

และจากเอกสารที่ [3], [17] และ [18] สามารถเขียนเป็นไคอะแกรมของการควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นโดยรวม ที่มีกระแสไฟฟ้าป้อนให้กับตัวอินเวอร์เตอร์ และวงจรอิเล็กทรอนิกส์กำลัง คอยจ่ายกำลังให้กับมอเตอร์ ซึ่งเขียนได้ดังภาพที่ 4.8



ภาพที่ 4.8 ไคอะแกรมจำลองการควบคุมแบบป้อนไปข้างหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (4.15) และ (4.20) แปลงย้ายอยู่ในรูป z-transform จะได้ฟังก์ชันถ่ายโอนในวงปิดโดยการแปลงไปทีเรียด้วยโปรแกรม Matlab และใช้ฟังก์ชันที่เขียนขึ้นจาก [19] ดังนั้นเขียนได้เป็น

```
c=[687910.2893,419380.4914,77446.3679,7032.8577];
```

```
d=[687910.2893,419380.4914,77656.0979,7037.5677];
```

```
T=0.1; Fs=0.1/T; [b,a]=bilinear(c,d,Fs)
```

```
[C,B,A]=dir2cas(b,a)
```

```
%b = 0.9999 -2.4528 2.0017 -0.5411; a=1.0000 -2.4528 2.0016 -0.5412
```

```
%C = 0.9999;B=1.0000 -1.7772 0.8008;1.0000 -0.6757 0
```

```
%A =1.0000 -1.7761 0.7998;1.0000 -0.6766 0
```

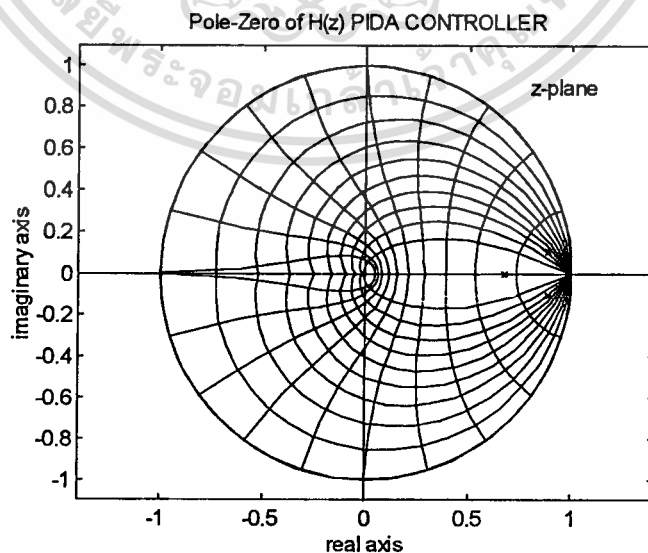
ดังนั้นจะได้ฟังก์ชันของ $H(z) = \frac{\omega_r(z)}{\theta(z)}$ เป็นดังนี้

$$H(z) = \frac{\omega_r(z)}{\theta(z)} = \frac{0.9999 - 2.4528z^{-1} + 2.0017z^{-2} - 0.5411z^{-3}}{1 - 2.4528z^{-1} + 2.0016z^{-2} - 0.5411z^{-3}} \quad (4.18)$$

และเนื่องจากฟังก์ชันถ่ายโอนเป็นระบบอันดับสามสามารถเขียน $H(z)$ ได้ใหม่เป็น

$$H(z) = \frac{0.9999(1 - 1.7772z^{-1} + 0.8009z^{-2})(1 - 0.6757z^{-1})}{(1 - 1.7761z^{-1} + 0.7998z^{-2})(1 - 0.6766z^{-1})} \quad (4.19)$$

ดังนั้นสามารถพิสูจน์เสถียรภาพของระบบได้ด้วยกราฟโพลด์ ซึ่งแสดงพฤติกรรมของระบบควบคุมเมื่อพิจารณาในระบบควบคุมในโดเมนของสัญญาณไม่ต่อเนื่อง สังเกตจะเห็นได้ว่าระบบยังคงมีเสถียรภาพ ดังภาพที่ 4.9



ภาพที่ 4.9 ผลการวิเคราะห์เสถียรภาพของระบบเวลาไม่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 บทสรุป

การออกแบบตัวควบคุม PID สำหรับกระบวนการอันดับสามนั้นมีความยุ่งยากและมีข้อกำหนดที่ตายตัว สามารถทำให้ปัญหาเหล่านั้นหมดไปได้โดยใช้ตัวควบคุม PIDA ที่มีข้อดีในการให้ผลตอบสนองที่เร็วสามารถวิเคราะห์และพิสูจน์ได้โดยการคำนวณหาค่าพารามิเตอร์ต่างๆที่สามารถควบคุมกระบวนการที่เป็นอันดับสาม เพื่อเป็นการพิสูจน์ในการนำเสนอวิธีการออกแบบตัวควบคุม PIDA ของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นนั้น ผลที่ได้รับจากการควบคุมจะใกล้เคียงกับผลการเขียนแบบมาก โดยจะเห็นได้ว่าการออกแบบตัวควบคุม PIDA ทำให้สามารถวิเคราะห์ง่ายเพียงแต่เพิ่มค่าเกณฑ์ ในช่วงที่มีการกำหนดไว้ โดยที่เกณฑ์ นี้จะเป็นเกณฑ์ ของตัวควบคุมที่นำไปพิจารณาค่ารากในวงรอบปิด ซึ่งเป็นส่วนที่สำคัญที่สุดในการออกแบบให้ระบบยังคงเสถียรภาพ ดังนั้นการออกแบบตัวควบคุม PIDA จึงเป็นอีกวิธีหนึ่งของผู้ที่ต้องการเริ่มออกแบบและวิเคราะห์ระบบควบคุมที่มีกระบวนการเป็นอันดับสามหรืออันดับที่มากกว่านั้น สามารถนำไปพัฒนาควบคุมกระบวนการอื่นๆ ได้ง่ายตามต้องการต่อไป



บทที่ 5

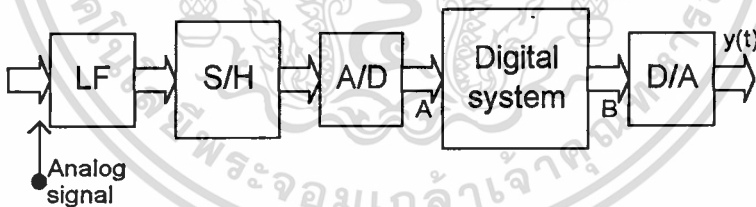
การออกแบบระบบควบคุมดิจิทัลด้วยตัวประมวลผล สัญญาณเชิงเลข

5.1 นิยามของการประมวลผลเชิงเลข

การประมวลผลเชิงเลขคือ การประมวลผลสัญญาณที่มีแอมป์พลิจูดของสัญญาณมีค่าเฉพาะเซตหนึ่งเท่านั้น เช่น การประมวลผลสัญญาณที่ออกมาจากวงจร A/D ซึ่งจะมีสัญญาณเพียงสองค่า คือ 0 โวลต์ กับ 5 โวลต์ เท่านั้น ซึ่งการประมวลผลสัญญาณชนิดนี้นั้นจะใช้อุปกรณ์เฉพาะอย่างในการประมวล อย่างเช่น ไมโครโปรเซสเซอร์ เป็นต้น

5.1.1 (ทฤษฎีการสุ่มตัวอย่าง)

อาจกล่าวได้ว่า สัญญาณในธรรมชาติส่วนมาก เช่นสัญญาณเสียง สัญญาณต้นสะเทือนของพื้นโลก คลื่นหัวใจ หรือ การแปรค่าไปของอุณหภูมิ เหล่านี้เป็นไปในลักษณะแบบต่อเนื่องกับพิสัยเวลา หรือ กล่าวได้ว่าเป็นสัญญาณเชิงอุปมาน การนำสัญญาณเหล่านี้ไปประมวลผลในลักษณะการประมวลผลสัญญาณเชิงเลข หรือ การประมวลผลสัญญาณเชิงเต็มหน่วย (discrete signal processing) ได้ ต้องใช้ระบบการประมวลผลดังภาพที่ 5.1



ภาพที่ 5.1 ระบบประมวลผลสัญญาณเชิงอุปมานโดยใช้ระบบประมวลผลสัญญาณเชิงเลข

วงจร S/H เป็นวงจรสุ่มและคงค่าสัญญาณไว้ เพื่อให้วงจร A/D ทำการแปลงเป็นตัวเลขอีกทีหนึ่ง ดังเหตุว่า ระบบประมวลผลสัญญาณเชิงเลข เป็นส่วนของระบบที่อยู่ระหว่าง A และ B โดยที่สัญญาณที่จุดนี้เป็นสัญญาณที่ถูกแปลงเป็นตัวเลขแล้ว อย่างไรก็ตามการเปลี่ยนแปลงสัญญาณเชิงอุปมานมาเป็นสัญญาณเชิงเลขนั้นในเบื้องต้นนั้นต้องมีการสุ่มตัวอย่างก่อน ซึ่งความถี่ในการสุ่มตัวอย่างโดยไม่ทำให้สัญญาณสูญเสียข้อมูลไปนั้น ทฤษฎีการสุ่มตัวอย่าง (sampling theory) ของแชนนอน (Shannon) กล่าวไว้ว่า “ ถ้าหากมีสัญญาณเชิงอุปมาน $x(t)$ ที่ค่า การแปลงฟูริเยอร์ หรือ สเปกตรัมกำลัง (power spectrum) ของมันมี แถบความถี่ปฏิบัติ เท่ากับ f_c หรือ $f_s \geq 2f_c$ โดยทั่วไป

เราอาจสุ่มตัวอย่างด้วยค่าความถี่ $f_{SN} = 2 f_0$ พอดี ค่าความถี่นี้มีชื่อเรียกว่าความถี่ไนควิสต์ (Nyquist frequency) และ คาบเวลา $T_0 = 1/(2f_0)$ นี้เรียกว่า ช่วงเวลาสุ่มตัวอย่างไนควิสต์ (Nyquist interval) ในทางปฏิบัติเพื่อหลีกเลี่ยงผลของ ปรากฏการณ์ไม่เป็นเชิงเส้นที่อาจเกิดจากการสุ่มตัวอย่าง ซึ่งมักใช้ความถี่ในการสุ่มตัวอย่าง f_s มากกว่าค่า ความถี่ไนควิสต์ หรือ f_{SN} ขึ้นไป ส่วนจะมีค่ามากกว่าเท่าใดนั้นขึ้นอยู่กับลักษณะงาน ไม่ได้มีการกำหนดค่าที่แน่นอน อาจใช้เป็น $2.5f_0$ หรือ $5f_0$ หรือ $10f_0$ ก็ได้ ส่วนรายละเอียดของความไม่เป็นเชิงเส้นนั้นไม่ขอกล่าวไว้ในที่นี้

ตามทฤษฎีการสุ่มตัวอย่างของ แซมพลอน นั้นเห็นได้ว่าการสุ่มตัวอย่างสัญญาณได้ถูกต้องก็ต่อเมื่อรู้ค่า แถบความถี่ปฏิบัติงานของสัญญาณ หรือ พูดอีกนัยหนึ่งว่า สัญญาณต้องมีแถบความถี่ปฏิบัติงาน จึงจะทำการสุ่มตัวอย่างได้ ดังนั้น ในบางครั้งเพื่อให้มั่นใจว่าสัญญาณที่ทำการประมวลผล ถูกประมวลอย่างถูกต้อง ในขั้นต้นของระบบประมวลผลเชิงเลขเต็มหน่วย และเชิงเลข จึงอาจมีวงจรกรองความถี่ต่ำผ่าน (low pass filter) ไว้เป็นตัวกำหนดแถบความถี่ปฏิบัติงานของสัญญาณ

5.1.2 (บทนิยามของการแปลงแซด (Z-Transform))

ถ้าหากเรามีลำดับสัญญาณเข้าในโดเมนเวลา (time domain) หรือ $x(0), x(1), \dots, x(n)$ ซึ่งสามารถเขียนแทนลำดับนี้ด้วยสมการ

$$X(n) = \sum x(m)\delta(n-m) \quad (5.1)$$

และถ้าหากเราทำการแปลงลาปลาซ ซึ่งจะเขียนด้วย ตัวดำเนินการ $L\{\}$ ของสมการจะได้ว่า

$$LX(nT) = L\{\sum x(mT)\delta((n-m)T)\}$$

เขียนใหม่ได้ว่า

$$LX(nT) = \sum x(mT)L\{\delta((n-m)T)\} \quad (5.2)$$

โดยที่ $L\delta(n-m)T = \exp\{-mST\}$ ดังนั้น นิยามของการแปลงแซด คือ

$$X(Z) = L\{x(nT)\} = \sum x(nT)\exp(-mST)$$

$$X(Z) = \sum x(mT)Z^{-m} \quad (5.3)$$

เมื่อ Z เป็นจำนวนเชิงซ้อน หรือ $Z = \exp(-ST)$ และโดยทั่วไปจะให้ $T=1$ เนื่องจากดัชนี m ของการแปลงแซดในสมการข้างบน แปรค่าจาก $-\infty$ ถึง ∞ ดังนั้นการแปลงแซด นี้จึงมีชื่อเรียกอีกอย่างว่า ผลการแปลง Z สองข้าง (two-side Z -transform) แต่ในทางปฏิบัติรามักจะมีลำดับ $X(m)$ โดยที่ m มีค่ามากกว่าศูนย์ เพราะฉะนั้น จึงเรียกว่า ผลการแปลงแซด ข้างเดียว (one-side Z -transform) ซึ่งนิยามได้ดังนี้

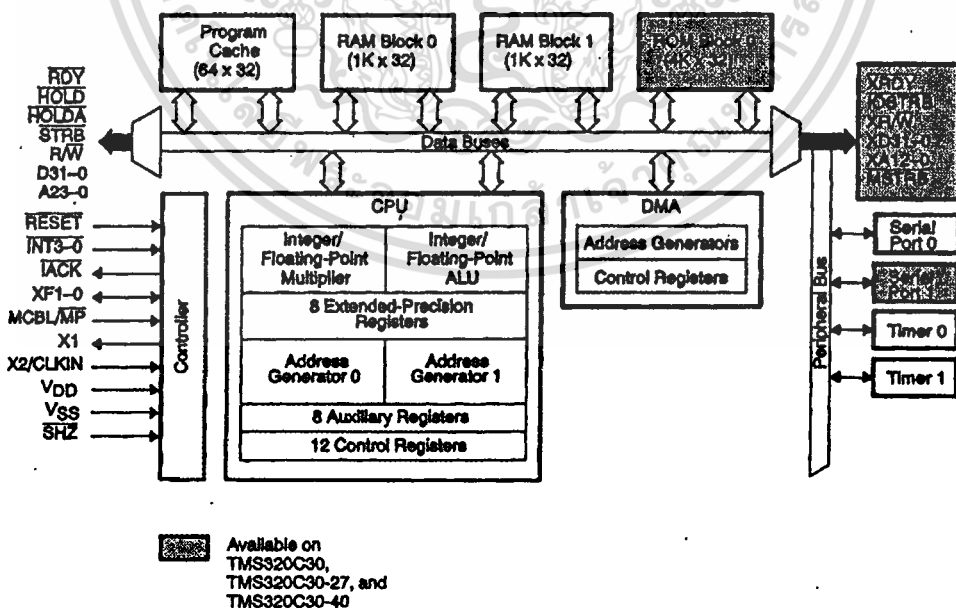
$$X(Z) = \sum x(m)Z^{-m} \quad (5.4)$$

จะเห็นได้ชัดเจนจากสมการ (5.2) และ (5.3) ว่าแท้จริงแล้ว การแปลงแซดก็คือ การประยุกต์ใช้การแปลงลาปลาซ กับ สัญญาณที่เป็นลำดับ นั่นเอง

5.2 สถาปัตยกรรมของหน่วยประมวลผลสัญญาณเชิงเลข TMS320C31

ไอซีเบอร์ TMS320C31 ถูกสร้างและออกแบบมาเพื่อตอบสนองความต้องการทางด้านการประมวลผลบนพื้นฐานของคณิตศาสตร์ขั้นสูงและการแก้ปัญหาทาง hardware และ software

ประสิทธิภาพอันสูงของ TMS320C31 เกิดจากความแน่นอนและความกว้างทางไดนามิกของส่วนคำนวณทางทศนิยม มีหน่วยความจำขนาดใหญ่ การทำงานแบบขนานและมี DMA Controller เพื่อการทำงานที่มีประสิทธิภาพมากขึ้น



ภาพที่ 5.2.1 แสดงสถาปัตยกรรมของไอซีเบอร์ TMS320C3X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรม TMS320C31 ประกอบด้วยส่วนต่าง ๆ ที่สำคัญดังนี้

5.2.1 (หน่วยประมวลผลกลาง Central Processing Unit)

ซึ่งหน่วยประมวลผลกลางจะประกอบไปด้วย

5.2.1.1 (หน่วยการคูณ Multiplier)

มีหน้าที่ในการคูณ และสามารถคูณจำนวนเต็ม 24 bit และจำนวนทศนิยม 32 bit โดยใช้เวลาเพียง 1 รอบ ซึ่งการคำนวณจำนวนทศนิยมจะใช้เวลา 50 ns ต่อรอบ และสามารถประมวลผลแบบขนานได้เพื่อที่จะเพิ่มความสามารถในการประมวลผลข้อมูล โดยใช้คำสั่งแบบขนานให้ทำการคูณและการทำงานของ ALU ภายในหนึ่งรอบ เมื่อหน่วยการคูณทำการคำนวณทศนิยม อินพุตที่ใช้จะมีขนาด 32 บิต และผลลัพธ์ที่ได้จะมีขนาด 40 บิต แต่ถ้า multiplier ทำการคูณจำนวนเต็มอินพุตจะมีขนาดเพียง 24 บิต และจะได้ผลลัพธ์ขนาด 32 บิต

5.2.1.2(หน่วยคำนวณทางคณิตศาสตร์ Arithmetic Logic Unit)

ใน 1 รอบการทำงานของ ALU จะประมวลผลจำนวนเต็มได้ 32 บิต และทศนิยมได้ 40 บิต โดยผลลัพธ์จาก ALU ยังคงมีขนาด 32 บิตถ้าเป็นจำนวนเต็ม และ 40 บิตถ้าเป็นทศนิยม barrel shifter จะมีหน้าที่ในการเลื่อนข้อมูล 32 บิต ทั้ง ซ้ายและขวาใน 1 รอบของการทำงาน บัสภายใน (CPU1/CPU2 และ REG1/REG2) จะเป็นตัวพาข้อมูลที่จะนำมาประมวลผล 2 ตัวจากหน่วยความจำ และอีก 2 ตัวจากรีจิสเตอร์ ดังนั้นจึงสามารถทำการคูณ บวก หรือลบแบบขนานได้ ใน 1 รอบการทำงาน

5.2.1.3(หน่วยช่วยคำนวณทางคณิตศาสตร์ Auxilliary Register Arithmetic Unit)

ARAUS เป็นรีจิสเตอร์ที่ช่วยในการทำงานแบบขนานกับตัวคูณและ ALU โดย ARAUs จะมี 2 ตัวคือ ARAU0 และ ARAU1 ARAU จะสามารถเก็บแอดเดรสได้ 2 แอดเดรสใน 1 รอบการทำงานเพื่อใช้ในการอ้างตำแหน่งแบบต่าง ๆ

5.2.1.3 (รีจิสเตอร์ของ CPU Register File)

ใน TMS320C31 จะมีรีจิสเตอร์อยู่ 28 ตัว ซึ่งตรงกับ CPU โดย รีจิสเตอร์ทั้งหลายเหล่านี้ จะทำงานกับ multiplier และ ALU และยังสามารถใช้เพื่อวัตถุประสงค์อื่นอีกด้วย อย่างไรก็ตามรีจิสเตอร์เหล่านี้ก็ยังมีหน้าที่พิเศษบางอย่างด้วย เช่น ใช้สำหรับการทำงานเกี่ยวกับจำนวนเลขทศนิยม

และรีจิสเตอร์ช่วย (Auxiliary register) ทั้ง 8 ตัวยังใช้ในการอ้างแอดเดรสแบบ indirect addressing register ด้วย รีจิสเตอร์ที่เหลือจะทำหน้าที่เกี่ยวกับระบบต่างๆ เช่น การอ้างแอดเดรส การจัดการเกี่ยวกับสแตค (stack) การบอกสถานะของโปรเซสเซอร์ การอินเตอร์รัพต์ และการเคลื่อนย้ายข้อมูลเป็นบล็อก

ชื่อต่างๆของรีจิสเตอร์และหน้าที่การทำงานแสดงดังตาราง 2.1

-exten-precision register (R7-R0) :สามารถใช้เก็บค่าจำนวนเต็ม 32 บิต และทศนิยม 40 บิต โดยจะเก็บค่าลงในบิตที่ 0-39 แต่ถ้าจำนวนเต็มแบบมีเครื่องหมายก็จะทำการเก็บที่ บิต 0-31 ส่วนบิตที่ 32-39 จะไม่ใช้

-auxilairy register (AR7-AR0) : จะถูกใช้โดย CPU และถูกควบคุมโดย ARAU ทั้ง 2 หน้าแรกที่แรกของ AR ยังใช้ให้ส่วนที่เก็บตัวนับในการวนลูปหรือจะใช้เพื่องานอื่นๆ ที่เกี่ยวกับตัวมัลติเพล็กซ์เซอร์และ ALU

-data page pointer (DP) : ส่วนรีจิสเตอร์ขนาด 32 บิต โดยที่ 8 บิตแรกจะถูกใช้ในการอ้างแอดเดรสแบบ direct addressing mode ซึ่งใช้เป็นตัวชี้ (pointer) ไปยังหน้า (page) ต่างๆ ของข้อมูล

-index register (IR0,IR1) : จะถูกใช้โดย ARAU ในการเก็บค่าของดัชนีในการชี้แอดเดรส (indexed address)

-block size register (BK) : มีขนาด 32 บิตซึ่ง ARAU จะใช้ในการอ้างแอดเดรสแบบเก็บ circular addressing เพื่อเป็นตัวเก็บค่าของขนาดของบล็อกข้อมูล

-system stack pointer (SP) : เป็นรีจิสเตอร์ขนาด 32 บิต ซึ่งเป็นค่าของแอดเดรสของยอดของสแตค โดยปกติ SP จะชี้ไปยังค่าสุดท้ายซึ่งถูกเก็บอยู่บนสแตค การ PUSH จะกระทำ ก่อนที่จะเพิ่มค่าของ SP และการ POP หลังการลดค่าของ SP คำสั่งที่จะทำงานกับ SP ได้คือ interrupts, traps, returns, PUSH และ POP

-status register (ST) : เป็นรีจิสเตอร์ที่เก็บข้อมูลต่างๆ ที่แสดงสถานะของ CPU เช่นค่าของ flag ซึ่งจะแสดงสถานะของผลลัพธ์ที่ได้จากการคำนวณทางคณิตศาสตร์ ว่า เป็น ศูนย์ ลบ ฯลฯ รวมถึงการคำนวณทางลอจิกด้วย

-CPU/DMA interrupt enable register (IE) : เป็นรีจิสเตอร์ขนาด 32 บิต ใช้สำหรับการกำหนดการเ็นนาเบิ้ลอินเตอร์รัพต์ของ CPU โดยการเ็นนาเบิ้ลของ CPU จะใช้บิตที่ 0-10 และการเ็นนาเบิ้ลของ DMA จะใช้บิตที่ 16-26 “1” จะเป็นการเ็นนาเบิ้ล และ “0” จะเป็นการคีสเ็นเนิ้ล

-I/O Flage register (IOF) : เป็นรีจิสเตอร์ในการควบคุมขา XF0 และ XF1

- repeat counter (RC) : เป็นรีจิสเตอร์ขนาด 32 บิตใช้เก็บค่าของจำนวนทำหรือจำนวนครั้งของบล็อกข้อมูลซึ่งถูกกระทำซ้ำเมื่อโปรเซสเซอร์ทำงานในโหมดของการกระทำในข้อมูลซ้ำ (repeat mode) repeat star address register (RS) รีจิสเตอร์ขนาด 32 บิต ถูกใช้ในการเก็บแอดเดรสเริ่มต้นของบล็อกข้อมูลที่จะถูกกระทำซ้ำ และ repeat end address register (RE) : ซึ่งเป็นรีจิสเตอร์ขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32 บิต จะถูกเก็บแอดเดรสสุดท้ายของบล็อกข้อมูลซึ่งจะถูกกระทำซ้ำ

-program counter (PC) : เป็นรีจิสเตอร์ขนาด 32 บิต ซึ่งเก็บแอดเดรสของคำสั่งที่จะถูกนำมาอ่าน คำสั่งถัดไป

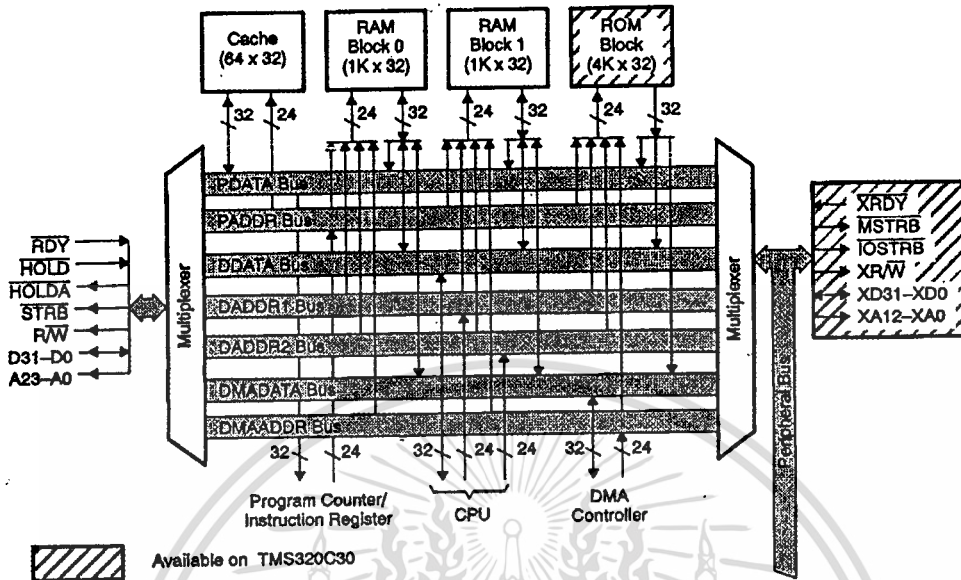
ตารางที่ 5.2.1 ชื่อและหน้าที่ต่างๆของรีจิสเตอร์ใน CPU

Register Name	Assigned Function
R0	Extended-precision register 0
R1	Extended-precision register 1
R2	Extended-precision register 2
R3	Extended-precision register 3
R4	Extended-precision register 4
R5	Extended-precision register 5
R6	Extended-precision register 6
R7	Extended-precision register 7
AR0	Auxiliary register 0
AR1	Auxiliary register 1
AR2	Auxiliary register 2
AR3	Auxiliary register 3
AR4	Auxiliary register 4
AR5	Auxiliary register 5
AR6	Auxiliary register 6
AR7	Auxiliary register 7
DP	Data-page pointer
IR0	Index register 0
IR1	Index register 1
BK	Block-size register
SP	System-stack pointer
ST	Status register
IE	CPU/DMA interrupt-enable
IF	CPU interrupt flag
IOF	I/O flag
RS	Repeat start-address
RE	Repeat end-address
RC	Repeat counter
PC	Program counter

5.2.2 (การจัดหน่วยความจำ Memory Organization)

หน่วยความจำของ TMS320C31 จะมีขนาด 16M และมีขนาดของ 1 word เท่ากับ 32บิต โดยในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำนี้จะใช้กับโปรแกรมข้อมูล และการอินพุต เอาท์พุต ดังนั้นค่าของสัมประสิทธิ์โปรแกรมหรือข้อมูล จะถูกเก็บได้ทั้ง RAM และ ROM



ภาพที่ 5.2.2 แสดงหน่วยความจำของไอซีเบอร์ TMS320C31

5.2.2.1 (หน่วยความจำแบบ RAM และ ROM)

ภาพที่ 5.2.2 จะแสดงการจัดหน่วยความจำภายใน TMS320C31 โดยแรมจะแบ่งออกเป็น บล็อก 0 และ 1 ซึ่งจะมีขนาดบล็อกละ 1K x 32 และ ROM จะมีขนาด 4K x 32 ทั้ง RAM และ ROM สามารถถูกเข้าใช้ได้โดย CPU 2 ครั้งใน 1 รอบของการทำงานได้ ดังนั้นการที่มีบัสของโปรแกรมแยกกัน และบัสของ DMA แยกกัน ทำให้การอ่านโปรแกรม การอ่านและการเขียนข้อมูลหรือการทำงานของ DMA ที่สามารถทำงานแบบขนานได้

5.2.2.2 (ตารางหน่วยความจำ Memory Maps)

ตารางหน่วยความจำจะขึ้นอยู่กับว่าจะให้โปรเซสเซอร์ประมวลผลในโหมดของไมโครโปรเซสเซอร์ หรือ ไมโครคอมพิวเตอร์ แต่ในโครงการนี้จะให้โปรเซสเซอร์ประมวลผลในโหมดของไมโครโปรเซสเซอร์ แสดงดังภาพที่ 5.2.3 ในแอดเดรสที่ 800000h จนถึง 801FFFh จะถูกใช้สำหรับ expansion bus ซึ่งจะถูกเข้าถึงได้โดยให้สัญญาณ / MSTRB แอดดีฟ แอดเดรสที่ 80200h จนถึง 803FFFh จะไม่ใช่แอดเดรสที่ 80400h ถึง 805FFFh จะถูกใช้สำหรับ expansion bus ซึ่งจะถูกทำให้เข้าถึงโดยให้สัญญาณ /IOSTRB แอดดีฟ แอดเดรส 80600h ถึง 807FFFh จะไม่ใช่ทุกๆ รีจิสเตอร์จะอยู่ที่แอดเดรส 80800h ถึง 8097FFFh RAM บล็อก 0 จะอยู่ที่ 809800h ถึง 809BFFFh และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.Parallel addressing mode :

- Register : ข้อมูลจะอยู่ใน extended-precision register
- Indirect

4.Long-immediate addressing mode :

- Long-immediate : ข้อมูลจะเป็นค่าที่โหลดโดยตรงขนาด 24 บิต จากคำสั่ง

5.Conditional branch addressing mode :

- Register
- PC-relative : คำที่มีเครื่องหมายขนาด 16 บิต จะถูกบวกกับค่าใน PC

5.2.3 (การติดต่อบัสภายใน Internal Bus Operation)

ข้อดีที่สำคัญประการหนึ่งของ TMS320C31 คือการที่มีบัสภายในและสามารถประมวลผลแบบขนานได้บัสโปรแกรมที่แยกกัน (PADDR และ PDATA) บัสข้อมูลที่แยกกัน (DADDR1,DADDR2และDDATA) และบัส DMA ที่แยกกัน (DMAADDR และ DMADATA) ทำให้สามารถทำการประมวลผลแบบขนานได้ บัสเหล่านี้สามารถต่อได้กับหน่วยความจำภายใน หน่วยความจำภายนอก หรือ อุปกรณ์ภายนอกอื่นๆ รูปที่ 5.2.2 จะแสดงการต่อบัสต่างๆ เหล่านี้

PC จะถูกต่ออยู่กับ PADDR (ขนาด 24 บิต) IR จะต่ออยู่กับ PDATA (ขนาด 32 บิต) ซึ่งบัสเหล่านี้สามารถอ่านคำสั่ง ทุกๆรอบของการทำงาน

DADDR1 และ DADDR2 (ขนาด 24 บิต) และ DDATA (ขนาด 32 บิต) จะใช้ในการติดต่อกับหน่วยความจำในหลายๆรอบของการทำงาน DDATA จะใช้ในการส่งข้อมูลไปยัง CPU โดยผ่านบัส CPU1 และ CPU2 บัส CPU1 และ CPU2 สามารถส่งข้อมูลจากหน่วยความจำ 2 ข้อมูลไปยัง Multiplier ALU และ รีจิสเตอร์ในหลายๆ รอบของการทำงาน นอกจากนี้ภายใน CPU มีบัสรีจิสเตอร์ REG1 และ REG2 ซึ่งใช้ในการหาข้อมูล 2 ค่า จากรีจิสเตอร์ไปยัง Multiplier และ ALU ในทุกๆรอบการทำงาน ภาพที่ 2.2 แสดงบัสภายในต่างๆ DMA Controller จะใช้ DMAADDR (ขนาด 24 บิต)และ DMADATA (ขนาด 32 บิต) โดยบัสเหล่านี้ทำให้ DMA สามารถติดต่อกับหน่วยความจำแบบขนานได้

5.2.4 (การติดต่อบัสภายนอก External Bus Operation)

หน่วยความจำและอุปกรณ์ภายนอกสามารถอินเทอร์เฟสกับ TMS320C31 ได้ 2 ทางคือ ทาง Primary bus expansion bus การอินเทอร์เฟสกับหน่วยความจำหรืออุปกรณ์ภายนอกที่มีความเร็วช้ากว่า TMS320C31 จะต้องเพิ่ม Wait state เข้าไป โดยอาศัยการควบคุมของ memory mapped control register และสัญญาณจากภายนอก

primary bus มีขนาดของ data bus 32 บิต, address bus 24 บิต และ control signal อีก 1 ชุด expansion bus มีขนาด data bus 32 บิต, address bus 13 บิต และ control signal อีก 1 ชุด โดยบัสทั้งสอง สามารถถูกควบคุมได้โดยการใช้โปรแกรมสั่งงานและสัญญาณควบคุมจากภายนอก

Register	Peripheral Address
Expansion-Bus Control (see subsection 7.1.2)†	808080h
Reserved	808081h
Reserved	808082h
Reserved	808083h
Primary-Bus Control (see subsection 7.1.1)	808084h
Reserved	808085h
Reserved	808086h
Reserved	808087h
Reserved	808088h
Reserved	808089h
Reserved	80808Ah
Reserved	80808Bh
Reserved	80808Ch
Reserved	80808Dh
Reserved	80808Eh
Reserved	80808Fh

† Reserved on the TMS320C31

ภาพที่ 5.2.4 การควบคุมรีจิสเตอร์ที่ติดต่อกับ memory-mapped ภายนอก

อุปกรณ์ภายนอกสามารถเข้าถึงได้โดยการใช้สัญญาณ /STRB/MSTRB และ /IOSTRB โดยสัญญาณ /STRB จะใช้เมื่อต้องการ primary bus ส่วน expansion bus จะสามารถใช้งานได้โดย 2 แบบ คือ

- การเข้าใช้ memory สามารถทำได้โดยให้ /MSTRB เป็น 0 โดย /MSTRB จะมีวงรอบการทำงานเหมือนกับ /STRB

- อุปกรณ์ภายนอกสามารถเข้าใช้ได้โดยให้สัญญาณ /IOSTRB เป็น 0 primary bus และ expansion bus จะมี control register ดังภาพที่ 5.2.5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
xx	xx	xx	BNKCMP				WTCNT			SWW		HIZ	NOHOLD	HOLDST	
R/W			R/W			R/W			R/W		R/W		R/W		R

ภาพที่ 5.2.5 บัสควบคุมรีจิสเตอร์ปฐมภูมิ primary bus control register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

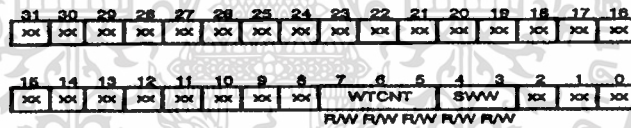
ตารางที่ 5.2.2 แสดงหน้าที่ต่าง ๆ ของบัสควบคุมรีจิสเตอร์ปฐมภูมิ

Bit	Name	Reset Value	Function
0	HOLDST	x†	Hold status bit. This bit signals whether the port is being held (HOLDST = 1) or is not being held (HOLDST = 0). This status bit is valid whether the port has been held via hardware or software.
1	NOHOLD	0	Port hold signal. NOHOLD allows or disallows the port to be held by an external HOLD signal. When NOHOLD = 1, the TMS320C3x takes over the external bus and controls it, regardless of serviced or pending requests by external devices. No hold acknowledge (HOLDA) is asserted when a HOLD is received. However, it is asserted if an internal hold is generated (HIZ = 1). NOHOLD is set to 0 at reset.
2	HIZ	0	Internal hold. When set (HIZ = 1), the port is put in hold mode. This is equivalent to the external HOLD signal. By forcing a high-impedance condition, the TMS320C3x can relinquish the external memory port through software. HOLDA goes low when the port is placed in the high-impedance state. HIZ is set to 0 at reset.
4-3	SWW	11	Software wait mode. In conjunction with WTCNT, this two-bit field defines the mode of wait-state generation. It is set to 1 1 at reset.
7-5	WTCNT	111	Software wait mode. This three-bit field specifies the number of cycles to use when in software wait mode for the generation of internal wait states. The range is 0 (WTCNT = 0 0 0) to 7 (WTCNT = 1 1 1) H1/H3 cycles. It is set to 1 1 1 at reset.
12-8	BNKCMP	10000	Bank compare. This five-bit field specifies the number of MSBs of the address to be used to define the bank size. It is set to 1 0 0 0 0 at reset.
31-13	Reserved	0-0	Read as 0.

†x = 0 or 1

control register ของ primary bus จะมีขนาด 32 บิต ดังภาพที่ 5.2.5 และตาราง 5.2.2

control register ของ expansion bus จะมีขนาด 32 บิต ดังภาพที่ 5.2.6 และตาราง 5.2.3



NOTE: xxx = reserved bit, read as 0.
R = read, W = write.

ภาพที่ 5.2.6 บัสควบคุมขยายรีจิสเตอร์ expansion bus control register

ตารางที่ 5.2.3 แสดงหน้าที่ต่าง ๆ ของบัสควบคุมขยายรีจิสเตอร์

Bit	Name	Reset Value	Function
2-0	Reserved	000	Read as 0.
4-3	SWW	11	Software wait-state generation. In conjunction with the WTCNT, this two-bit field defines the mode of wait-state generation. It is set to 1 1 at reset.
7-5	WTCNT	111	Software wait mode. This three-bit field specifies the number of cycles to use when in software wait mode for the generation of internal wait states. The range is 0 (WTCNT = 0 0 0) to 7 (WTCNT = 1 1 1) H1/H3 clock cycles. It is set to 1 1 1 at reset.
31-8	Reserved	0-0	Read as 0.

ในทุกๆ รอบของสัญญาณบนสายบัสจะขึ้นอยู่กับสัญญาณนาฬิกา H1 โดย 1 รอบของ H1 เริ่มนับจากขอบขาถลงจนถึงขอบขาถลงของลูกถัดไป สำหรับการทำงานแบบความเร็วเต็มที่(คือ ไม่ใช่ wait state) ในการเขียนข้อมูลจะใช้เวลา 2 รอบของ H1 และการอ่านข้อมูลจะใช้ 1 รอบของ H1 แต่ถ้าการไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ่านข้อมูลกระทำต่อจากคำสั่งการเขียนข้อมูล การอ่านข้อมูลจะใช้เวลา 2 รอบของ H1 ซึ่งหลักการนี้ จะใช้ได้ทั้งกับ primary bus และ expansion bus (เมื่อใช้ /MSTRB)

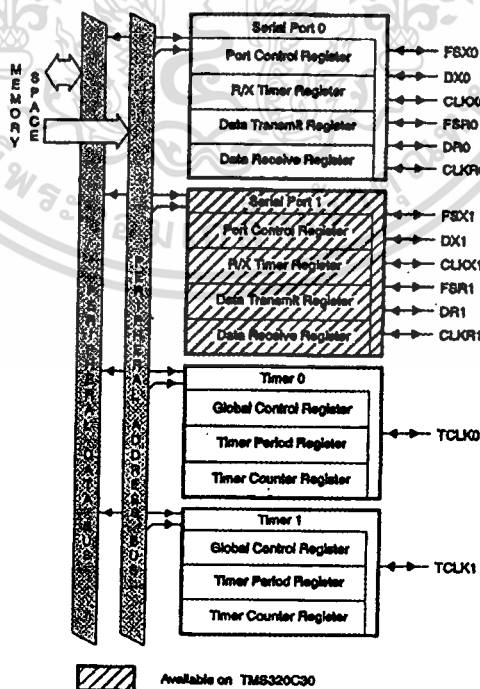
ถ้าการเขียนข้อมูลเป็นการกระทำภายใน โดยไม่มีการอินเตอร์เฟสกับอุปกรณ์ภายนอก(คือการใช้จาก CPU และ DMA) วงรอบของการเขียนข้อมูลจะใช้เพียง 1 รอบของ H1 เท่านั้น

สัญญาณ /MSTRB และ /STRB จะแอกตีฟที่สถานะ 0 สำหรับการอ่านและการเขียนข้อมูลใน ขณะที่สภาวะก่อนหลังของการแอกตีฟของ /MSTRB และ /STRB ในการเขียนข้อมูลจะมี transition cycle ของ H1 ในระหว่าง transition cycle จะเกิด

- 1) /MSTRB และ /STRB จะเป็นสถานะ high
- 2) (X)R/W จะเปลี่ยนสถานะที่ขอบขาขึ้นของ H1
- 3) ถ้าเป็นสถานะของการเขียนข้อมูล address จะเปลี่ยนสถานะที่ขอบขาขึ้นของ H1 และถ้าเป็นสถานะของการอ่านข้อมูล address จะเปลี่ยนที่ขอบขาลงของ H1

5.2.5 (ระบบติดต่อภายนอก Peripherals)

อุปกรณ์ภายนอกที่ติดต่อกับ TMS320C31 จะควบคุมโดยผ่านหน่วยความจำในตำแหน่ง 8097FFh ถึง 808000h บัสที่ติดต่อกับอุปกรณ์ภายนอกหรือ peripheral bus จะมีบัสข้อมูลขนาด 32 บิต และบัสแอกเดรสขนาด 24 บิต การติดต่อกับอุปกรณ์ภายนอกจะผ่านไทม์เมอร์ 2 ตัว และพอร์ตอนุกรม 2 ตัว ภาพที่ 5.2.7 จะแสดงบัสและขาสัญญาณต่างๆ ที่ใช้ในการติดต่อกับอุปกรณ์ภายนอก



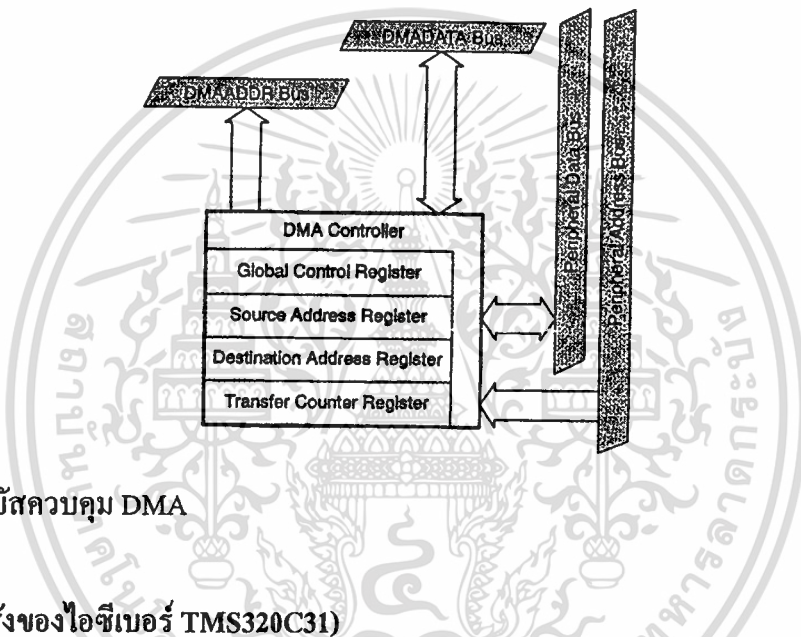
ภาพที่ 5.2.7 การติดต่อภายนอกของ peripheral module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.6 (ระบบถ่ายโอนข้อมูล Direct Memory Access (DMA))

TMS320C31 จะมีตัวควบคุม DMA ภายใน ซึ่งสามารถอ่านหรือเขียนลงไปตำแหน่งใดๆ ก็ได้ บนหน่วยความจำโดยไม่ไปขัดการทำงานของ CPU ดังนั้น TMS 320C31 จึงสามารถติดต่อกับหน่วยความจำภายนอกที่มีความเร็วเข้าได้ และสามารถติดต่อกับอุปกรณ์ภายนอก โดยไม่ไปรบกวนการทำงานของ CPU

ตัวควบคุม DMA จะมี แอดเดรสเจนเนอเรเตอร์, แหล่งกำเนิดรีจิสเตอร์, จุดมุ่งหมายของรีจิสเตอร์ และส่งผ่านการนับ การที่ DMA มีบัสข้อมูลและบัสแอดเดรส เป็นของตัวเอง จะทำให้การทำงานจะไม่ไปรบกวนการทำงานของหน่วยประมวลผลกลาง การทำงานของ DMA จะทำการย้ายข้อมูลเป็นบล็อกหรือเพียง 1 word ไปลงหน่วยความจำหรือนำออกมาจากหน่วยความจำ ภาพที่ 5.2.8



ภาพที่ 5.2.8 บัสควบคุม DMA

5.2.7 (คำสั่งของไอซีเบอร์ TMS320C31)

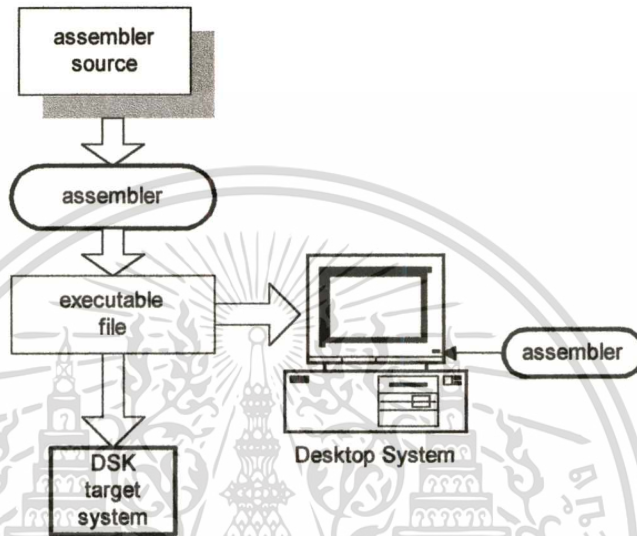
คำสั่งของไอซีเบอร์ TMS320C31 เป็นคำสั่งภาษาแอสเซมบลี ซึ่งเป็นคำสั่งเฉพาะของ ไอซีตระกูล TMS320C3x ซึ่งประกอบไปด้วยคำสั่งแบบธรรมดา กับคำสั่งแบบขนาน ที่จะทำให้มีการประมวลผลที่เร็วขึ้นซึ่งคำสั่งต่างๆ สามารถศึกษาได้จากหนังสืออ้างอิง [16]

5.2.8 (แอสเซมเบลอร์)

มีหน้าที่ในการแปลงไฟล์ข้อมูลภาษาแอสเซมบลี ให้เป็นไฟล์ภาษาเครื่อง หรือมีชื่อเรียกอีกอย่างว่า Object code ซึ่งทำให้ชิพ DSP สามารถนำข้อมูลไปปฏิบัติงานได้โดยตรงตัวแอสเซมเบลอร์ของDSKเป็นแบบที่สามารถกำหนด directive ก่อนที่จะทำการแอสเซมบลีได้ทำให้สามารถกำหนดตำแหน่งแอดเดรส ของโปรแกรมเป็นแบบสัมพัทธ์ คือเป็นแอดเดรสหลายๆที่กำหนดตำแหน่งที่แน่นอนได้ในภายหลัง ทำให้ไม่จำเป็นต้องมีขั้นตอนของการ Linker

5.2.9 (ดีบั๊กเกอร์)

มีหน้าที่ใช้ช่วยในการกำหนดตำแหน่ง และแก้ไขข้อผิดพลาด หรือ bug ต่างๆ ที่อาจเกิดขึ้นในตัวโปรแกรม การทำงานของโปรแกรมจะแบ่งการทำงานในส่วนต่างๆ แยกเป็นคณหน้าต่างๆ ที่ง่ายต่อการใช้งานโปรแกรมดีบั๊กเกอร์ จะมีหน้าที่พื้นฐานที่สำคัญได้แก่ Single step-Execution , Breakpoint setting , และ Runtime Execution Halt เป็นต้น



ภาพที่ 5.2.9 ขั้นตอนการพัฒนาซอฟต์แวร์



ภาพที่ 5.2.10 หน้าจอโปรแกรมดีบั๊กเกอร์ขณะติดต่อกับไมโครคอมพิวเตอร์

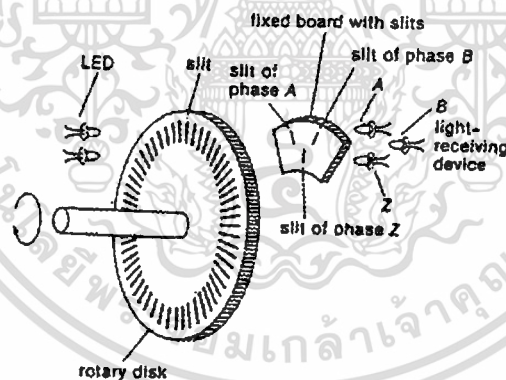
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 เซ็นเซอร์และเอนโคเดอร์ในระบบควบคุมมอเตอร์

ตัวเซ็นเซอร์ที่ติดอยู่กับเพลลาของมอเตอร์ มีจุดประสงค์เพื่อต่อร่วมกับระบบควบคุมให้เหมาะสม สำหรับการตรวจจับตำแหน่ง หรือตำแหน่งที่ต้องการควบคุมให้มีความแม่นยำ อย่างไรก็ตาม ตัวเซ็นเซอร์แต่ละชนิดมีการนำมาใช้งานในลักษณะที่ต่างกันขึ้นอยู่กับความเหมาะสมในการต่อร่วมกับมอเตอร์ชนิดนั้นๆ ด้วย และตัวเซ็นเซอร์บางชนิดไม่สามารถต่อร่วมกับระบบควบคุมที่มีการตรวจจับการหมุนของมอเตอร์ได้ทั้งหมด ดังนั้นในหัวข้อต่อไปนี้จะได้แสดงถึงลักษณะสำคัญและการใช้งานของตัวเซ็นเซอร์ชนิดอินคริमेंท์เอนโคเดอร์ดังต่อไปนี้

5.3.1 (อินคริमेंท์เอนโคเดอร์)

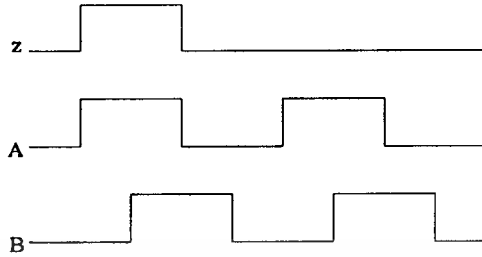
ในระบบการควบคุมตำแหน่งหรือความเร็วของมอเตอร์จะใช้อินคริमेंท์เอนโคเดอร์หรือเอนโคเดอร์สำหรับรักษาตำแหน่งและสำหรับสร้างสัญญาณป้อนกลับโดยที่ตัวเอนโคเดอร์จะสร้างสัญญาณพัลส์ที่แปรผันตรงกับการหมุนของเพลลาซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วของเพลลา มอเตอร์ในรูปของอัตราจำนวนพัลส์ได้ ดังในภาพที่ 5.3.1 แสดงกลไกของอินคริमेंท์เอนโคเดอร์ ซึ่งเป็นโครงสร้างของอินคริमेंท์เอนโคเดอร์ชนิดที่ตัวตรวจจับตำแหน่งของมุมมีเอาท์พุทเป็น A และ B โดยมีสัญญาณ Z เป็นมุมศูนย์ที่รับสัญญาณจากภายนอก



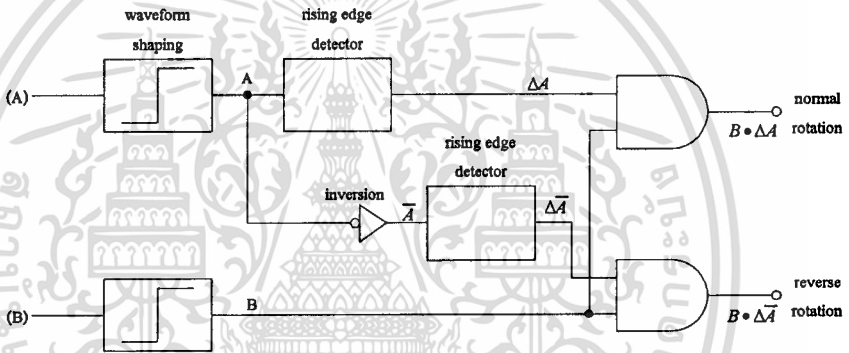
ภาพที่ 5.3.1 โครงสร้างของอินคริमेंท์เอนโคเดอร์

อินคริमेंท์เอนโคเดอร์ประกอบด้วยส่วนที่สำคัญคือตัวกำเนิดแสง งานหมุน (rotaty disk) งานอยู่กับที่และตัวเซ็นเซอร์ บนแผ่นงานหมุนทำเป็นช่องโคจรอบดังภาพที่ 5.3.1 และบนแผ่นอยู่กับที่จะมีช่องสำหรับให้แสงผ่านตรงเข้าไปยังตัวเซ็นเซอร์ได้ ส่วนตัวกำเนิดแสงอาจจะเป็นหลอดไฟหรือ LED ฉายแสงผ่านช่องที่หมุนรอบแผ่นกลมผ่าน แต่ละช่องของแผ่นดิสก์ที่ติดอยู่กับที่ของตัวรับแสง A ,B และ Z โดยที่แผ่นงานที่ติดอยู่กับที่แต่ละช่องของ A และ B มีมุมต่างเฟสกัน 90°

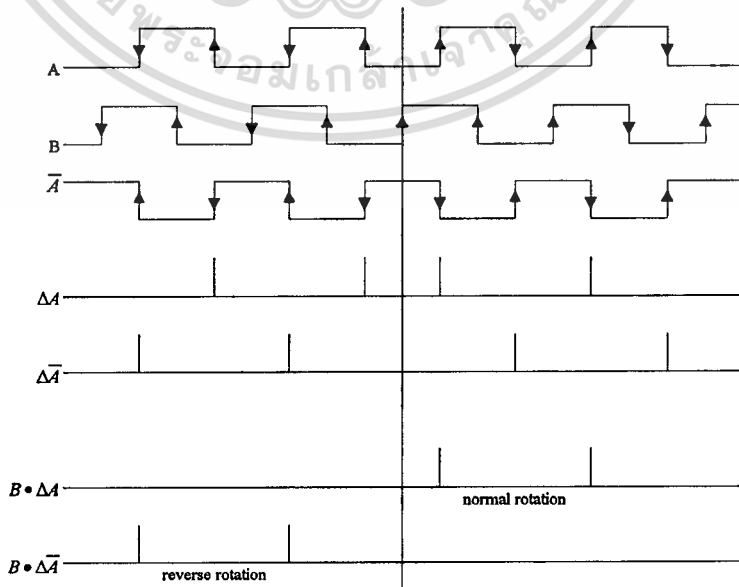
องศา, โดยที่เอาต์พุตจะเป็นรูปคลื่นสี่เหลี่ยมมีมุมต่างเฟสกัน 90° องศา, เช่นกันดังภาพ 5.3.2 ซึ่งเอาต์พุตสุดท้ายมีทิศทางตรงกันข้ามกับตัวรับแสง ส่วนในภาพที่ 5.3.3 และ 5.3.4 แสดงโครงสร้างและหลักการทำงานในการกำหนดทิศทางหมุน



ภาพที่ 5.3.2 แสดงถึงลูกคลื่นเอาต์พุตของอินคริमेंท์



ภาพที่ 5.3.3 บล็อก ไดอะแกรมของการกำหนดทิศทางการหมุน



ภาพที่ 5.3.4 ไดอะแกรมลูกคลื่นของการกำหนดทิศทางการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1.1 (ความละเอียดของอินคริमेंท์เอนโคดเดอร์)

ความละเอียดของเอนโคดเดอร์คือจำนวนคาบเวลาของสัญญาณเอาต์พุตต่อการหมุนของเพลา 1 รอบ ซึ่งบอกเป็นจำนวนพัลส์ต่อรอบหรือจำนวนไซเคิลต่อ 360° องศา มุมทางเชิง กล หรือไซเคิลต่อองศา เอนโคดเดอร์ที่ใช้กันทั่วๆ ไปมีความละเอียดตั้งแต่ 15 ถึง 10,000 พัลส์ต่อรอบ จำนวนพัลส์ต่อ 1 รอบของสัญญาณ ที่ตัวเอนโคดเดอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุน และความกว้างของช่องว่างกับความกว้างของแถบที่ระหว่างช่องว่างจะเท่ากัน เพราะฉะนั้นเราสามารถคำนวณหาความกว้างของช่องว่าง (W) ได้จาก

$$w = \frac{\pi D}{2N} \quad (5.3.1)$$

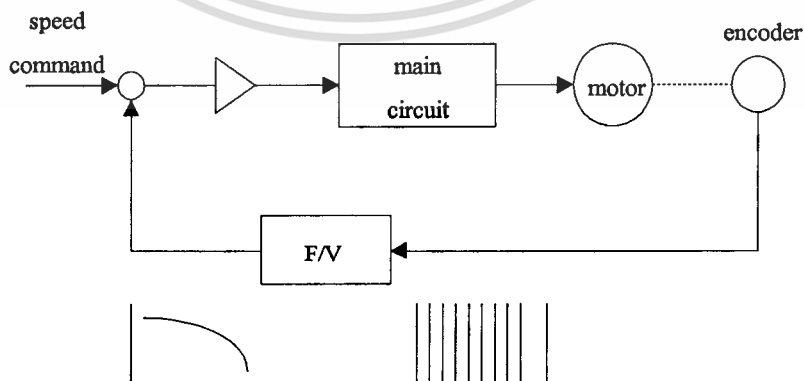
เมื่อ D = เป็นเส้นผ่าศูนย์กลางเฉลี่ยของเพ็นเทิน

N = จำนวนของความละเอียดเป็นพัลส์ต่อรอบ

W = ความกว้างของช่องว่าง

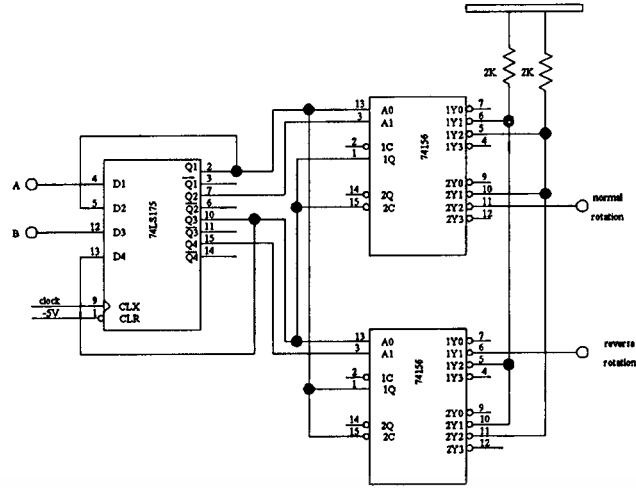
5.3.1.2 (การตรวจจับความเร็วในการหมุนของอินคริमेंท์เอนโคดเดอร์)

อินคริमेंท์เอนโคดเดอร์จะไม่ผลิตพัลส์ ดังนั้นจำนวนเอาต์พุตของพัลส์ที่ตัวเอนโคดเดอร์จะแปลงสัญญาณอนาลอกที่เป็นสัดส่วนกับความถี่ของพัลส์ โดยจะแปลงจากความถี่ไปเป็นแรงดันโวลต์ (F/V) , ดังแสดงในภาพที่ 5.3.5 ซึ่งอาจจะเป็นไปได้ที่สัญญาณอนาลอกเป็นตัวตรวจจับความเร็วในการหมุน และในกรณีที่เมื่อจำนวนพัลส์มีค่าน้อยๆ ที่เป็นเหตุทำให้ยากต่อการนำมาใช้งาน ดังนั้นจึงสามารถเพิ่มจำนวนพัลส์ได้โดยใช้วงจรหารความถี่ด้วย 4 ดังแสดงในภาพที่ 5.3.6 ซึ่งจะทำการแปลงสัญญาณอนาลอกโดยแปลงจากความถี่เป็นแรงดัน หรือ F/V



ภาพที่ 5.3.5 บล็อกไดอะแกรมแสดงการตรวจจับความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

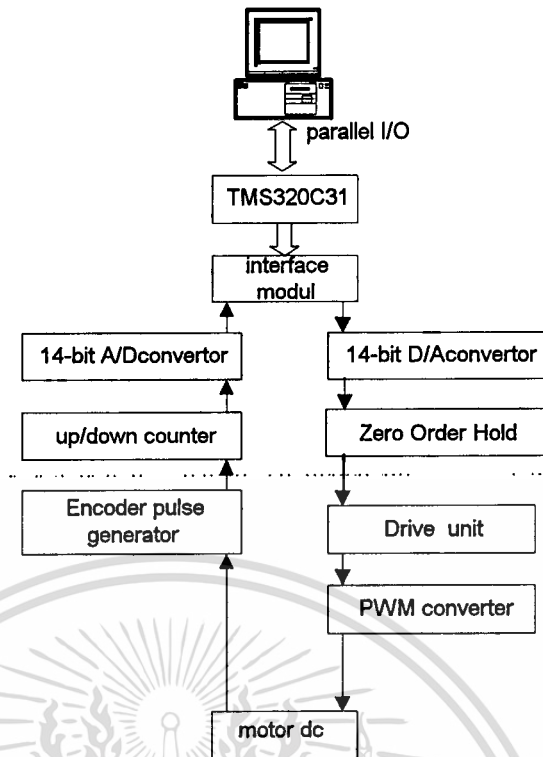


ภาพที่ 5.3.6 วงจรหารความถี่

5.4 การออกแบบโครงสร้างทางฮาร์ดแวร์

ในบทนี้จะเป็นการนำหลักการของ PIDA มาสร้างเป็นเครื่องต้นแบบเพื่อนำไปควบคุมมอเตอร์ไฟฟ้ากระแสตรงชนิดแยกตัวกระตุ้น โดยระบบควบคุมดิจิทัลที่สร้างขึ้นใช้ DSP เป็นหน่วยประมวลสัญญาณเชิงเลข มาช่วยในการคำนวณทางด้านคณิตศาสตร์ ที่มีความเร็วในการทำงานคำสั่งทางคณิตศาสตร์ 50 MFLOPS และมีความเร็วในการทำงาน 25 ล้านคำสั่งต่อวินาที เป็นชิพของ Texas Instrument เบอร์ TMS320C31 ต่ออินเตอร์เฟซกับพอร์ตขนาน ของไมโครคอมพิวเตอร์ โดยใช้วงจรแปลง A/D และ D/A เบอร์ TLC320C40 ที่มีความละเอียดในการแปลงข้อมูลขนาด 14 บิต ซึ่งต่อรวมอยู่ในบอร์ดประมวลผลสัญญาณเชิงเลขของชิพ DSP สามารถศึกษาได้จากคู่มือ [16] และในการเขียนโปรแกรมนั้นพิจารณาให้อุปกรณ์เป็น Zero Order Hold (ZOH) ต่อรวมอยู่กับกระบวนการ จากนั้นทำตามอัลกอริทึมของตัวควบคุม PIDA ซึ่งเอาต์พุตที่ได้นั้นเป็นการกระทำด้วยอุปกรณ์ (ZOH) Z-Transform เพื่อต่อเชื่อมในการควบคุมกระบวนการที่เป็นมอเตอร์กระแสตรงชนิดแยกตัวกระตุ้น และสามารถเขียนโคดแอมการทดลองได้ดังภาพที่ 5.4.1

ในการทดลองการควบคุมความเร็ว และตำแหน่งที่มอเตอร์ได้เคลื่อนที่ไปได้อย่างแม่นยำนั้น มีหลักการสำคัญ ๆ อยู่ว่าถ้าทำการป้อนสัญญาณเข้าไปขับให้มอเตอร์หมุนแล้ววัดตำแหน่งของการหมุนด้วยเอ็นโคเดอร์ จากนั้นนำค่าที่วัดได้ไปเปรียบเทียบกับค่าเป้าหมาย หรือ set point เดิมจากนั้นทำการควบคุมสัญญาณส่งไปเพิ่ม หรือ ลดความเร็วของมอเตอร์เพื่อทำให้มอเตอร์มีความเร็วตามต้องการ โดยที่การรับค่าจากเอ็นโคเดอร์นั้น ตัวเอ็นโคเดอร์ จะส่งสัญญาณตรวจจบการหมุนของมอเตอร์ออกมาในรูปของพัลส์ความถี่ในหน่วยของพัลส์ต่อรอบ จากนั้นนำสัญญาณส่วนที่ได้นี้ไปเข้าวงจรนับ 32 บิต ที่ทำการออกแบบไว้ เพื่อแปลงค่าพัลส์ให้เป็นข้อมูลดิจิทัลขนาด 32 บิต แล้วจึงนำค่าขนาด 32 บิต นี้ไปประมวลผลด้วย DSP ต่อไป ซึ่งโดยทั่วไปแล้วการส่งข้อมูลเข้า

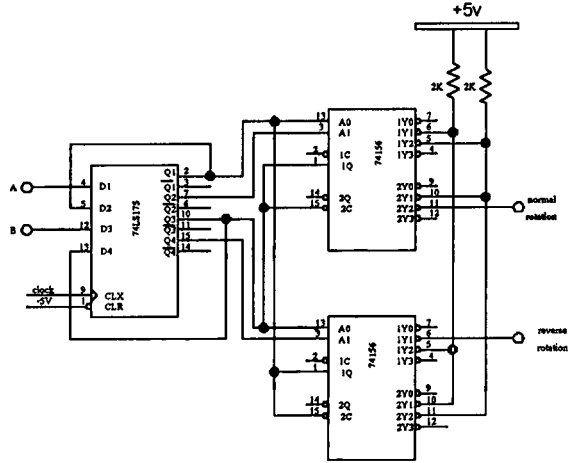


ภาพที่ 5.4.1 ใคอะแกรมการควบคุมความเร็วและตำแหน่งมอเตอร์กระแสตรง

ไปประมวลผลนั้นจะต้องมีการอ้างตำแหน่ง Address ให้หน่วยประมวลผลสัญญาณเชิงเลข เพื่อที่หน่วยประมวลผลกลางจะได้อ้างอิงตำแหน่งของข้อมูลได้ถูกต้องและแม่นยำ แล้วส่งค่าออกมาเป็นสัญญาณอนาล็อกอีกครั้งหนึ่ง เพื่อไปควบคุมมอเตอร์ต่อไปและจากการที่จะรับข้อมูลการหมุนจาก Encoder ได้นั้น จะต้องมียวงจรสองส่วนทำงานร่วมกันคือ วงจรตรวจจับทิศทางการหมุน และวงจรนับ 32 บิต

5.4.1 (วงจรตรวจจับทิศทางการหมุน)

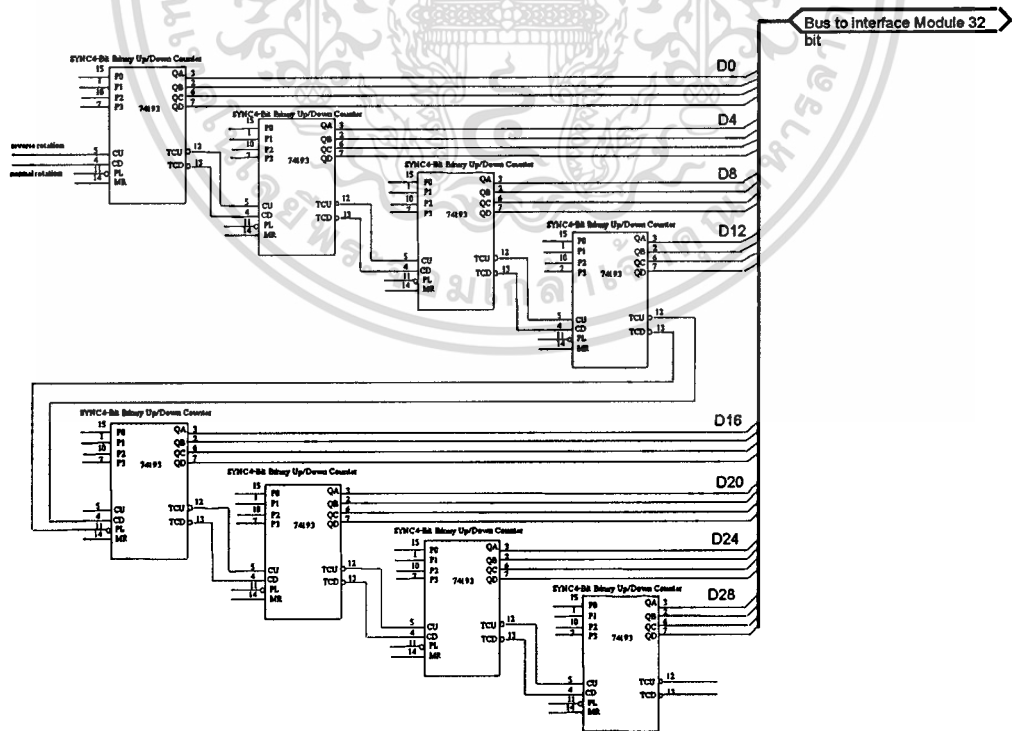
จากวงจรดังภาพที่ 5.4.2 จะทำหน้าที่ตรวจจับทิศทางการหมุนของมอเตอร์ ว่ามีทิศทางการหมุนตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา โดยรับอินพุตจากตัว Encoder ในรูปของสัญญาณ A,B และรับสัญญาณการทำงานของ Clock จากเครื่องกำเนิดความถี่ จากการออกแบบวงจรโดยใช้ IC TTL 74LS175 จำนวน 1 ตัว และ IC TTL 74LS156 จำนวน 2 ตัว ซึ่งการทำงานและการต่อวงจรเป็นดังภาพที่ 5.4.2



ภาพที่ 5.4.2 วงจรตรวจจับทิศทางการหมุนและหาความถี่

5.4.2 (วงจรมับ 32 บิต)

วงจรมับทำหน้าที่นับพัลส์เพื่อแสดงตำแหน่งที่มอเตอร์ได้เคลื่อนที่ไปในรูปของ normal rotation และ reverse rotation โดยใช้ IC TTL 74LS193 ทำการออกแบบให้วงจรมับได้ 32 ตำแหน่งซึ่งพัฒนามาจากวงจร 16 บิต [2] เพื่อใช้ข้อมูล 32 บิต นี้ส่งให้กับหน่วยประมวลผลสัญญาณ TMS320C31 ซึ่งใช้ IC TTL 74LS193 ต่อ cascade กันจำนวน 8 ตัว ดังภาพที่ 5.4.3



ภาพที่ 5.4.3 วงจรมับ 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 (วงจรรินเตอร์เฟสโมดูล)

วงจรรินเตอร์เฟสออกแบบมาเพื่อส่งข้อมูลไปยังหน่วยประมวลผลสัญญาณตามเงื่อนไขในการส่งค่า 32 บิต จากวงจรริน 32 บิต ไปยังหน่วยประมวลผลสัญญาณ TMS320C31 โดยใช้สัญญาณที่เกี่ยวข้องกับการอ่านค่าของหน่วยประมวลผลสัญญาณ TMS320C31 ซึ่งที่ตัวมีขาสัญญาณที่เกี่ยวข้องและค่า Logic ที่ใช้ในการอ่าน และเขียนตามตารางที่ 5.4.1 โดยต่อขาทั้งหมดกับ

ตารางที่ 5.4.1 แสดงสถานะของขาสัญญาณ

ขาสัญญาณที่เกี่ยวข้อง	Logic
R/W	high
STRB	high

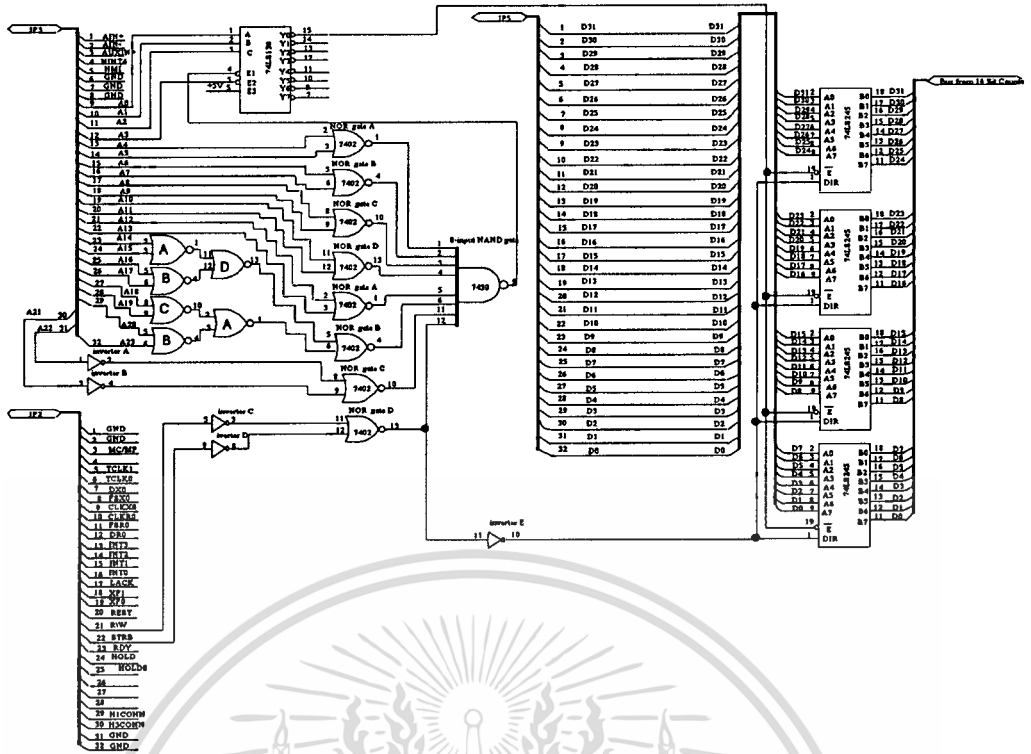
Logic gate เพื่อสร้างเงื่อนไขในการอ่านและเขียน ซึ่งขณะที่ขาสัญญาณทั้งสอง ที่แสดงค่าดังตารางที่ 5.4.1 นั้นต้องการใช้สำหรับส่งสัญญาณอย่างเดียว โดยอาจจะเป็น Logic high หรือ Logic low ก็ได้ ซึ่งตามวงจรที่ได้ออกแบบจะนำขา R/W และ STRB มาผ่าน Not gate ทั้งคู่ แล้วนำไปเข้าอินพุตของ NOR gate A เพื่อให้ได้ Logic high แต่การอ้างอิง Logic ข้างต้นนั้นยังไม่เพียงพอเพราะจะต้องมีการอ้างอิง Address ด้วย ซึ่งการจะรับค่า 32 บิต จากภายนอกนั้นหน่วยประมวลผลสัญญาณที่นำมาใช้ ได้กำหนดพอร์ตรับข้อมูลจากภายนอกไว้แล้ว โดยกำหนดให้สามารถรับได้ทั้งหมด 32 พอร์ต และได้กำหนดชื่อว่า PA0-PA31 หรือหมายถึงพอร์ต 0 ถึงพอร์ต 31 ซึ่งกำหนดจาก Address map ดังนั้นพอร์ตที่รับข้อมูลมีเพียงพอร์ตเดียว ส่วนการกำหนดพอร์ตรับค่าที่ Address ที่อ้างอิงคือตำแหน่ง 0110 0000 0000 0000 0000 0000 ในเลขฐานสองนั้นก็คือ พอร์ต 0 หรือ Address 0coch แสดงได้ดังนี้

MSB	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ในการอ้าง Address นั้นจะนำบิตที่มี Logic low ยกเว้น LSB ตำแหน่งที่ 1, ตำแหน่งที่ 2, และตำแหน่งที่ 3. มาผ่าน NOR gate เป็นคู่ๆ เพื่อสร้าง Logic high ส่วนตำแหน่งที่เป็น Logic high นั้นก็จะนำมาผ่าน inverter แล้วนำมาเข้า NOR gate เพื่อสร้างเงื่อนไขขึ้นมาทั้งหมด แล้วป้อนให้กับ 8 อินพุต NAND gate เพื่อสร้าง Logic high จากนั้นนำสัญญาณที่ได้ขึ้นไป Enable 3-Line-to-8-Line Decoder โดยนำเอา 3 ขาแรกที่ยังไม่ใช้มาถอดรหัสจาก 000 เป็น 10000000 (จาก 3 บิต เป็น 8 บิต) เพื่อนำสัญญาณที่ได้ใหม่นี้ไป Enable Octal Bus Transceivers 2 ตัว โดยจะต้องกำหนดทิศทางการไหลของข้อมูลขนาด 32 บิต ด้วยเงื่อนไขอีกอันหนึ่งเพื่อไปกำหนดขา DIR ของ Octal Bus Transceivers โดยที่จะใช้ Logic ที่สร้างไว้แล้วมาใช้ประโยชน์คือ Logic high ที่ได้จากการสร้างเงื่อนไขของขาสัญญาณ STRB และ R/W มาใส่ inverter แล้วนำมาเข้าขา DIR อีกครั้งหนึ่ง โดยวงจรทั้งหมดแสดงได้ดังภาพที่ 5.4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

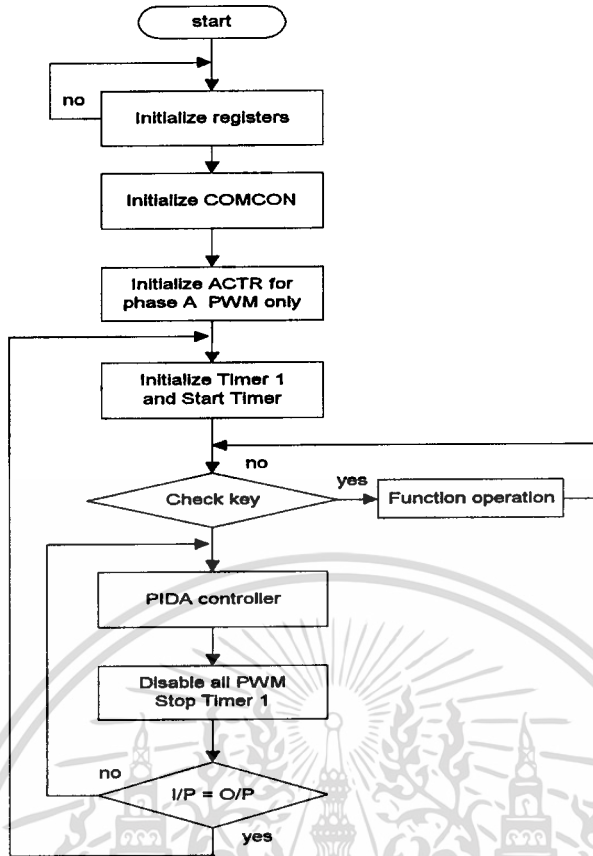
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



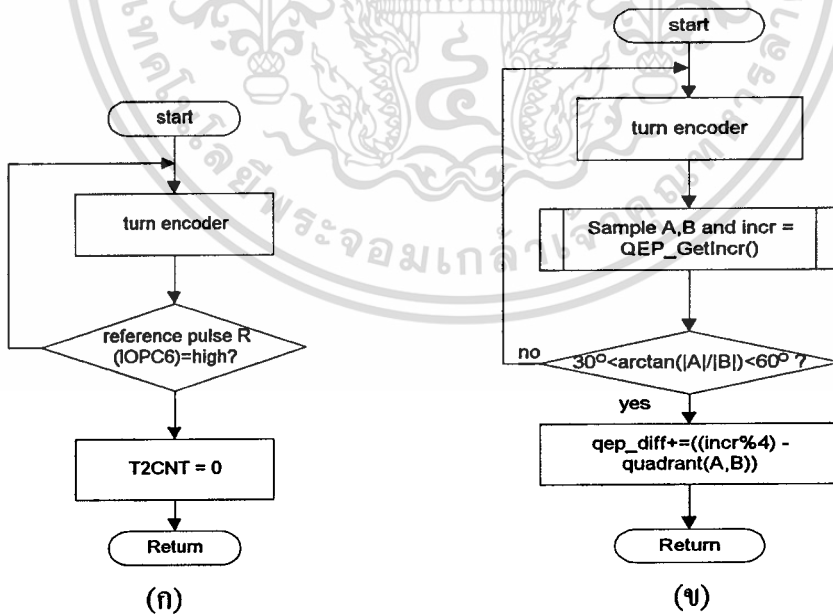
ภาพที่ 5.4.4 วงจรอินเทอร์เฟซโมดูล TMS320C31

5.6 การออกแบบซอฟต์แวร์

ในการควบคุมมอเตอร์ดิจิตอลแยกตัวกระตุ้นแบบป้อนกลับนี้ โดยการทำงานเริ่มจากโปรแกรมจะทำการ Initialization ระบบก่อน โดยที่การปรับอัตราเร็วของมอเตอร์ดิจิตอลนั้นใช้วงจรสับอิเล็กทรอนิกส์กำลัง โดยอาศัยขุมเฟสระหว่างสองขบวนพัลส์ A และ B ซึ่งได้จากการกำหนดทิศทางหมุนของอินคริमेंท์ดังกล่าวมาแล้วในหัวข้อ 5.3.1 และค่าของพัลส์ นี้จะมีค่าขึ้นอยู่กับผลต่างของอัตราเร็วที่ต้องการกับอัตราเร็วที่แท้จริงของมอเตอร์ โดยโปรแกรมจะทำให้มีการหน่วงเวลาเพื่อให้อุปกรณ์จากภายนอกได้เตรียมพร้อม หลังจากนั้นไมโครคอมพิวเตอร์จะทำการตรวจสอบคีย์เพื่อกำหนดฟังก์ชันการทำงาน แล้วนำไปเปรียบเทียบกับอัตราเร็วที่แท้จริงของมอเตอร์ว่าได้เคลื่อนที่ไป ซึ่งไมโครคอมพิวเตอร์จะส่งข้อมูลขนาด 32 บิต ที่ได้จากวงจรนับ 32 บิต ข้อมูลจะถูกส่งไปสู่พอร์ทศูนย์ของหน่วยประมวลผลเชิงเลข ที่กำหนด Address ในการอ้างอิงตำแหน่งดังกล่าวในหัวข้อ 5.4 ในเลขฐานสองนั้นคือ 0110 0000 0000 0000 0000 0000 ที่พอร์ท 0 หรือ Address 0c0h จากนั้นหน่วยประมวลผลเชิงเลข จะทำการคำนวณอัตราเร็วของมอเตอร์ โดยกระบวนการควบคุมแบบ PIDA ทำให้ส่งสัญญาณควบคุมไปยังวงจรอิเล็กทรอนิกส์กำลัง จากนั้นจะทำการตรวจสอบค่าความเร็วอินพุทที่กำหนดไว้กับความเร็วเอาต์พุทตามต้องการที่กำหนดไว้ ซึ่งถ้ามีค่าเท่ากันก็จะทำขั้นตอนต่อไปดังภาพที่ 5.6.1

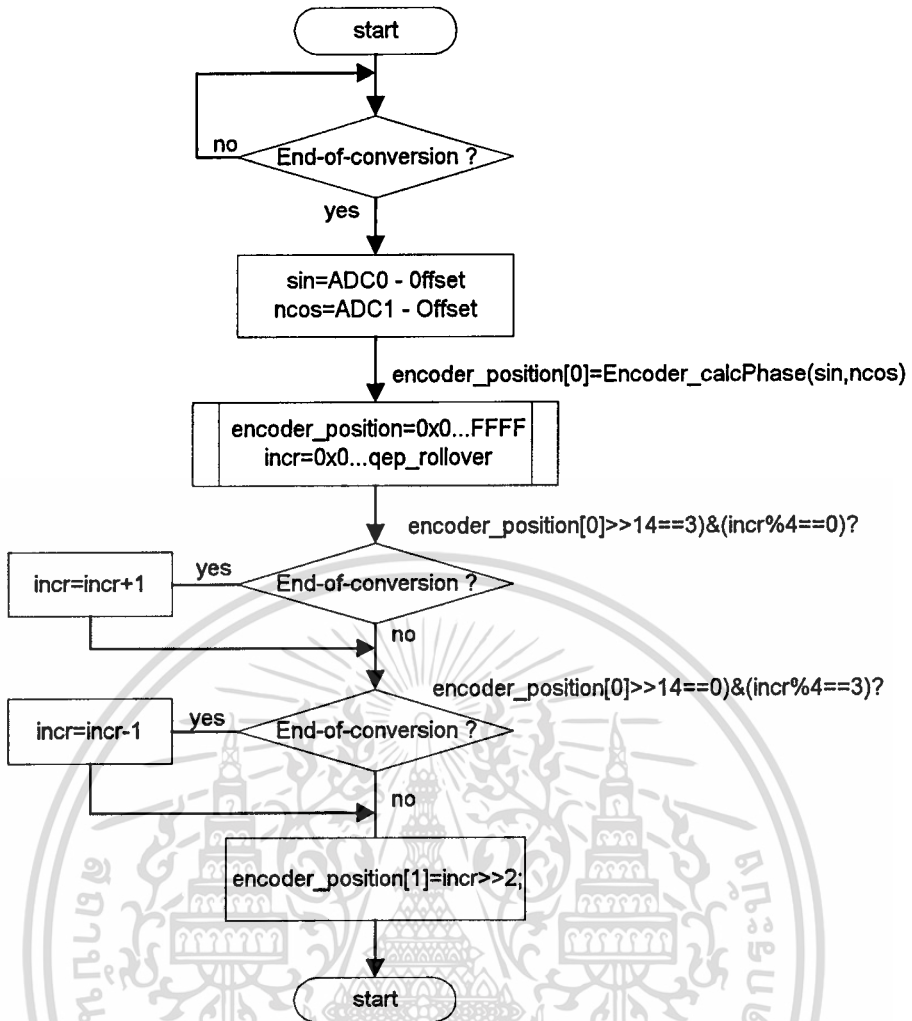


ภาพที่ 5.6.1 โปรแกรมหลักในการควบคุมวงจรถอนิกส์กำลังด้วยตัวควบคุม PIDA สำหรับควบคุมมอเตอร์



ภาพที่ 5.6.2 โค้ดแกรมโปรแกรมย่อยในการควบคุมตำแหน่ง (ก) เมื่อขณะอยู่ที่ตำแหน่งศูนย์องศา (ข) เมื่อขณะเริ่มเคลื่อนที่และนับตำแหน่งที่เคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.6.3 โค้ดแกรมแสดงการคำนวณหาตำแหน่งของ เอนโคเดอร์

บทที่ 6

ผลการเลียนแบบและผลการทดลอง

องค์ประกอบส่วนใหญ่ของระบบควบคุมตำแหน่งอาจแบ่งระบบควบคุมมอเตอร์ ออกเป็น 2 ส่วนคือวงจรกำเนิดสัญญาณพลัส และวงจรอิเล็กทรอนิกส์กำลังโดยใช้วงจรตีซีชอปเปอร์ในการจ่ายแรงดันให้กับวงจรอาร์มเจอร์ ส่วนระบบวัดความเร็วหรือความเร็วอ้างอิงตำแหน่งของมอเตอร์ ใช้หน่วยประมวลผลสัญญาณเชิงเลขและไมโครคอมพิวเตอร์ โดยที่ค่าพารามิเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงแบบแยกตัวกระตุ้นที่ใช้ในการทดลองมีขนาด 330 วัตต์ แรงดันไฟฟ้า 220 โวลต์ และ กระแส 2.5 แอมป์ ความเร็วรอบสูงสุด 2250 รอบ/นาที และค่าพารามิเตอร์ที่ใช้ในการทดลองมีดังนี้ $R_a = 0.038 \Omega$, $L_a = 0.2295 \text{ H}$, $K_e = 0.0934 \text{ Vs/rad}$, $K_t = 0.934 \text{ Nm/A}$, $J = 2.0975 \times 10^{-3} \text{ Kg-m}^2$, $B = 0.00595 \text{ Nm-s/rad}$, ทำการหาค่าตัวควบคุม PIDA ตามขั้นตอนดังแสดงในบทที่ 4 และฟังก์ชันถ่ายโอนของมอเตอร์ที่ใช้ในการทดลองนั้น เมื่อแทนค่าพารามิเตอร์ลงในสมการที่ (2.56) จะได้สมการดังนี้

$$\frac{674.6187}{s^3 + 1.215s^2 + 674.6187s} \quad (6.1)$$

และจะได้ค่าเกณฑ์ของตัวควบคุมที่ใช้ในการทดลองดังนี้ $K_p = 21.6768$, $K_D = 920.8518$, $K_I = 1203.7$, $K_A = 42345.679$ โดยที่ค่าเกณฑ์ หาได้จากหลักการของตัวควบคุม PIDA ด้วยวิธีพิจารณาค่ารากหลักของรูตโลกัส ดังนั้นที่ฟังก์ชันถ่ายโอนในวงรอบเปิดของ $G(s)$ ของตัวควบคุมกับแบบจำลองของกระบวนการในรูปของ $G(s) = G_c(s) G_p(s)$ เป็นดังนี้

$$G(s) = \frac{21.6768s^3 + 920.8518s^2 + 12703.7s + 42345.679}{s} \times \frac{674.6187}{s^3 + 1.215s^2 + 674.6187s} \quad (6.2)$$

เมื่อต่อตัวควบคุม PIDA เป็นแบบป้อนไปข้างหน้า (Forward control) หรืออาจเรียกการต่อตัวควบคุมในลักษณะนี้ว่า “ การควบคุม PIDA โรบัสต์ ” ดังภาพที่ 4.8 ซึ่งเป็นวิธีการต่อที่ทำให้ระบบควบคุมมีเสถียรภาพสัมพัทธ์ที่ดี และมีความคงทนมากยิ่งขึ้น ดังนั้นเขียนฟังก์ชันถ่ายโอนตัวชดเชยแบบป้อนไปข้างหน้าได้ดังนี้ โดยที่สามารถศึกษารายละเอียดวิธีการออกแบบได้ในหัวข้อที่ 7.3.1 และฟังก์ชันถ่ายโอนตัวชดเชยแบบป้อนไปข้างหน้าหาได้จากสมการ (7.3) และจากภาพที่ 4.8 ความเร็วถูกควบคุมโดยตัวควบคุม PI ส่วนการควบคุมตำแหน่งนั้นใช้ตัวควบคุม PIDA ดังนั้นจึงกำหนดตัวควบคุมใหม่เป็น $K_A = K_3$, $K_D = K_2$, $K_p = K_1$, $K_I = K_0$, ดังนั้นเขียนฟังก์ชันถ่ายโอนในวง

รอบปิดที่ได้จากการออกแบบเป็นดังนี้

$$\frac{\theta(z)}{\omega_r(z)} = \frac{KK_0}{s^4 + (1.215 + KK_3)s^3 + (674.6187 + KK_2)s^2 + KK_1s + KK_0} \quad (6.3)$$

และจากสมการ (6.3) จะได้ฟังก์ชันของตัวควบคุมในวงรอบปิดดังนี้ เมื่อเกณฑ์ $K = 1$

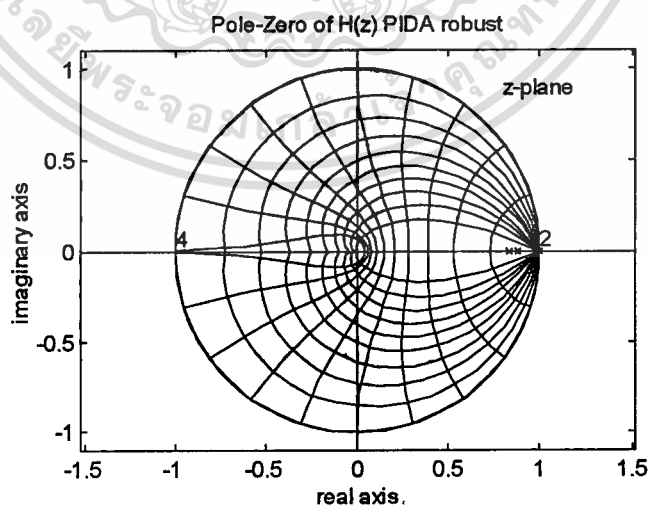
$$\frac{\theta(z)}{\omega_r(z)} = \frac{2.857 \times 10^7}{s^2 + 1.462 \times 10^4 s^3 + 6.219 \times 10^5 s^2 + 8.57 \times 10^6 s + 2.857 \times 10^7} \quad (6.4)$$

ผลตอบสนองของเอาต์พุตต่ออินพุตด้วยฟังก์ชันขั้นหนึ่งหน่วยในวงรอบปิดจะได้ดังภาพที่ 6.2 จะเห็นว่าระบบควบคุมมีความนุ่มนวลในการเข้าสู่สภาวะคงตัว และให้ผลตอบสนองตรงตามคุณสมบัติหรือจะพิจารณาในโดเมนที่ของแชนเดราน์พร้อมจากสมการที่ (6.4) ให้อยู่ในรูปของ H

$(z) = \frac{\theta(z)}{\omega_r(z)}$ ได้ด้วยการใช้การแปลงไบลิเนียร์จากโปรแกรม Matlab ได้ดังนี้

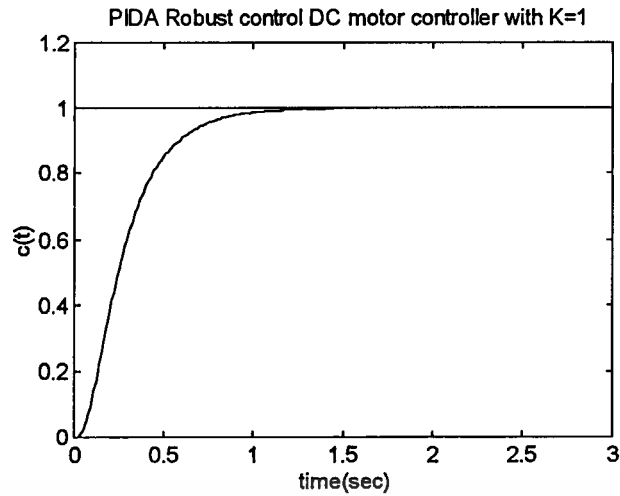
$$H(z) = \frac{\theta(s)}{\omega_r(s)} = \frac{0.0541 + 0.2164z^{-1} + 0.3246z^{-2} + 0.2164z^{-3} + 0.5541z^{-4}}{1 - 3.7215z^{-1} + 5.1834z^{-2} - 3.2023z^{-3} + 0.7404z^{-4}} \quad (6.5)$$

จากสมการที่ (6.5) นำไปวิเคราะห์ในเชิงดิจิทัลด้วยกราฟโพลัส ดังภาพที่ 6.1 ซึ่งจะเห็นได้ว่าระบบควบคุมยังคงมีเสถียรภาพ



ภาพที่ 6.1 ผลการวิเคราะห์เสถียรภาพของระบบเวลาไม่ต่อเนื่อง

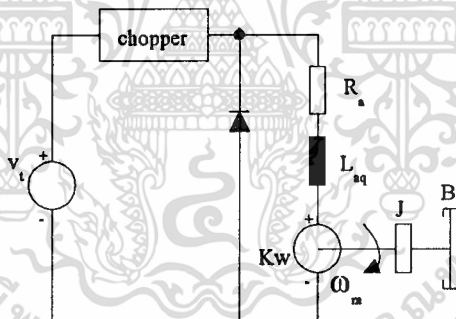
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.2 ผลตอบสนองของฟังก์ชันขั้นหนึ่งหน่วยเมื่ออัตราขยายเกนที่ $K = 1$

6.1 การออกแบบควบคุมความเร็วโดยใช้ ดีซีชอปเปอร์ (dc chopper)

วงจรโดยทั่วไปที่ใช้ในการจำลองแบบของ ดีซีชอปเปอร์ หรือแหล่งจ่ายแรงดันไฟกระแสตรงที่มีค่าไม่คงที่ จะเป็นดังภาพที่ 6.3 โดยที่ free-wheeling diode มีเพื่อช่วยปล่อยพลังงานที่สะสมในขดลวดเหนี่ยวนำ



ภาพที่ 6.3 วงจรควบคุมความเร็วมอเตอร์ด้วยการจำลองโดยใช้ ดีซีชอปเปอร์

การทำงานของวงจรชอปเปอร์ที่ใช้ควบคุมความเร็วมอเตอร์กระแสตรงอธิบายได้ดังนี้

T_{ON} คือ on time ของดีซีชอปเปอร์

T_{OFF} คือ off time ของดีซีชอปเปอร์

D คือ % ของ duty cycle ของดีซีชอปเปอร์

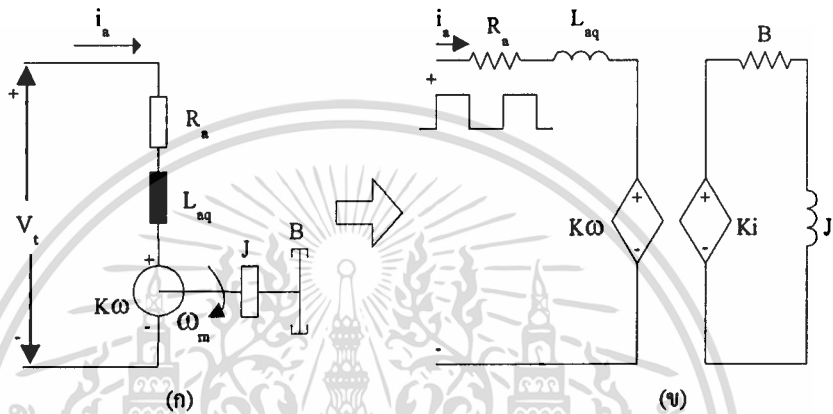
$$D = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100 \quad (6.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ ซอปเปอร์อยู่ในสภาวะ no time ความต่างศักย์เท่ากับ V_t ดังสมการที่ (2.30) ในบทที่ 2 และเมื่อ อยู่ในสภาวะ off time ความต่างศักย์เท่ากับศูนย์ ตามกฎของ KVL จะได้สมการเป็นดังนี้

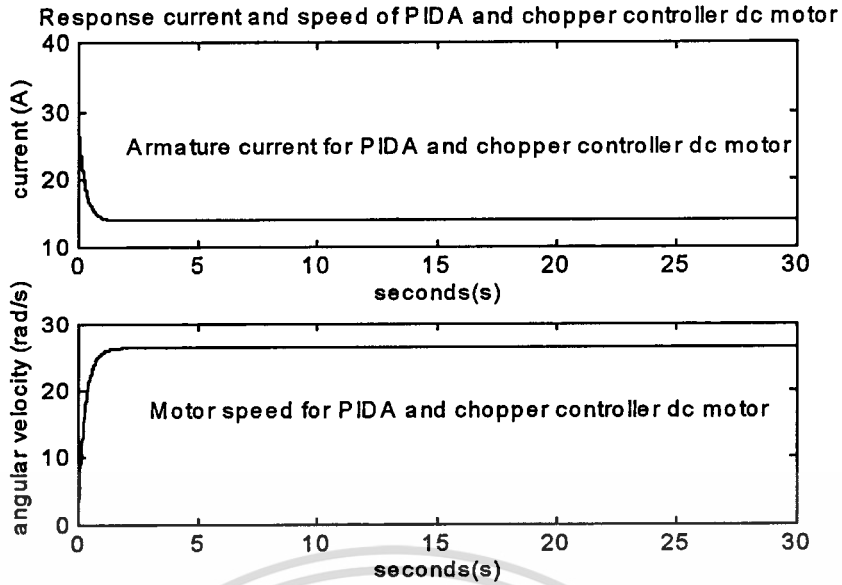
$$0 = e_L + R_a i_a + L_{aq} \frac{di_a}{dt} \quad (6.7)$$

และการทำงานของวงจรคิซีซอปเปอร์ แสดงดังภาพที่ 6.2 (ก)



ภาพที่ 6.4 แบบจำลองการทำงานของวงจรคิซีซอปเปอร์

จากภาพที่ 6.4 (ข) เป็นวงจรที่ใช้ในการเขียนแบบวงจรควบคุมของคิซีซอปเปอร์ ผลการเขียนแบบของคิซีซอปเปอร์ ดังภาพที่ 6.5 ซึ่งเป็นจำลองแบบพฤติกรรมการทำงานของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นที่ควบคุมด้วยคิซีซอปเปอร์ เพื่อนำไปใช้งานในทางปฏิบัติจริงโดยแสดงผลตอบสนองของระบบ ซึ่งเป็นสิ่งสำคัญในการศึกษาการทำงานของระบบ โดยการใช้การจำลองแบบการทำงานกับอุปกรณ์จริงซึ่งในการทดลองกับอุปกรณ์จริงนั้นมีความยุ่งยากเนื่องจากการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ดังนั้นจึงได้ทำการเขียนแบบการทำงานของระบบควบคุมด้วยโปรแกรม Matlab โดยการใช้การแปลงระบบเครื่องกลให้เป็นทรานเฟอร์ฟังก์ชันของระบบ เพื่อดูผลตอบสนองของระบบควบคุมดังกล่าวว่าระบบควบคุมที่ได้ออกแบบนั้นมีคุณสมบัติในการตอบสนองที่เร็วตามต้องการหรือไม่ และเพื่อดูเสถียรภาพของระบบด้วยว่ามีความคงทนมากน้อยเพียงใด ในการนำไปใช้งานจริง ซึ่งในที่นี้จะแสดงพฤติกรรมผลตอบสนองของกระแสอาร์เมเจอร์ และความเร็วของมอเตอร์ที่ควบคุมโดยใช้คิซีซอปเปอร์ ดังแสดงในภาพที่ 6.5

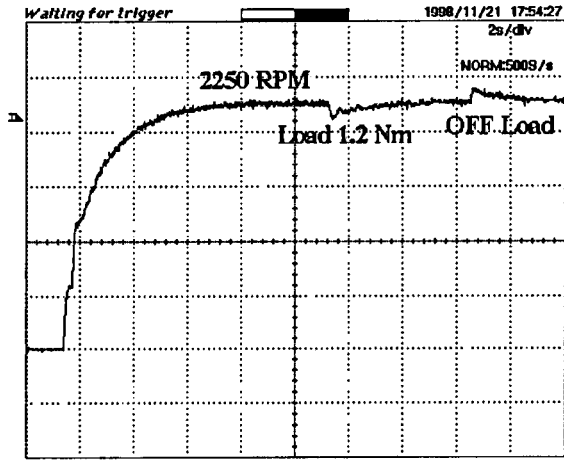


ภาพที่ 6.5 ผลตอบสนองของกระแสอาร์เมเจอร์ และความเร็วของมอเตอร์

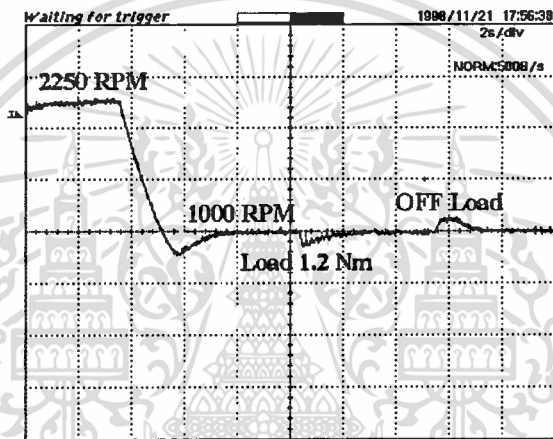
6.2 การทดลองควบคุมมอเตอร์ด้วยไมโครคอนโทรลเลอร์

วิธีการนี้จะเป็นการควบคุมความเร็วของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น โดยอาศัยวงจรอิเล็กทรอนิกส์กำลัง ซึ่งเป็นวิธีการทางดิจิทัลเชิงตัวเลข ที่ควบคุมด้วยหน่วยประมวลผลสัญญาณเชิงเลข เบอร์ TMS320C31 ดังแสดงในภาพที่ 5.3.1 วิธีการนี้ให้ข้อได้เปรียบในการทำงานที่เร็วและมีความแม่นยำสูง ราคาถูก ความสามารถในการปรับปรุงตัว ความเชื่อถือได้ และความต้องการกำลังป้อนเข้า ในการวัดความเร็วที่แท้จริงจะได้จากการวัดความถี่ของเอาท์พุท จากชาฟต์เอ็นโคเดอร์ ซึ่งใช้แบบอินคริเมนต์ ที่มีความละเอียด 360 พัลส์ต่อรอบ ติดตั้งไว้บนเพลลาของมอเตอร์ โดยจะให้สัญญาณในรูป A,B ที่ทำการสุ่มในช่วงเวลา 100 มิลลิวินาที แล้วทำการนับด้วยวงจรมับความถี่และโปรแกรมคำนวณการนับพัลส์ ซึ่งจะให้ความเร็วของมอเตอร์ที่แท้จริงในหน่วยรอบต่อนาที แล้วเก็บไว้ที่ในหน่วยประมวลผลสัญญาณเชิงเลขกับไมโครคอนโทรลเลอร์ตอนปลายของช่วงเวลามุมเพื่อเปรียบเทียบกับความเร็วที่ต้องการ ทำให้ได้สัญญาณผลต่างเกิดขึ้น หรือค่าผิดพลาด สัญญาณผลต่างที่ได้นี้จะทำให้โปรแกรมมับความถี่ของวงจรถดเชย (compensator) ทำการควบคุมเฟสระหว่างสองขบวนพัลส์ A และ B เพื่อควบคุม “เวลาเปิดกระแส” ของเอสซีอาร์ในวงจรถดเชยอิเล็กทรอนิกส์กำลัง ซึ่งให้แรงดันเอาท์พุทเข้าควบคุมมอเตอร์กระแสตรง โดยอาศัยวัฏจักรหน้าที่ ที่ขึ้นอยู่กับมุมเฟสสัมพันธ์ ระหว่างขบวนพัลส์ทั้งสอง วงจรมับควบคุมจะส่งสัญญาณดิจิทัลไปยังวงจรถดเชยเพื่อปรับเปลี่ยนมุมเฟส ให้มีพิสัยจาก 0 องศา ถึง 360 องศา โดยมุมเฟสนี้มีค่าเป็นสัดส่วนกับความเร็วผลต่าง ระหว่างความเร็วที่ต้องการ กับความเร็วที่แท้จริงของมอเตอร์ที่ได้เคลื่อนที่ไปยังตำแหน่งที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.8 เมื่อความเร็วที่ 2250 rpm และต่อโหลดขนาด 1.2 Nm



ภาพที่ 6.9 เมื่อความเร็วที่ 2250 rpm เปลี่ยนความเร็วอย่างฉับพลัน ไปเป็น 1000 rpm และต่อ โหลด 1.2 Nm

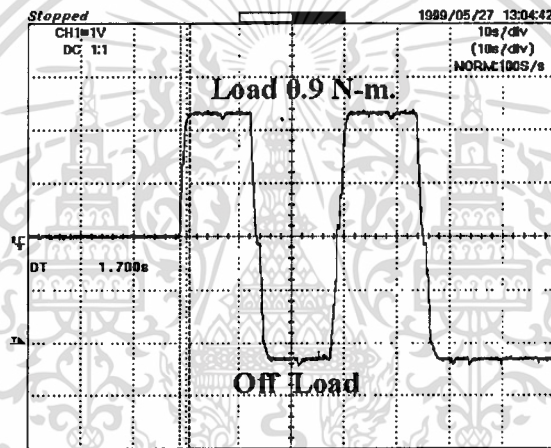
ตารางที่ 6.1 ผลการควบคุมแรงดันและกระแส ต่อความเร็วเมื่อจำกัดกระแสอินพุตขณะต่อโหลด

T(Nm)	0.00	0.3	0.6	0.9	1.2	T(Nm)	0.00	0.3	0.6	0.9	1.2
N(rpm)	2250	2250	2250	2250	2250	N(rpm)	1000	1000	1000	1000	1000
V_f (v)	150	178	193	205	218	V_f (v)	81	92	112	119	135
I_v	0.81	0.89	1.12	1.38	1.54	I_v	0.18	0.28	0.34	0.91	1.14

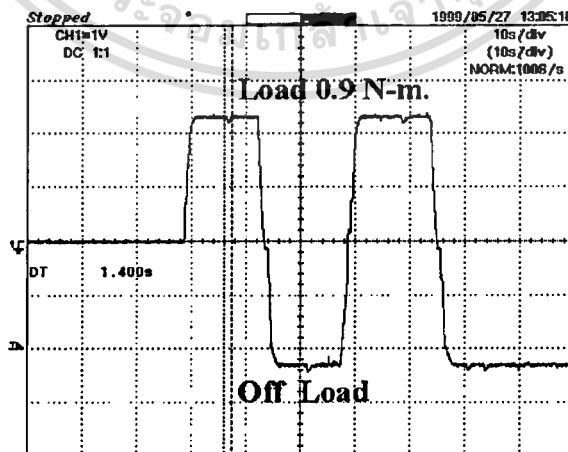
จากภาพที่ 6.7 แสดงผลตอบสนองในสภาวะชั่วคราวซึ่งเวลาเมื่อเปลี่ยนความเร็วลดลงอย่างฉับพลันจากความเร็ว 0 รอบต่อนาทีไปเป็น 2250 รอบต่อนาที และลดความเร็วลงไปเป็น 1000 รอบต่อนาที จะเห็นว่าใช้เวลาประมาณ 4.5 วินาที ในการเข้าสู่สภาวะคงตัว ส่วนภาพที่ 6.8 แสดงถึงสภาวะคงตัวเมื่อปรับความเร็วไว้ที่ 2250 รอบต่อนาที พร้อมกับต่อโหลดที่มีการเปลี่ยนแปลงโดยต่อเข้ากับ

โหลดขนาด 1.2 Nm จะเห็นว่าในการเข้าสู่สภาวะคงตัวทำได้อย่างรวดเร็ว และเมื่อเปลี่ยนความเร็วอย่างฉับพลันจาก 2250 รอบต่อนาที ไปเป็น 1000 รอบต่อนาที ขณะต่อกับโหลด 1.2 Nm ดังภาพที่ 6.9 จะเห็นว่าใช้เวลาประมาณ 1.5 วินาที ในการปรับเข้าสู่สภาวะคงตัว ในวงรอบปิด ผลจากการทดลองกำหนดความเร็วที่ต้องการเป็น 2250 รอบต่อนาที ไปเป็น 1000 รอบต่อนาที และจากตารางที่ 6.1 เป็นผลที่ได้จากการกำหนดกระแสอ้างอิงโดยจำกัดไว้ที่ไม่เกิน 2 แอมป์ โดยใช้ current probe ซึ่งสามารถที่จะกำหนดให้แรงดันและกระแส อ้างอิงให้เหมาะสมสอดคล้องกันกับในขณะต่ออยู่กับโหลด เพื่อให้ความเร็วถูกต้องตามต้องการ ดังตารางที่ 6.1

6.4 การทดลองผลตอบสนองของความเร็วต่อการกลับทิศทางการหมุนในขณะมีโหลด 0.9 N-m และขณะไม่มีโหลด

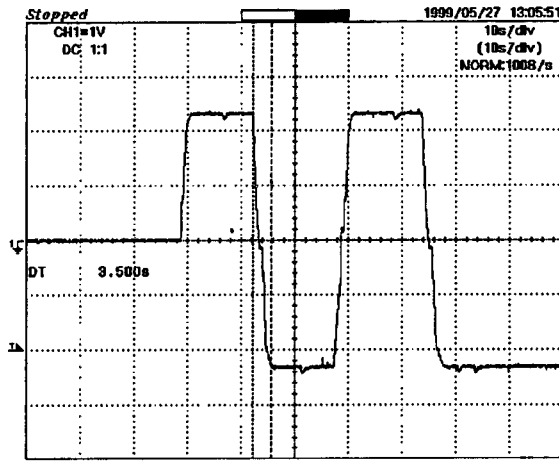


ภาพที่ 6.10 การวัดผลตอบสนองของเวลาขณะเข้าสู่ในสภาวะคงตัว



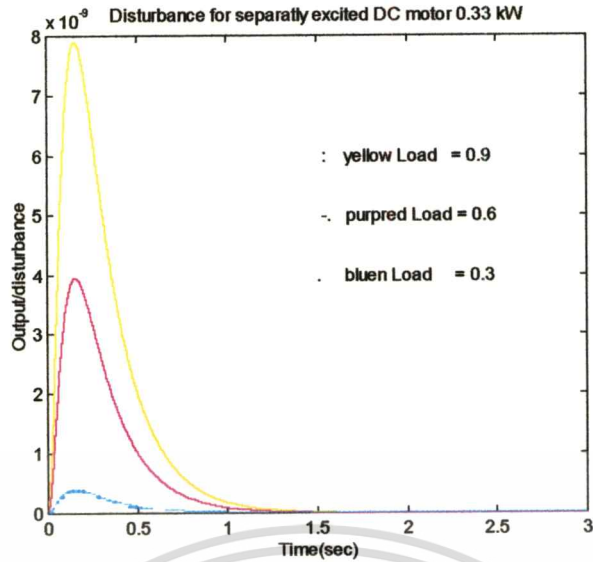
ภาพที่ 6.11 ผลตอบสนองของเวลาในสภาวะคงตัวเมื่อเพิ่มเกนที่ K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

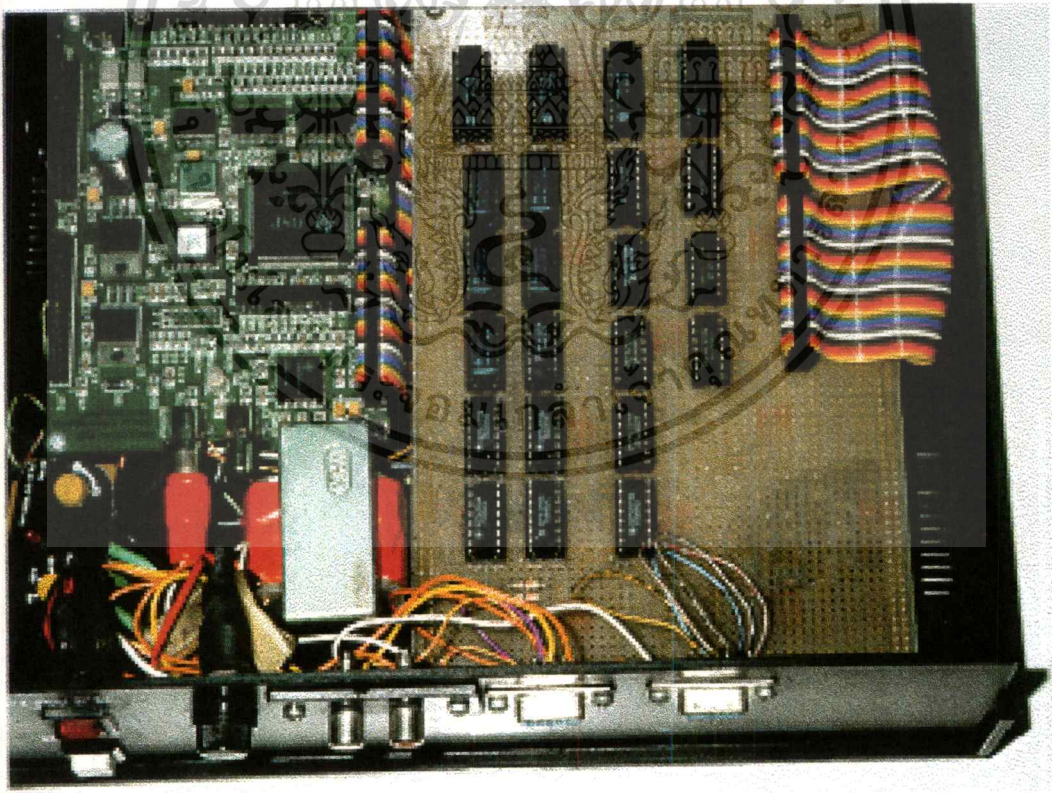


ภาพที่ 6.12 ผลตอบสนองของเวลาในสภาวะคงตัวเมื่อกลับทิศทางการหมุน

จากภาพที่ 6.10 เป็นผลที่ได้จากการวัดผลตอบสนองในสภาวะคงตัวขณะเริ่มหมุนในทิศทางตามเข็มนาฬิกาและต่อโหดขนาด $0.9 \text{ N}\cdot\text{m}$ จะเห็นว่าใช้เวลาเร็วกว่าเดิมเนื่องจากการป้อนกลับในรูปของการชดเชยกระแส ส่วนภาพที่ 6.11 นั้นจะเห็นว่าใช้เวลาเร็วขึ้นเนื่องจากผลของการเพิ่มค่าเกนที่ K ที่นำมาต่อเข้ากับตัวควบคุมเพื่อนำมาชดเชยในขณะมีโหด และไม่มีโหด ส่วนการกลับทิศทางการหมุนนั้นจะมีลักษณะเช่นเดียวกันเมื่อปรับความเร็วและต่อโหนดที่เท่ากันกับทิศทางการสภาวะปกติ แต่จะให้ผลตอบสนองของเวลาในสภาวะคงตัวช้ากว่าเดิม เนื่องจากการเปลี่ยนแปลงของทิศทางการสแกและแรงดันที่จ่ายให้กับวงจรรีเมเจอร์ และเนื่องจากช่วงที่มีการกลับทิศทางการหมุนนั้นจะทำให้กระแสและแรงดันของวงจรมอเตอร์ มีการเปลี่ยนแปลงเล็กน้อยเนื่องจากการหน่วงเวลาในรูปป้อนกลับทิศทางการหมุน แต่สามารถที่จะปรับให้ความเร็วเข้าสู่ในสภาวะคงตัวให้เร็วขึ้นได้โดยการเพิ่มค่าอัตราขยายเกนที่ K ที่ต่อของเค็ชเคดกับตัวควบคุม PIDA ซึ่งตัวควบคุม PIDA นี้จะเร่งให้ผลตอบสนองต่อการควบคุมตำแหน่งความเร็วตามได้ตามต้องการสอดคล้องกับภาระดังแสดงในภาพที่ 6.12 จากนั้นเพิ่มโหนดที่ค่าต่างๆ จาก 0.3 , 0.6 และ $0.9 \text{ N}\cdot\text{m}$ ตามลำดับดังแสดงในภาพที่ 6.13 จากการจำลองจะเห็นว่าให้ผลตอบสนองในการเข้าสู่สภาวะคงตัวได้อย่างรวดเร็วประมาณ 1.5 วินาที เช่นกัน ตามค่าความเร็วที่กำหนด และที่ความเร็วอ้างอิงจาก 0 รอบต่อวินาที ถึง 2250 รอบต่อวินาที จะเห็นได้ว่าตัวควบคุม PIDA ที่นำมาต่อกับกระบวนการที่เป็นมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นนั้นสามารถที่จะทำให้ผลตอบสนองต่อเวลาในการเข้าสู่ในสภาวะคงตัวทำได้เหมาะสมสอดคล้องกับสภาวะการใช้งาน

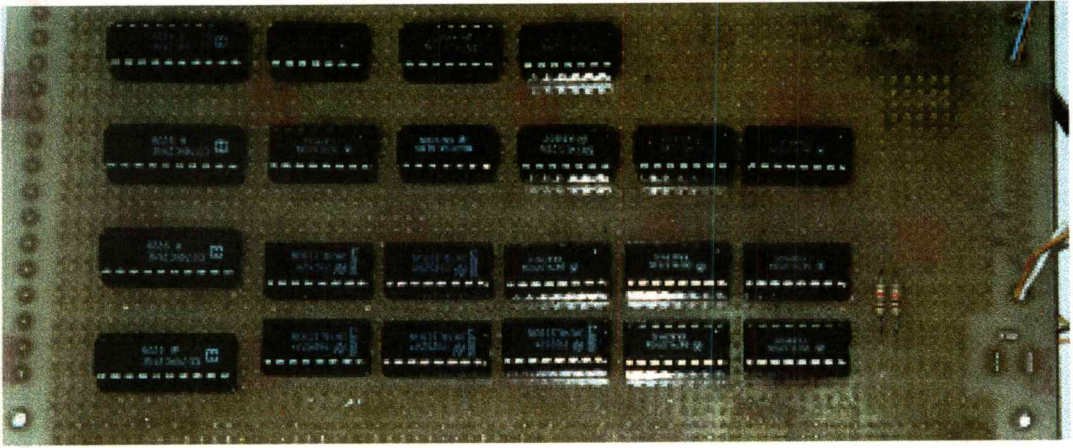


ภาพที่ 6.13 ผลการเขียนแบบที่เอาที่พูดต่อสัญญาณรบกวนค่าต่างๆ

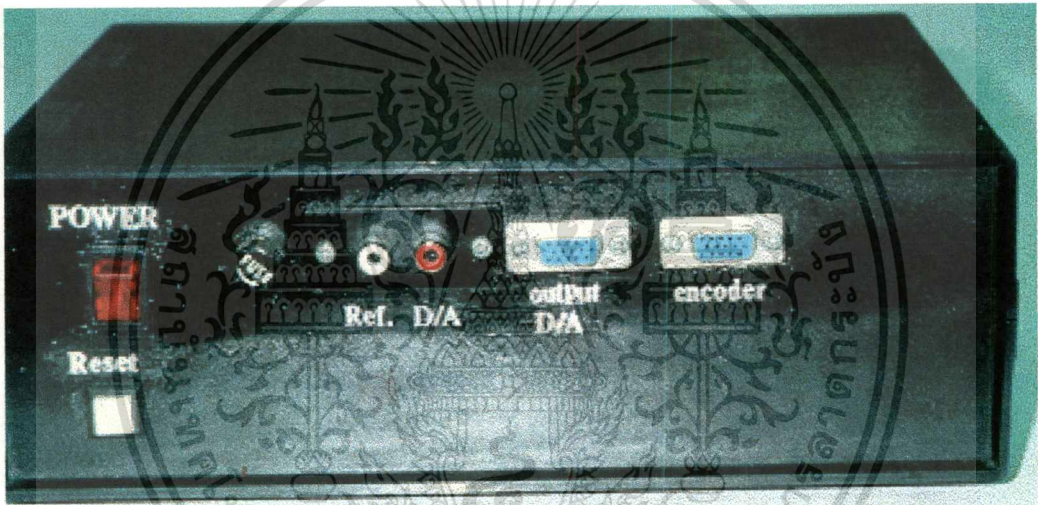


ภาพที่ 6.14 เครื่องต้นแบบที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.15 วงจรนับ 32 บิต วงจรตรวจจับทิศทาง การหมุน และวงจรอินเทอร์เฟซโมดูล



ภาพที่ 6.16 ภาพด้านหน้าของแผงทดลอง



เอกสารนี้ **ภาพที่ 6.17** ภาพระบบโดยรวม ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการวิจัยและข้อเสนอแนะ

สิ่งที่ต้องการในระบบควบคุมคือ ระบบต้องมีเสถียรภาพที่ดี และมีผลตอบสนองในสภาวะคงตัวที่มีความแม่นยำ ซึ่งการออกแบบจำเป็นจะต้องใช้ความรอบคอบ ระหว่างสองสิ่งที่กล่าวนี้ ดังนั้นจึงทำการวิเคราะห์ และพิจารณาให้ระบบเป็นเชิงเวลาไม่ต่อเนื่อง ดังภาพที่ 5.3.1 โดยสามารถเปลี่ยนข้อมูลได้ก็ต่อเมื่อขณะที่เวลาไม่ต่อเนื่อง ขณะเวลาเหล่านี้ระยะเวลาที่ซึ่งได้มีการวัดทางฟิสิกส์ คัดออกมา หรือเวลาที่ซึ่งได้อ่านข้อมูลออกมาจากส่วนของหน่วยความจำของไมโครคอมพิวเตอร์ เนื่องจากในการวัดอัตราเร็ว ใช้ช่วงเวลาในการสุ่ม 100 มิลลิวินาที อัตราเร็วผิดพลาดและข้อมูลของความเร็วจึงเกิดในลักษณะดิจิทัล 6 ครั้งต่อวินาที ดังนั้นคาบเวลาของการสุ่ม T จึงเท่ากับ 0.1 วินาที ซึ่งเอาต์พุตที่ได้จากวงจร ZOH จะทำให้เกิดมุมเฟสเคลื่อนที่อย่างเหมาะสม ซึ่งจะเป็นค่าคงที่เป็นช่วงๆ จากค่าอินพุตสุดท้ายไปจนถึงค่าอินพุตที่จะเข้ามาต่อไป

7.1 ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข

การออกแบบและการสร้างตัวควบคุม PIDA สำหรับควบคุมกระบวนการที่มีอันดับสาม และเมื่อนำมาประยุกต์ใช้งานกับการควบคุมตำแหน่งของมอเตอร์นั้นสิ่งที่ควรพิจารณาในการออกแบบเครื่องวัดอัตราเร็วมีลักษณะสำคัญดังนี้คือ ความละเอียดของซาวด์เอ็นโคเดอร์ และช่วงเวลาของการวัด ซึ่งการออกแบบที่เกี่ยวข้องกับการจัดสเกล เวลาจะถูกจำกัดโดยความแม่นยำเชิงมุมของแผ่นดิสก์ที่ตัวเอ็นโคเดอร์ และอัตราเร็วของไมโครคอมพิวเตอร์ ความถี่ของขบวนพัลส์จากซาวด์เอ็นโคเดอร์ จะเปลี่ยนเป็นอัตราเร็วเชิงมุมโดยอาศัยการแก้ไขสเกลเวลาให้ถูกต้องและการอินทิเกรตด้วยไมโครคอมพิวเตอร์ โดยที่เราสามารถเพิ่มความแม่นยำให้สูงขึ้นได้ด้วยการแบ่งคาบระหว่างพัลส์แต่ละพัลส์ออกเป็น 4096 หน่วย แล้วทำการเพิ่มบิตของวงจรนับขึ้นเพียงหนึ่งบิตก็จะทำให้ความแม่นยำดีขึ้นมาก ส่วนช่วงเวลาระหว่างการตรวจสอบของสัญญาณอินพุตเท่ากับ 30 ไมโครวินาที ตามทฤษฎีสุ่มความถี่อินพุตสูงสุดต้องมีคาบมากกว่า 2×30 ไมโครวินาที ที่ความถี่ต้องมีค่า 20 กิโลเฮิร์ตซ์ เนื่องจากอัตราการสุ่มข้อมูลที่อ่านจากอินพุตพอร์ต จะถูกจำกัดด้วยอัตราเร็วของไมโครคอมพิวเตอร์ เครื่องวัดอัตราเร็วเชิงดิจิทัลนี้ไม่สามารถวัดอัตราเร็วเชิงมุมโดยตรงไปตรงมาได้สูงกว่า 2000 รอบต่อนาที ตัวประกอบสองตัวนี้มีผลกระทบต่อความแม่นยำระหว่างการวัดช่วงเวลาเกี่ยวกับจำนวนพัลส์ต่อรอบที่เกิดขึ้นจากซาวด์เอ็นโคเดอร์ หรือการตรวจสอบคาบเวลาของไมโครคอมพิวเตอร์ และอัตราเร็วสูงสุดของมอเตอร์ที่จะวัดด้วย นอกจากนั้นการทำงานของระบบจะดีขึ้น ถ้าใช้ส่วนความจำที่สามารถเรียกใช้ได้เร็ว ซึ่งใช้ตัวประมวลสัญญาณผลเชิงเลขควบคุมในการทำงานของอุปกรณ์อิเล็กทรอนิกส์กำลังได้เป็นอย่างดี และใช้ซาวด์เอ็นโคเดอร์ที่มีความละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูงๆ ซึ่งถ้าอัตราเร็วของพัลส์หนึ่งลดลงค่าผิดพลาดในการควบคุมก็จะลดลงตามไปด้วย นอกจากนี้การทำงานของระบบจะดียิ่งขึ้นอยู่กับการทำงานที่แม่นยำของวงจรมอเตอร์โฮลด์ “Zero Order Hold” (ZOH) และวงจรมอเตอร์โฮลด์หรือวงจรมอเตอร์โฮลด์ที่กล่าวถึงข้างต้นด้วย

7.2 ข้อเสนอแนะ และการพัฒนาการทำงานวิจัยต่อไป

ตัวประมวลผลเชิงเลข จะทำการควบคุมในลักษณะอนุกรม หรือเรียงต่อกัน คือถ้ามีงานหลายงานที่จะต้องทำพร้อมกัน หน่วยประมวลผลเชิงเลข จะทำได้ทีละงาน ซึ่งจะต้องใช้วิธีการแบ่งเวลาให้เหมาะสม โดยที่ควรพิจารณาเวลาของการสุ่ม (Sampling time = ช่วงเวลาที่ซึ่งวงป้อนกลับแต่ละวงผลัดกันเข้าทำงาน) อย่างรอบครอบลักษณะสำคัญของการใช้ตัวประมวลผลสัญญาณเชิงเลข เข้าควบคุมมอเตอร์คือ ระบบฮาร์ดแวร์จะช่วยให้งานของระบบง่ายขึ้น ทำงานได้เป็นที่น่าเชื่อถือ มีราคาถูกลง และเนื่องจากการควบคุมเป็นเชิงเลข จึงช่วยป้องกันสัญญาณรบกวนจากภายนอกและสัญญาณที่เกิดขึ้นภายในระบบควบคุมเอง โดยเฉพาะอย่างยิ่งจากสัญญาณชั่วคราวในการทำงานของวงจรมอเตอร์โฮลด์ที่กล่าวถึงข้างต้น ระบบซอฟต์แวร์ที่ใช้ในการควบคุมนั้นเป็นการง่ายที่จะเปลี่ยนแปลงและปรับปรุงขั้นตอนการทำงานให้ดียิ่งขึ้นโดยไม่ต้องเปลี่ยนระบบฮาร์ดแวร์ นอกจากนี้ยังสามารถเปลี่ยนแปลงโครงสร้างและตัวแปรของระบบควบคุมให้ได้ในเวลาจริง (Real Time) เป็นผลให้สามารถปรับปรุงการควบคุมให้เหมาะสมกับลักษณะคุณสมบัติของมอเตอร์ด้วย ตัวประมวลผลสัญญาณเชิงเลขนั้นสามารถทำการคำนวณที่ยุ่ยาก และตัดสินใจขั้นต้นต่างๆ ในการนำเอาทฤษฎีของการควบคุมแบบต่างๆ มาใช้ ทั้งยังตรวจสอบระบบให้ถูกต้องก็สามารถทำได้ด้วยซอฟต์แวร์ ประกอบกับเทคโนโลยีได้ก้าวหน้าอยู่เสมอ ทำให้มีระบบฮาร์ดแวร์ใหม่ๆ เพิ่มขึ้นและซอฟต์แวร์สามารถพัฒนาให้ใช้ได้กับภาษาระดับสูงๆ

จากการวิจัยพบว่าตัวควบคุม PIDA ที่นำมาใช้ควบคุมระบบควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นนั้นยังมีข้อจำกัดคือสามารถใช้ได้กับกระบวนการที่เป็นอันดับสามเท่านั้น ซึ่งก็เท่ากับมีการยืดหยุ่นในการควบคุมกระบวนการน้อย ดังนั้นผู้วิจัยเห็นว่าในโอกาสต่อไปจะได้หาวิธีการที่จะทำให้ตัวควบคุม PIDA นั้นสามารถใช้ได้กับกระบวนการที่มีอันดับสูงกว่าอันดับสาม เพื่อนำไปประยุกต์ใช้งานกับกระบวนการชนิดอื่นๆ เช่นการควบคุมระดับ หรือกระบวนการที่ซับซ้อนในโรงงานอุตสาหกรรมที่มีอันดับสูงๆต่อไป ดังนั้นจึงได้ศึกษาและจำลองแบบกระบวนการที่เป็นมอเตอร์กระแสสลับชนิดเหนี่ยวนำ โดยใช้ตัวควบคุมดังกล่าวมาทำการจำลองระบบการควบคุมดังใน [3] ด้วยการต่อตัวควบคุมเป็นแบบป้อนไปข้างหน้าซึ่งจะได้แสดงการจำลองและผลการจำลองระบบควบคุมในหัวข้อ 7.3 ดังต่อไปนี้

7.3 การออกแบบตัวควบคุม PIDA สำหรับควบคุมตำแหน่งมอเตอร์เหนี่ยวนำกระแสสลับ

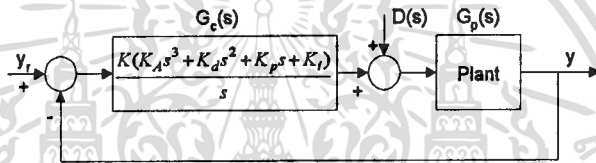
พิจารณาโครงสร้างในวงรอบปิดแบบทั่วไป เมื่อมีสัญญาณป้อนกลับเป็นหนึ่งหน่วย และตัวเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปเผยแพร่บนการคำ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุม PIDA ถูกกำหนดให้เป็นดังสมการที่ (7.1) หรือ (4.1) ดังนี้

$$G_c(s) = K_p + \frac{K_I}{s} + \frac{K_D s}{(s+d)} + \frac{K_A s^2}{(s+d)(s+e)} \quad (7.1)$$

โดยที่ค่า a, b, z และ d, e เป็นซีโร และ โพล ของตัวควบคุมโดยเหตุที่ว่า $a, b, z \ll d, e$. พิจารณาที่โพล d, e มีค่าเล็กน้อยมากจึงไม่นำมาคิด ดังนั้นถ้าจัดรูปสมการที่ (7.1) ใหม่ โดยมีการต่อแอสเคดเคนต์ K เข้าไปที่ตัวควบคุม และเพื่อต้องการพิจารณาค่า K_A, K_D, K_p และ K_I ได้อย่างอิสระต่อกัน ดังภาพที่ 7.1 จึงจัดสมการใหม่ได้เป็น

$$G_c(s) = K \left[\frac{K_A s^3 + K_D s^2 + K_p s + K_I}{s} \right] \quad (7.2)$$

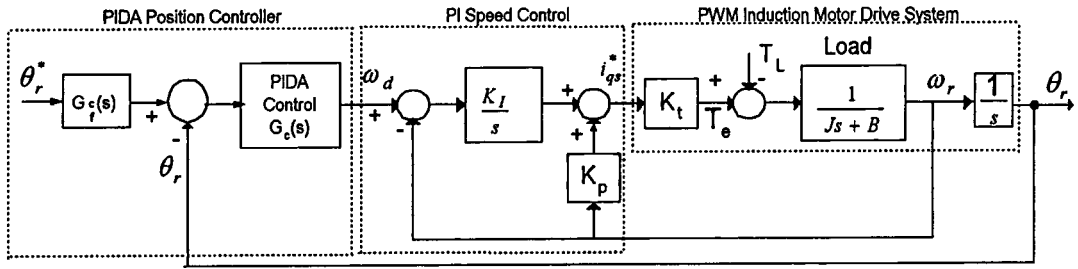


ภาพที่ 7.1 โครงสร้างของระบบควบคุม PIDA ในวงปิดแบบทั่วไปที่จัดใหม่

7.3.1 (การออกแบบตัวควบคุม PIDA และ PIDA โรบัสต์)

การออกแบบตัวควบคุม PIDA ด้วยวิธี Dominant Roots เป็นการประยุกต์ใช้การวางตำแหน่งค่ารากตามขอบเขต ใน s -plane เช่นเดียวกันกับในบทที่ 4 ภาพที่ 4.2 เมื่อโพลของกระบวนการค่ามากที่สุดไม่ได้อยู่ที่ตรงจุดกำเนิดแต่มีค่าน้อย $-\xi\omega_n$ ดังนั้น จะแทนตำแหน่งของโพล ทั้งสองให้ครอบตำแหน่งค่ารากหลัก ดังได้กล่าวไว้ในเอกสารอ้างอิง [12,13 และ 2] ซึ่งยิ่งทำให้การพิจารณาการออกแบบง่ายกว่าเดิมเมื่อใช้ตัวควบคุมตามสมการที่ (7.2) ส่วนการออกแบบตัวควบคุม PIDA โรบัสต์นั้นเป็นการออกแบบตัวควบคุม PIDA เพื่อให้ระบบควบคุมมีเสถียรภาพดียิ่งขึ้น ในกรณีที่มีการรบกวนจากภายในระบบควบคุม (Internal Disturbance) เช่นค่าพารามิเตอร์เปลี่ยนแปลง (Uncertain) หรือเป็นการรบกวนที่เกิดจากภายนอก (External Disturbance) ซึ่งระบบควบคุมแบบที่ใช้กันอยู่ทั่วไปอาจจะกำจัดผลของการรบกวนได้ไม่ดีพอ หรือใช้เวลานานกว่าผลของการรบกวนจะหมดไป ดังนั้นจึงได้นำเสนอการออกแบบตัวควบคุม PIDA ที่มีการต่อตัวควบคุมเป็นแบบป้อนไปข้างหน้า (Feed forward) ซึ่งสามารถวิเคราะห์เสถียรภาพในโดเมนความถี่ และในโดเมนเวลาได้อย่างง่าย แสดงดังภาพที่ 7.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 7.2 โค้ดโปรแกรมโดยรวมของระบบควบคุม PIDA โรบัสต์ สำหรับมอเตอร์เหนี่ยวนำ

7.3.1.1 (แบบจำลองการควบคุมความเร็ว และตำแหน่งของมอเตอร์เหนี่ยวนำ)

ในการนำเสนอการประยุกต์ใช้งาน เมื่อพิจารณาถึงวิธีการออกแบบ กับปัญหาจริง ดังนั้นเมื่อทำการออกแบบตัวควบคุม PIDA จะทำให้ง่ายต่อการควบคุมทั้งความเร็วและตำแหน่งที่โรเตอร์ของมอเตอร์ การควบคุมแนวใหม่นี้มีการนำมาสร้าง เพื่อพัฒนาใช้กับการควบคุมอัตราเร็วและตำแหน่ง ของมอเตอร์กระแสตรงแบบแยกตัวกระตุ้นซึ่งได้กล่าวไว้อย่างละเอียดในวิทยานิพนธ์ฉบับนี้แล้ว ซึ่งในบทนี้จะเป็นการนำตัวควบคุมดังกล่าวมาประยุกต์ใช้งานกับแบบจำลองของมอเตอร์เหนี่ยวนำกระแสสลับ เพื่อวิเคราะห์ว่าตัวควบคุมดังกล่าวมีความยืดหยุ่นในการนำไปประยุกต์ใช้งานกับกระบวนการชนิดต่างๆ ได้อย่างเหมาะสม ซึ่งแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำที่ใช้จำลองเป็นชนิด 3 เฟส ต่อแบบ Y และมี 4 โพล ขนาด 2.2 Kw 50 Hz 380v/5.4A โดยที่โครงสร้างนั้นถูกออกแบบให้เป็นเชิงเส้น [12] ดังนั้นจึงแสดงแบบจำลองของระบบควบคุมของมอเตอร์ได้ดังภาพที่ 4.3 ซึ่งความเร็วถูกควบคุมโดยตัวควบคุม PI ส่วนการควบคุมตำแหน่งใช้ตัวควบคุม PIDA ทำให้ต้องกำหนดค่าตัวควบคุม K_A, K_D, K_p และ K_I ใหม่เป็น K_1, K_2, K_3 และ K_0 ตามลำดับ ดังนั้นฟังก์ชันถ่ายโอนของการควบคุมความเร็วมอเตอร์เหนี่ยวนำจากภาพที่ที่ 7.3 และให้ $T_L(s) = 0$ จะได้เป็นดังนี้

$$\left. \frac{\omega_d(s)}{\omega_r(s)} \right|_{T_L(s)=0} = \frac{K_I K_t}{J s^2 + (B + K_p K_t) s + K_I K_t} \quad (7.3)$$

เขียนอยู่ในรูปของ สมการอันดับสองได้เป็น

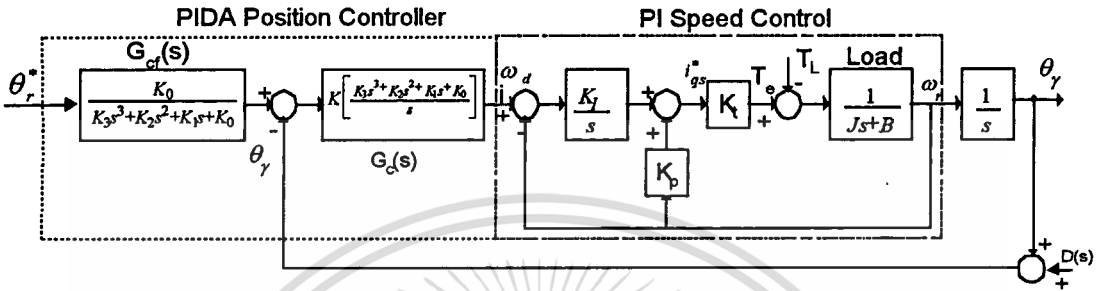
$$\frac{\Delta}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (7.4)$$

$$\text{เมื่อกำหนดให้} \quad \zeta = \frac{B + K_p K_t}{2\sqrt{K_I K_t}}, \quad \omega_n = \left(\frac{K_I K_t}{J} \right)^{1/2} \quad (7.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (7.5) ต้องการให้ผลตอบสนองของเปอร์เซ็นต์โอเวอร์ชูต น้อยกว่า 5% โดยอัตราส่วนของการหน่วงที่ตัวมอเตอร์เท่ากับ $\xi = 0.68$ หรือประมาณ $\cong 0.7$ ดังนั้นสามารถหาผลตอบสนองในเชิงเวลาขั้นหนึ่งหน่วยของสมการ (7.5) ได้ดังนี้

$$\omega_r(t) = 1 - e^{-\omega_n t} (1 + \omega_n t) \quad (7.6)$$



ภาพที่ 7.3 การควบคุมมอเตอร์เหนี่ยวนำกระแสสลับด้วยตัวควบคุม PIDA โรบัสต์ โดยการชดเชยแบบป้อนไปข้างหน้า

ซึ่งสมการที่ (7.5) เป็นตัวบ่งชี้ว่าสถานะคงตัวคู่เข้ามีความผิดพลาด เป็นศูนย์ ซึ่งค่าพารามิเตอร์ที่ใช้ของมอเตอร์ ในกรณีที่ใช้ความเร็วเชิงมุม $\omega_r = 125.56 \text{ rad/s}$, $K_t = 0.4636$, $J = 0.0039 \text{ kg} \times \text{m}$, $B = 0.098$ ค่าเวลาคงที่ของมอเตอร์มีค่า 0.3 วินาที และ $\omega_n = 12.963$ และสามารถหาค่าพารามิเตอร์ของตัวควบคุม PI ได้จากสมการ (7.7) เช่นกันเพื่อหาค่า K_i , K_p ซึ่งหาได้จากสมการดังนี้

$$K_i = J\omega_r^2 / K_t, \quad K_p = (2J\omega_r - B) / K_t \quad (7.7)$$

จะได้ $K_i = 1.4136$, $K_p = 0.0067$ และเมื่อนำตัวเซ็นเซอร์มาต่ออนุกรมกับแบบจำลองของมอเตอร์เพื่อนำไปต่อตัวควบคุม PIDA สำหรับการควบคุมตำแหน่งของแกนเพลลาของมอเตอร์ที่เคลื่อนตำแหน่งไป จะได้ฟังก์ชันถ่ายโอนของแบบจำลองมอเตอร์เหนี่ยวนำใหม่ได้เป็นดังสมการที่ (7.8)

$$G_p(s) = \frac{\theta_r(s)}{\omega_r(s)} \Big|_{\tau_r(\omega)=0} = \frac{K_t K_i}{s(Js^2 + (B + K_t K_i)s + K_t K_i)} \quad (7.8)$$

จากสมการที่ (7.8) เมื่อแทนค่าพารามิเตอร์ลงในสมการของ $G_p(s)$ ของแบบจำลองมอเตอร์เหนี่ยวนำจะได้ฟังก์ชันถ่ายโอนของการควบคุมความเร็ว PI ได้ดังสมการที่ (7.9) ดังนี้

$$G_p(s) = \frac{168.037}{s(s^2 + 25.962s + 168.037)} \quad (7.9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำสมการคุณลักษณะและค่าพารามิเตอร์ต่างๆ มาหาค่า K_3 , K_2 , K_1 , และ K_0 จะได้ดังนี้ $K_3 = 21.6768$, $K_2 = 920.8518$, $K_1 = 12703.7$ และ $K_0 = 42345.679$ ดังนั้นฟังก์ชันถ่ายโอนในวงรอบเปิดของ $G(s)$ ที่ตัวควบคุม PI กับแบบจำลองของกระบวนการได้ในรูปของ $G(s) = G_c(s) G_p(s)$ เป็น

$$G(s) = \frac{21.6768s^3 + 920.8518s^2 + 12703.7s + 42345.679}{s^3 + 25.962s^2 + 168.037s} \quad (7.10)$$

เมื่อต่อตัวควบคุม PID เป็นแบบป้อนไปข้างหน้า (Forward control) หรืออาจเรียกการต่อตัวควบคุมในลักษณะนี้ว่า “การควบคุม PIDA โรบัสต์” ดังภาพที่ 7.3 ซึ่งเป็นวิธีการต่อที่ทำให้ระบบควบคุมมีเสถียรภาพสัมพัทธ์ที่ดี และมีความคงทนมากยิ่งขึ้น ดังนั้นเขียนฟังก์ชันถ่ายโอนตัวควบคุมแบบป้อนไปข้างหน้าได้ดังนี้

$$G_c = \frac{K_0}{K_3s^3 + K_2s^2 + K_1s + K_0} \quad (7.11)$$

ดังนั้นเขียนฟังก์ชันถ่ายโอนในวงรอบปิด ที่ได้จากการออกแบบเป็นดังนี้

$$\frac{\theta_r(s)}{\theta_i(s)} = \left[\frac{KK_0}{s^4 + (25.962 + KK_3)s^3 + (168.037 + KK_2)s^2 + KK_1 + KK_0} \right] \quad (7.12)$$

และจากสมการ (7.12) จะได้ฟังก์ชันของตัวควบคุมในวงรอบปิดดังนี้ เมื่อ $K = 1$

$$\frac{\theta_r(s)}{\theta_i(s)} = \left[\frac{7.116 \times 10^4}{s^4 + 3668s^3 + (1.549 \times 10^4)s^2 + (2.135 \times 10^4)s + 7.116 \times 10^4} \right] \quad (7.13)$$

ดังนั้นเพื่อความชัดเจนในการพิสูจน์ว่าระบบควบคุมที่ได้ทำการออกแบบนั้นมีเสถียรภาพมากเพียงใดด้วยฟังก์ชันถ่ายโอนในวงรอบปิดดังสมการ (7.13) โดยพิจารณาทั้งโพลไดอะแกรม ดังภาพที่ 7.4 และเส้นโค้งไนควิสต์ในวงรอบเปิดดังสมการ (7.10) ได้ดังภาพที่ 7.5 จะเห็นว่าเส้นคอนทัวร์ไนควิสต์ไม่ได้ครอบคลุมโพลของฟังก์ชันในวงรอบปิด และก็ไม่ได้ครอบคลุมที่ -1 ดังนั้นระบบจึงมีเสถียรภาพ ทำให้ระบบควบคุมมีการตอบสนองระหว่างอินพุตกับเอาต์พุต ด้วยฟังก์ชันขั้นหนึ่งหน่วยในวงรอบปิด จะได้ดังภาพที่ 7.6 จะเห็นว่าระบบควบคุมมีความนุ่มนวลในการเข้าสู่สภาวะคงตัว และให้ผลตอบสนองตรงตามคุณสมบัติหรือจะพิจารณาในโดเมนซ์ของแซดทรานส์ฟอร์มจากสมการที่ (7.13) ให้อยู่ในรูปของ $H(z) = \frac{\theta_r(s)}{\theta_i(s)}$ ได้ด้วยการใช้การแปลงไบลิเนียร์จาก

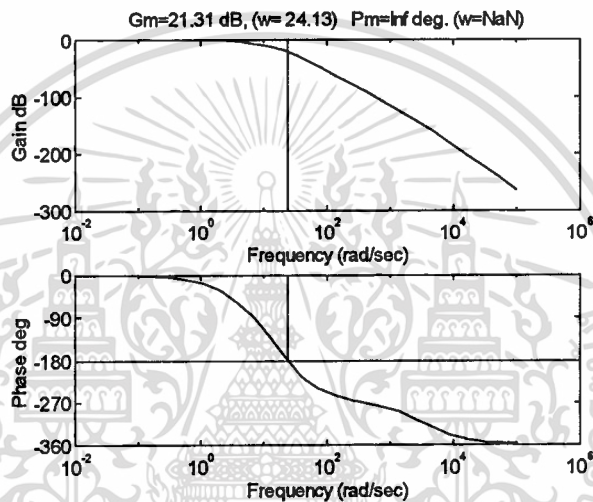
โปรแกรม Matlab [19] ได้ดังนี้

$$H(z) = \frac{\theta_r(s)}{\theta_r^*(s)} = \frac{0.0541 + 0.2164z^{-1} + 0.3245z^{-2} + 0.2164z^{-3} + 0.0541z^{-4}}{1 - 3.7212z^{-1} + 5.1830z^{-2} - 3.2021z^{-3} + 0.7402z^{-4}} \quad (7.14)$$

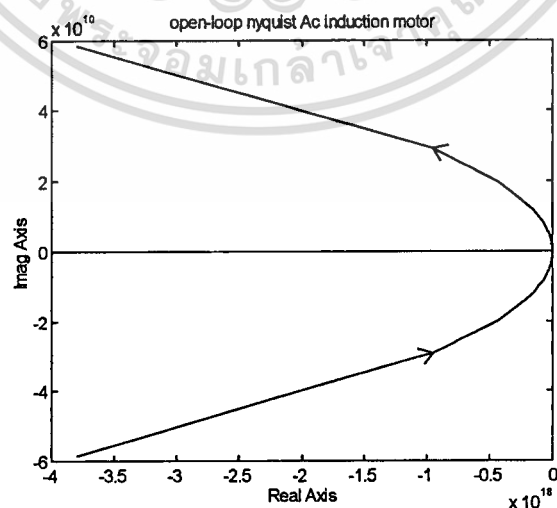
เนื่องจากฟังก์ชันถ่ายโอนเป็นระบบอันดับสี่สามารถเขียน $H(z)$ ได้ใหม่เป็นดังนี้

$$H(z) = \frac{0.5913(1 + 2.007z^{-1} + 1.0007z^{-2})(1 + 1.9993z^{-1} + 0.9993z^{-2})}{(1 - 1.9089z^{-1} + 0.9112z^{-2})(1 - 1.8123z^{-1} + 0.8123z^{-2})} \quad (7.15)$$

ซึ่งทั้งสมการที่ (7.14) และ (7.15) สามารถนำไปวิเคราะห์ในเชิงดิจิทัลได้เช่นกัน



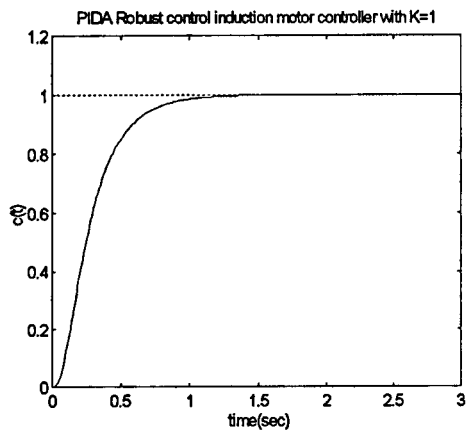
ภาพที่ 7.4 โพลในวงรอบปิดของตัวควบคุม PIDA โรบัสต์



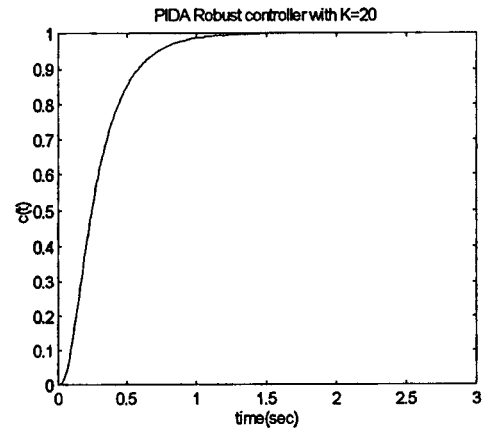
ภาพที่ 7.5 เส้นคอนทัวร์ในควิสต์ในวงรอบเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

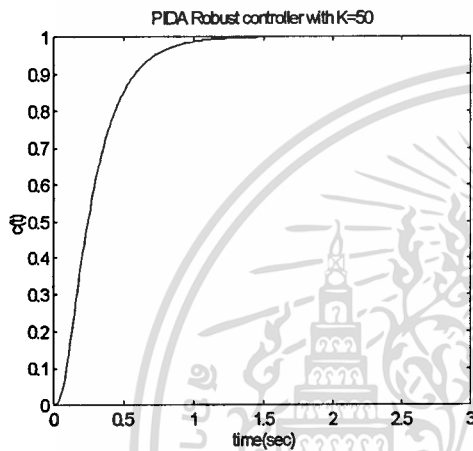
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



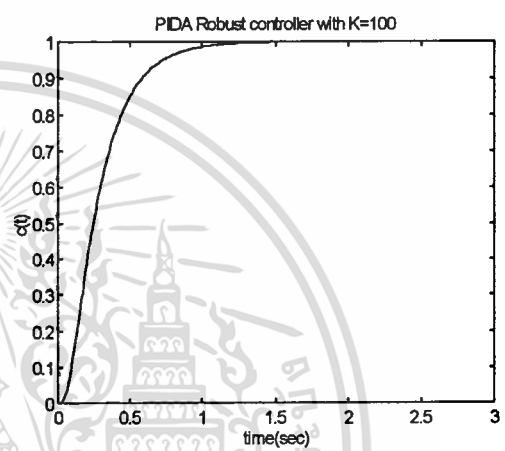
(ก)



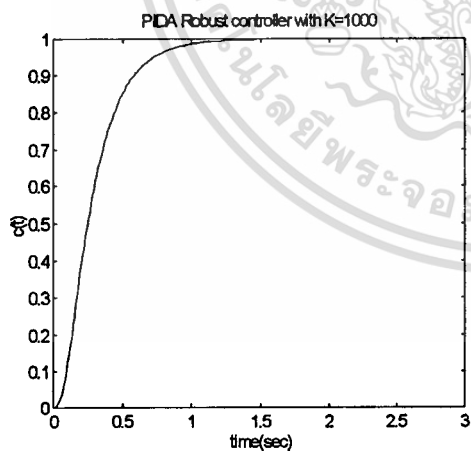
(ข)



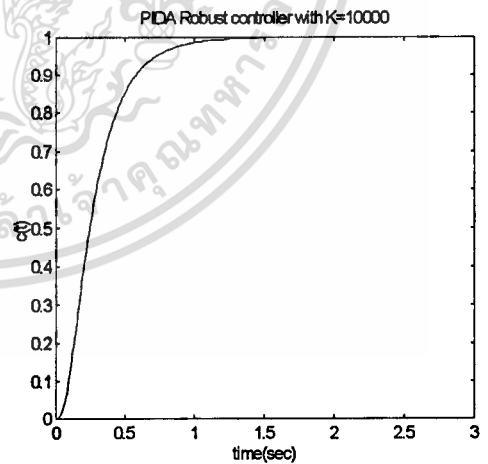
(ค)



(ง)



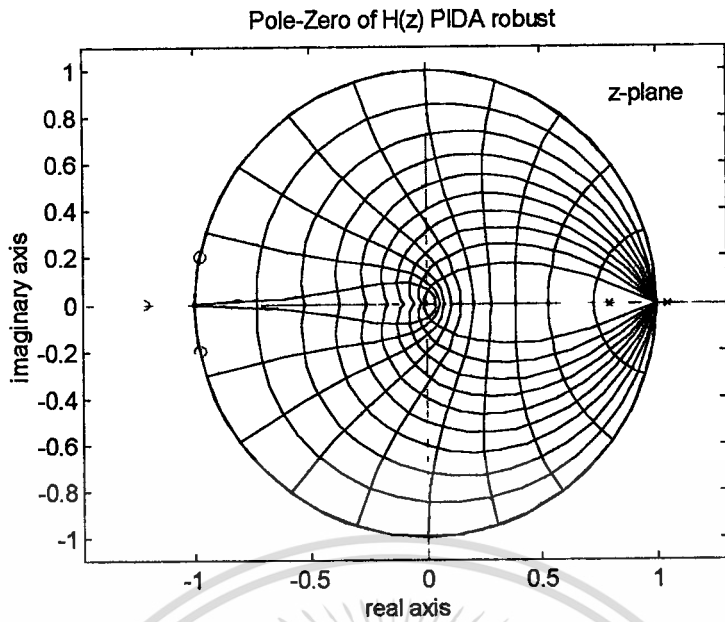
(จ)



(ฉ)

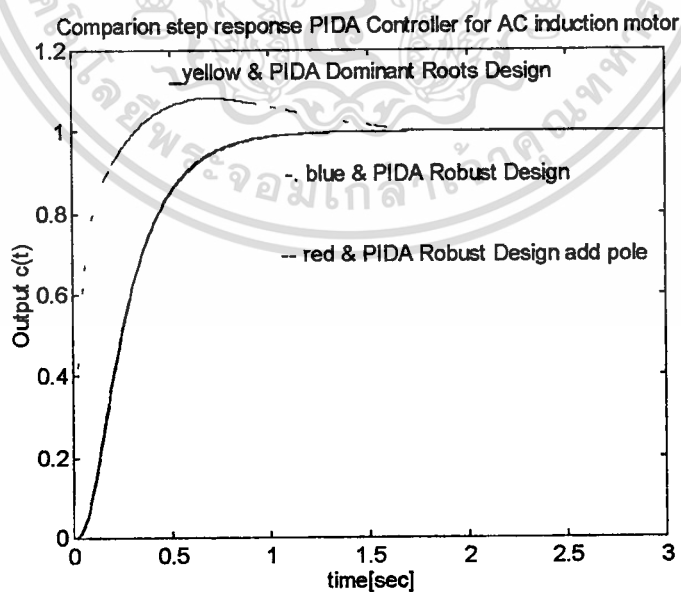
ภาพที่ 7.6 ผลการตอบสนองของฟังก์ชันขั้นหนึ่งหน่วย เมื่อเพิ่มค่าอัตราขยาย เกณฑ์ $K = 1, 20, 50, 100, 1000$ และ 10000 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 7.7 ผลการวิเคราะห์เสถียรภาพโดยใช้รูตโรกัส

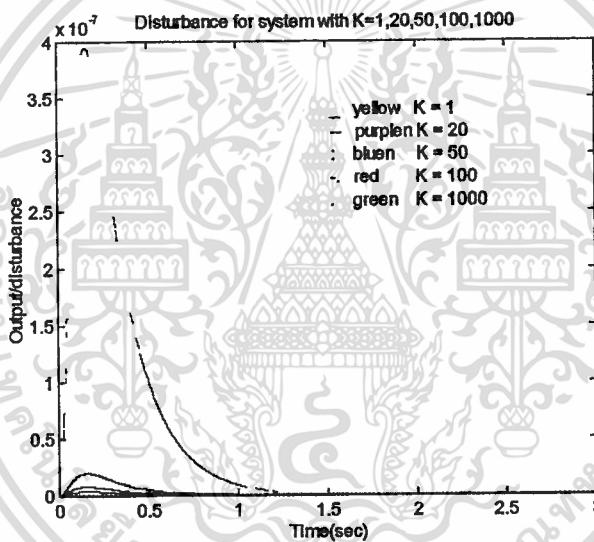
ดังนั้นสามารถวิเคราะห์เสถียรภาพของระบบโดยใช้วิธีการของรูตโรกัสในระนาบแซคเพลนดังแสดงในภาพที่ 7.7 ซึ่งจะเห็นว่าระบบควบคุมมีเสถียรภาพในทุกกรณี และเมื่อเปรียบเทียบผลการตอบสนองของฟังก์ชันขั้นหนึ่งหน่วยของตัวควบคุม PIDA กับตัวควบคุมที่มีการต่อแบบโรบัสต์ แสดงได้ดังภาพที่ 7.8 ดังนี้



ภาพที่ 7.8 เปรียบเทียบผลการตอบสนองของฟังก์ชันขั้นหนึ่งหน่วยของตัวควบคุม PIDA กับ

เอกสารนี้เป็นเอกสารตัวควบคุมPIDA โรบัสต์ ซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 7.8 ที่ค่า $K = 1$ นั้นตัวควบคุม PIDA โรบัสต์มีคุณสมบัติดังนี้คือ $P.O = -0.1163\%$, $T_p = 0.001$ วินาที, $T_s \leq 1.4$ วินาที ทั้งกรณีการต่อแบบ PIDA โรบัสต์ และ แบบ PIDA โรบัสต์ที่มีการเพิ่มตัวกรองอันดับสองเข้าไปในระบบควบคุมวงปิด ส่วนเส้นกราฟอีกเส้นหนึ่งเป็นของตัวควบคุม PIDA ที่ใช้จำลองตามสมการที่ (7.10) ซึ่งจะมี $P.O \geq 5\%$ แต่ค่าไรท์ไทม์ Rise Time (t_r) นั้นน้อยกว่ากรณีที่เป็นแบบโรบัสต์ และเมื่อต้องการลดโอเวอร์ชูต หรือ P.O ให้ทำการปรับเกณฑ์ K ขึ้นเล็กน้อย ดังนั้นสรุปได้ว่าการใช้งานตัวควบคุม PIDA โดยการต่อตัวควบคุมแบบป้อนไปข้างหน้า นั้นมีความเหมาะสมกับการนำไปประยุกต์ใช้งานกับการควบคุมมอเตอร์เหนี่ยวนำกระแสสลับทุกประการ เพื่อการควบคุมตำแหน่งที่แม่นยำและนุ่มนวล ที่ซึ่งให้ผลตอบสนองของฟังก์ชันขั้นหนึ่งหน่วย ด้วยการเพิ่มค่าเกณฑ์ K สูงๆ แล้วยังเหมือนกับกรณีที่มีค่าเกณฑ์ K น้อยๆ ดังภาพที่ 7.6 ส่วนภาพที่ 7.9 เป็นการพิจารณาผลตอบสนองของเอาต์พุตต่อสัญญาณรบกวนเมื่อเพิ่มค่าเกณฑ์ K ดังภาพดังกล่าว



ภาพที่ 7.9 เปรียบเทียบผลการตอบสนองของเอาต์พุตต่อสัญญาณรบกวนเมื่อเพิ่มค่าเกณฑ์ K ของตัวควบคุม PIDA โรบัสต์

จากการใช้ตัวควบคุม PIDA ตามสมการที่ (7.2) ซึ่งใช้ออกแบบสำหรับการควบคุมตำแหน่งของมอเตอร์เหนี่ยวนำกระแสสลับ ที่ให้ผลตอบสนองในสภาวะคงตัว และผลตอบสนองในสภาวะชั่วคราว ได้อย่างรวดเร็วตามผลที่ได้รับจากการเขียนแบบดังแสดงในภาพที่ 7.6 และ 7.8 โดยที่วิธีการออกแบบตัวควบคุม PIDA แบบนี้ทำให้วิเคราะห์ผลง่ายเพียงแต่เพิ่มค่าเกณฑ์ที่ต่อแอสต์กับตัวควบคุมบวกกับค่าราคในวงรอบปิด ซึ่งเป็นส่วนที่สำคัญในการออกแบบระบบควบคุม ดังนั้นการออกแบบตัวควบคุม PIDA จึงเป็นอีกก้าวหนึ่งของผู้ที่ต้องการออกแบบและวิเคราะห์ระบบที่สามารถพัฒนาได้ง่ายตามต้องการ สำหรับการควบคุมกระบวนการที่เป็นอันดับสามที่มีชนิดมากกว่า 1 ซึ่งสามารถสร้างได้ง่ายขึ้น การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร “ การสร้างสัญญาณด้วยสัญญาณเสียงที่มีการสูญหายไปขึ้นมาใหม่” คอมพิวเตอร์ ISSN 0125-2100 ปีที่ 21, ฉบับที่ 114, กรกฎาคม-สิงหาคม 2538. หน้า 49-55.
- [2] เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร, สุพรรณ กุลพานิชย์ “ การสร้างตัวควบคุม PIDA สำหรับควบคุมมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น” วิศวกรรมลาดกระบัง ปีที่ 14, ฉบับที่ 2, มิถุนายน 2541 หน้า 49-57.
- [3] เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร, ชัยณรงค์ ไพโรพถกษ, กฤษชัย วิถีพานิช, “การออกแบบตัวควบคุม PIDA สำหรับการควบคุมตำแหน่งมอเตอร์เหนี่ยวนำกระแสสลับ”การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 21 วันที่ 12-13, พฤศจิกายน 2541. CT016 หน้า 413-416 ณ. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ถ.สุขสวัสดิ์ 48 เขตทุ่งครุ กรุงเทพมหานคร
- [4] วัลลภ สุระกำพลธร “การประมวลผลสัญญาณเชิงเลข (Digital Signal Processing)” ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ตุลาคม พ.ศ. 2533
- [5] A.E.Fitzgerald. Charles Kingsley, Jr and Stephen D.Umans. “Electric Machinery fourth Edition” McGraw-Hill @ 1985.
- [6] Bahram shahian. Michael hassul, “Control system design using MATLAB[®] Perntice hall International edition.1993.
- [7] Frank I. Lewis., “Applied Optimal Control & Estimation, Digital Design & Implementation”, 1992 by Texas Instrument Incorporated., Prentice Hall.
- [8] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, “Feedback Control of Dynamic System”, Third Edition, Addison Wesley.
- [9] Hans Walcher, David Kerr, M. J. Shields, Flnfsc, “Position Sensing Angle and Distance Measurement for Engineers”, Butterworth-Heinemann Ltd 1994.
- [10] K. Ogata, Modern Control Engineering, 2nd edition,
- [11] P.N Paraskevopoulos, "On the design of PID output feedback controllers for linear multi-variable system", IEEE Trans. on Industrial Electronics and Control Instrumentation, Vol.IECI-27, PP.16- 8, Feb. 1980
- [12] R.C.Dorf and Seul Jung "Analytic PIDA Controller Design Technique for A Third Order system" Proceedings of the 35th Conference on Decision Control Kobe, Japan. PP

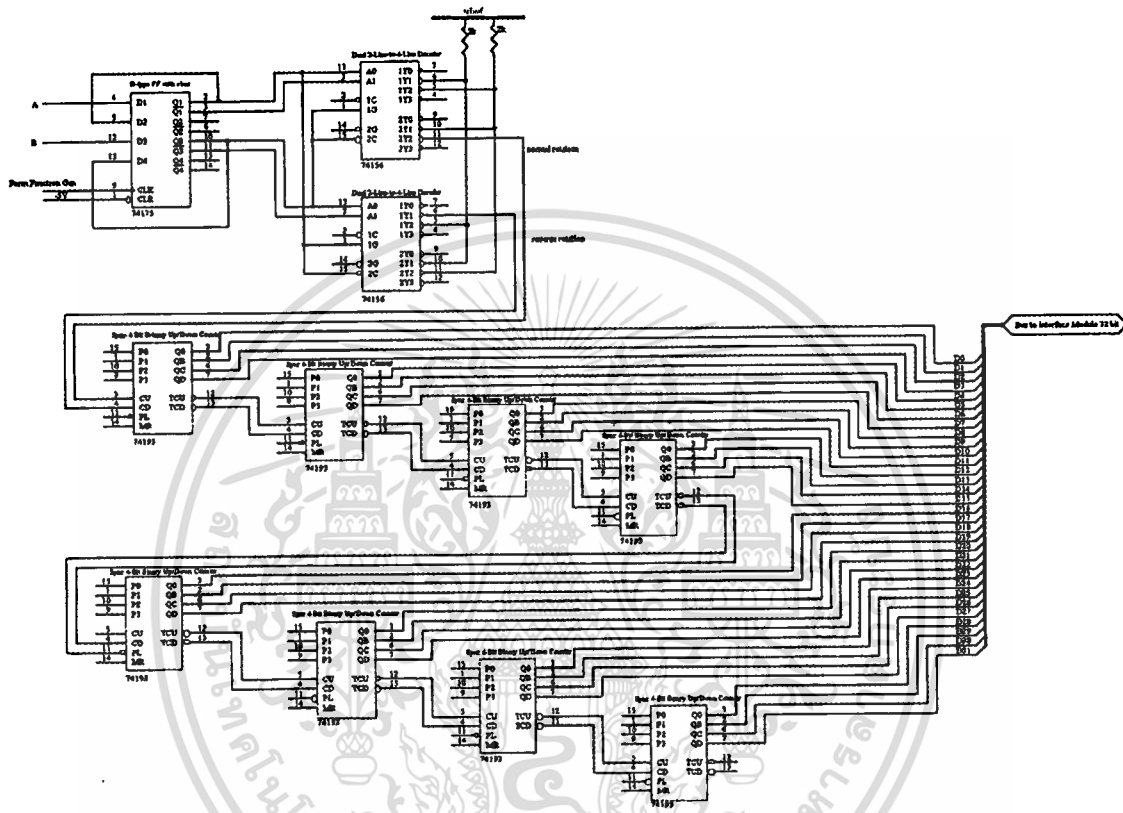
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2513- 2518 December 1996

- [13] R.C.Dorf and Seul Jung “Novel Analytic Technique for PID and PIDA Controller Design”
PP ,163-168 , **IFAC**,Jly, 1996 San Francisco
- [14] R.C. Dorf and R. H. Bishop, **Modern Control System**, 7th edition, Addison Wesley,1995.
- [15] R. C. Dorf and D. R. Miller, "A method for enhanced PID controller design", **Journal of Robotics and Automation**, Vol. 6, pp. 44-47, 1991.
- [16] **TMS320C3X “Users Guide”** Digital signal Processing Products (liter ture number SPRA 013A Texas Instruments January 1993)
- [17] Yasuhiko Dote, Ph.D. “**Servo Motor and Motion Control Using Digital Signal Processors**” Prentice Hall and Texas Instruments
- [18] Yasuhko Takahashi., Michael J. Rabina, David M. Auslancler, “ **Control and Dynamic System**” , Addison Wesley.
- [19] Vinay K. Ingle, John G. Proakis. Northeastern University. “ **Digital Signal Processing Using Matlab V.4[®]**” THE PWS BookWare Companion SeriesTM
- [20] C.M.Liaw and F.J.Lin, “A Robust controller for induction motor drives” **IEEE Transactions on Industrial Electronics**, vol, 41, pp. 308-315.1994
- [21] WALTER W. TURNER. “Simulation of the Response of a dc Motor Driven from an SCR-Controller Source”**IEEE Transaction on Education**,Vol.31,No.3.,pp. 231-234, August 1988.

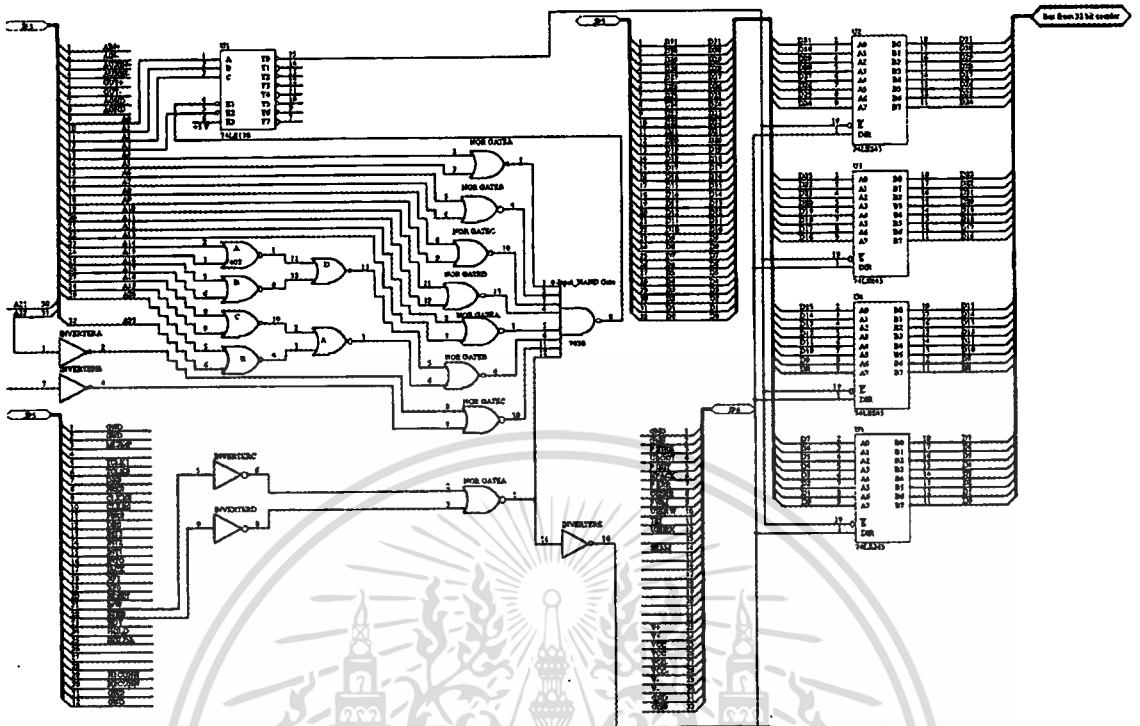
ภาคผนวก

(ก) รายละเอียดวงจรโดยรวมที่ใช้ในวิทยานิพนธ์



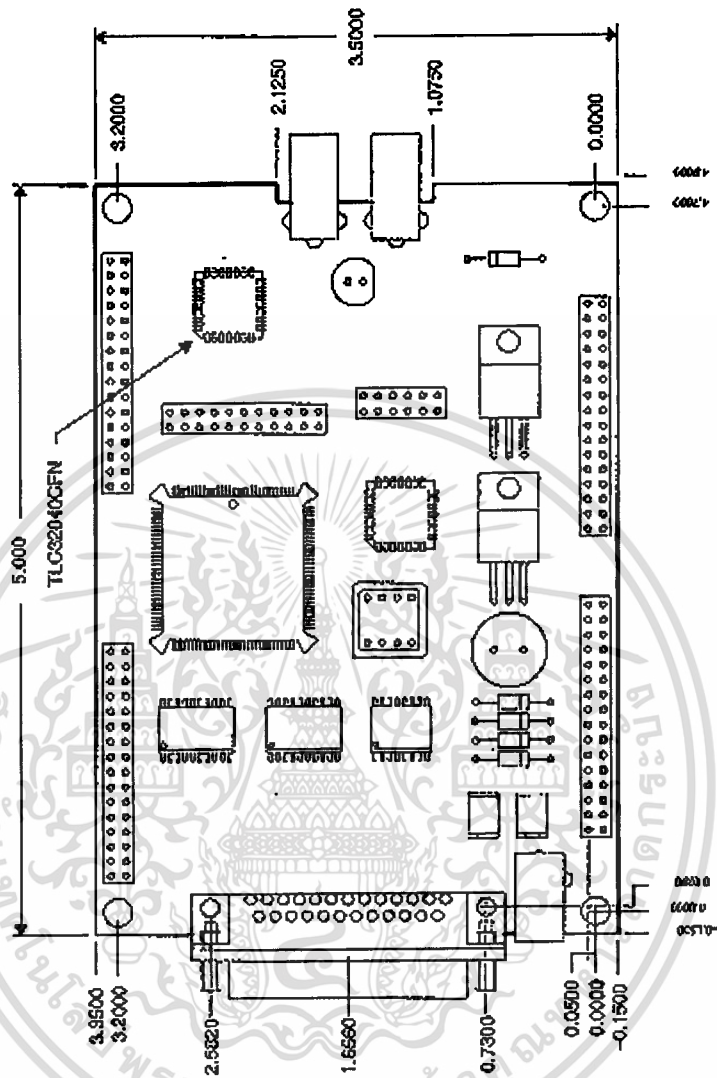
ภาพที่ 1(ก) วงจรนับ 32 บิต และวงจรตรวจับทิศทางการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



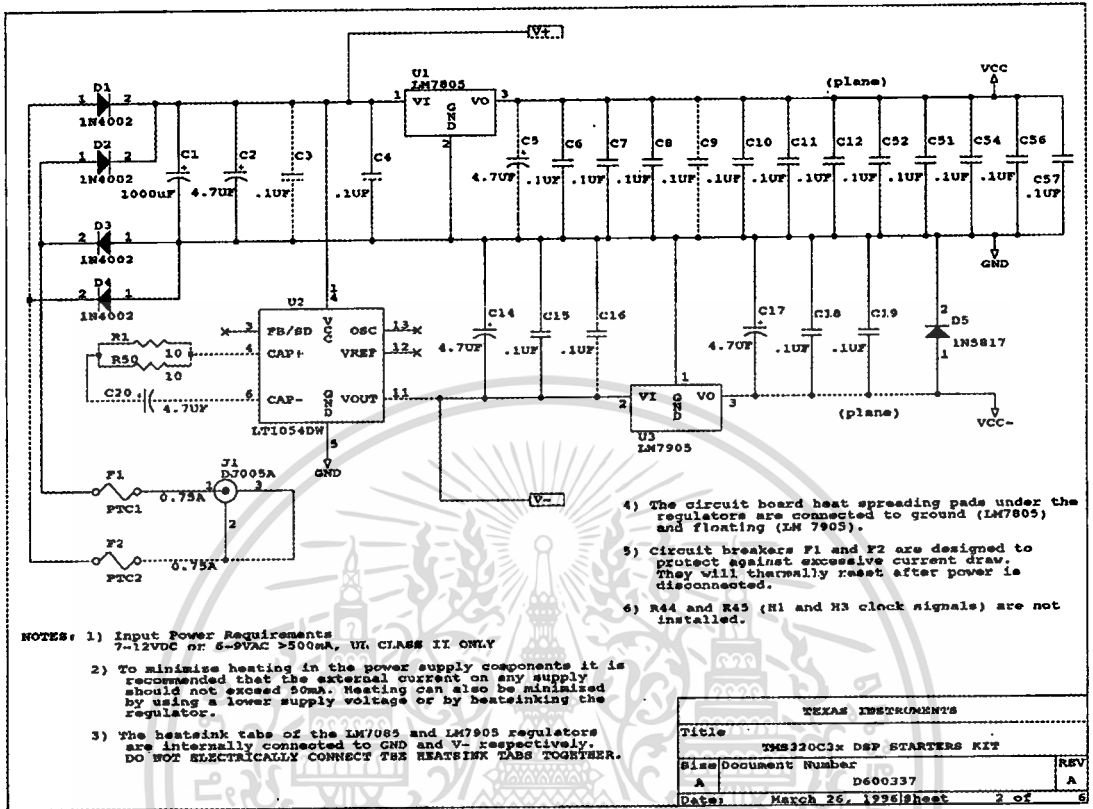
ภาพที่ 2(ก) วงจรอินเทอร์เฟซโมดูล 32 บิต กับ TMS320c31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



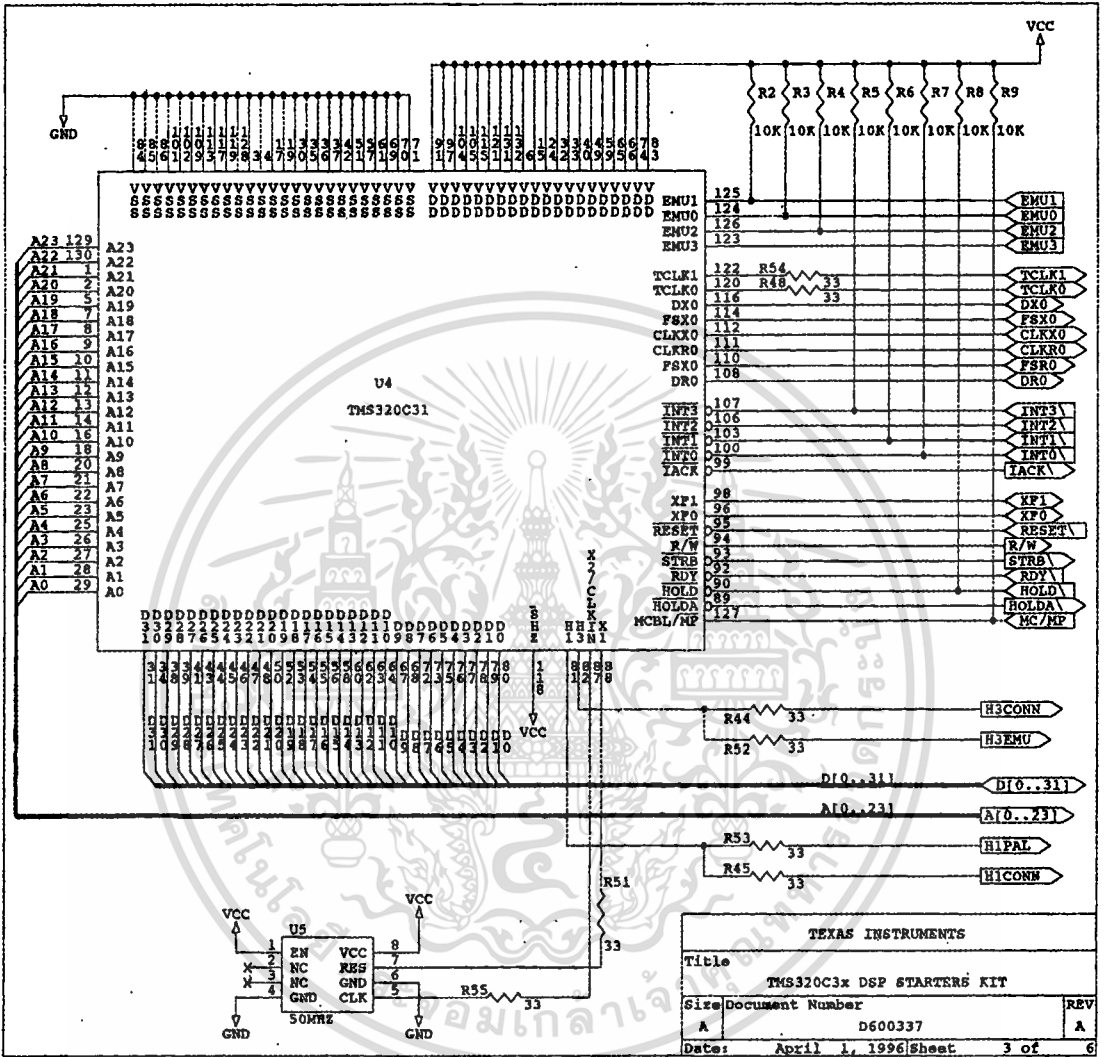
ภาพที่ 3(ก) โมดูลหน่วยประมวลผลสัญญาณเชิงเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



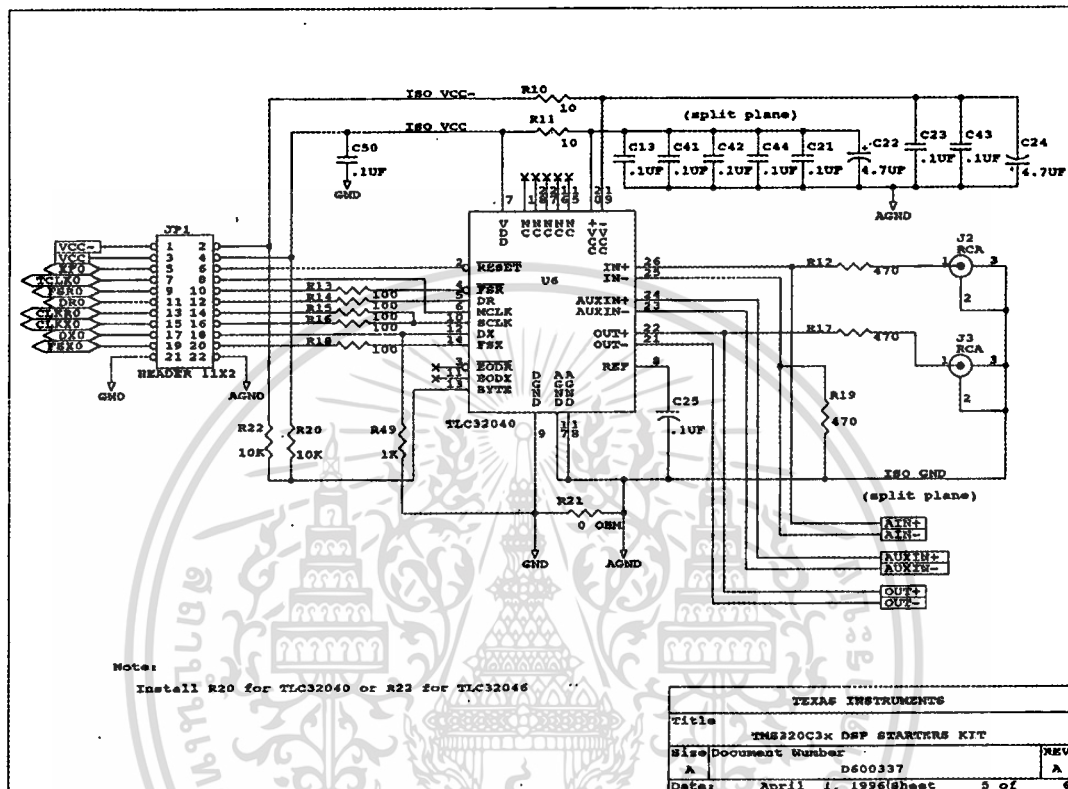
ภาพที่ 4(ก) วงจรแหล่งจ่ายไฟสำหรับโมดูล TMS320c31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



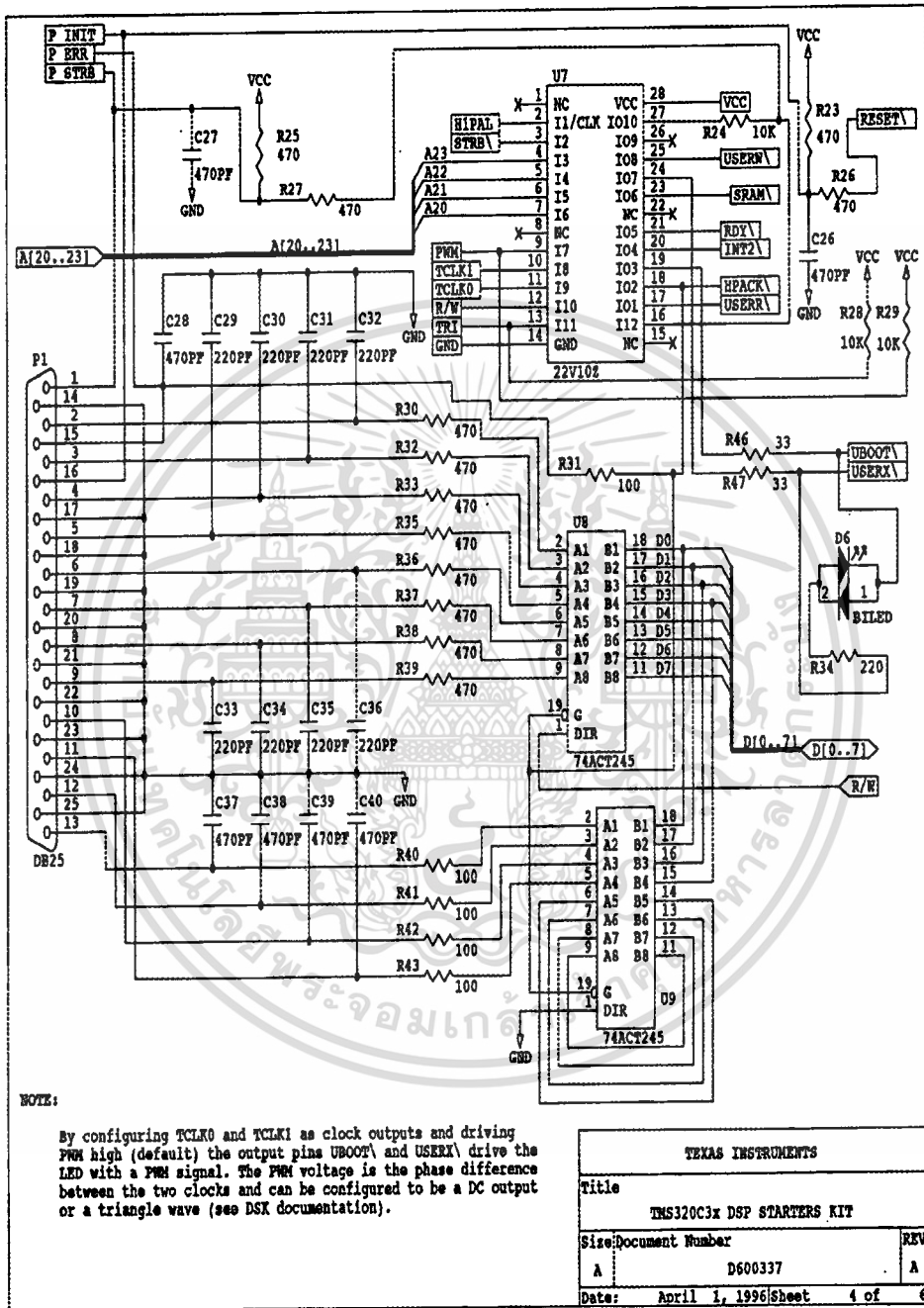
ภาพที่ 5(ก) วงจรภายในของโมดูล TMS320c31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



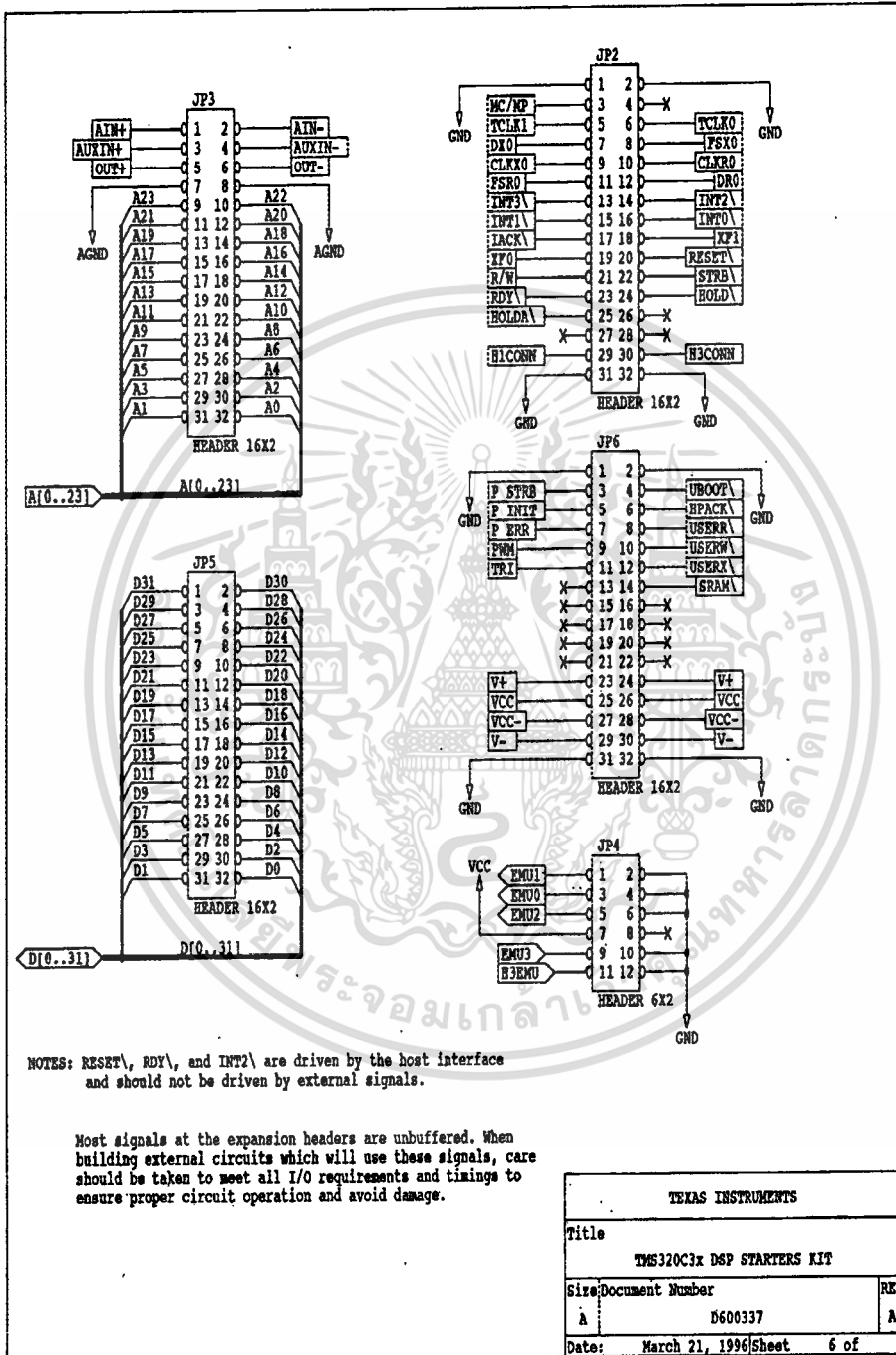
ภาพที่ 6(ก) วงจรแปลงสัญญาณ A/D และ D/A ขนาด 14 บิต ของ TLC32040

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 7(ก) วงจรพอร์ทเชื่อมต่อกับไมโครคอมพิวเตอร์ทางพอร์ทขนานของ TMS320c31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 8(ก) พอร์ที่เชื่อมต่อภายนอกของโมดูล TMS320c31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ข) ตัวอย่างการออกแบบตัวควบคุม PIDA กับกระบวนการอันดับสาม type ต่าง ๆ

จุดประสงค์เพื่อแสดงคุณสมบัติเฉพาะที่ $P.O \leq 5\%$, $T_r \leq 2 \text{ sec}$ และ $\frac{|C(t)|}{|D(t)|} \leq 0.0$

กำหนดให้กระบวนการเป็นชนิด type 0 และมี: $G(s) = \frac{1}{(s+1)(s+3)(s+6)}$

พิจารณาขั้นตอนการหาค่าตัวควบคุม PIDA ตามขั้นตอนที่ 1, 2, 3 และ 4 จะได้ $\xi\omega_n \leq -2$ และ $\zeta \geq 0.707$ สำหรับตำแหน่งที่ต้องการครอบคลุมรากจากจุดกำเนิด ดังนั้นเลือกค่าที่ต้องการครอบคลุมค่ารากที่ $-2.1 \pm 2.0j$, ส่วนค่าแกน real ที่ $R = -2.1$ โดยเหตุที่โพลของ $G(s)$ มากกว่า -1 ดังนั้น $\xi\omega_n = -2$ จึงมีค่ามากกว่าและน้อยกว่าค่ารากที่ $r = -30$.

ขั้นตอนที่ 5. เขียนสมการคุณลักษณะได้เป็น: $1+G(s)G_c(s) = (s+2.1)(s+30)(s+2.1 \pm 2.0j)$

ขั้นตอนที่ 6. จะได้สมการคุณลักษณะเฉพาะของสมการเป็น:

$$s^4 + (10+K)s^3 + (K(a+b+z) + 27)s^2 + (18+K(ab+z(a+b)))s + Kabz = s^4 + 36.6s^3 + 206.23s^2 + 534.561s + 529.83$$

ขั้นตอนที่ 7. สมการทั้งสี่สมการสามารถหาได้ดังนี้

$$10 + K = 36.3,$$

$$K(a+b+z) + 27 = 206.23,$$

$$18 + K(ab+z(a+b)) = 534.561,$$

$$Kabz = 529.83,$$

แก้สมการเพื่อหาค่า a, b, z และ K จะได้ $a, b = 2.3931 \pm 2.0501j$, $z = 2.0288$ และ $K = 26.3$

ดังนั้นจะได้สมการ $G_{c1}(s)$ เป็นดังนี้

$$G_{c1}(s) = 26.3 \frac{(s^2 + 4.7390s + 9.3611)(s + 2.0288)}{s} \quad (1.ข)$$

หาค่าตัวควบคุมที่ทำการออกแบบที่ 2. ของ $G_{c2}(s)$ เมื่อ $R = 2.9$ ทำตามขั้นตอนดังกล่าวจะได้

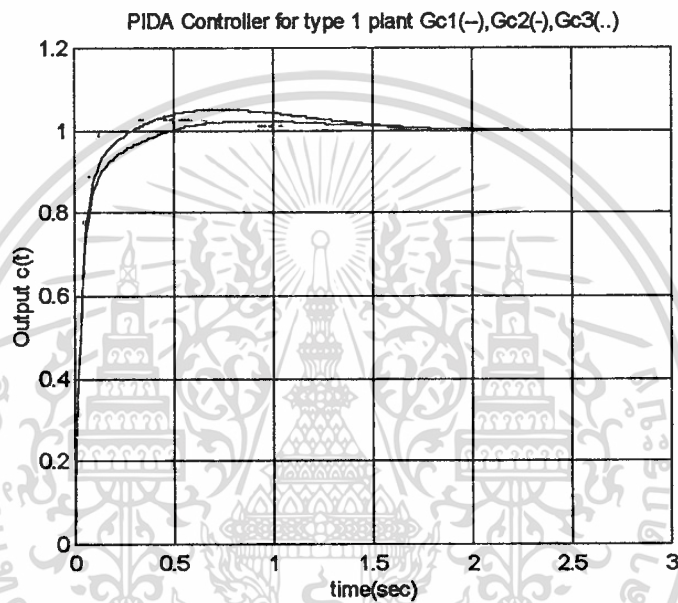
$$G_{c2}(s) = 27.1 \frac{(s^2 + 4.7390s + 9.3606)(s + 2.8843)}{s} \quad (2.ข)$$

ในทำนองเดียวกันที่ $G_{c3}(s)$ เมื่อ $R = 5.9$ จะได้สมการของ $G_{c3}(s)$ เป็น

$$G_{c3}(s) = 30.1 \frac{(s^2 + 4.3556s + 8.3585)(s + 5.9168)}{s} \quad (3.ข)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำสมการที่ (1.๗),(2.๗) และ (3.๗) ไปทำการเขียนแบบเพื่อดูผลตอบสนองของตัวควบคุมทั้งสามดังแสดงในภาพที่ 1(๗) และบันทึกค่าลงในตารางที่ 1(๗) และจากตารางที่ 1(๗) นั่นคือผลที่ได้จากตัวควบคุมที่ได้ออกแบบที่มีค่าความแตกต่างของค่า R ที่ใช้ ผลตอบสนองของตัวควบคุมทั้งหมดมีคุณสมบัติเป็นที่น่าพอใจดังในภาพที่ 1(๗) ผลตอบสนองทั้งหมดได้ผลตรงตามคุณสมบัติขั้นแรกของการออกแบบกระบวนการ. เช่นถ้าเราย้ายค่า R ให้ห่างจากจุดกเนคจะทำให้เปอร์เซ็นต์โอเวอร์ ชูท (P.O) เพิ่มขึ้นแต่เวลาเข้าที่ (T_p) และเวลาคายอด (T_s) จะเร็วซึ่งจากภาพที่ 1(๗) ตัวควบคุมที่ให้ผลดีปลานกลางจะเป็น $G_{c2}(s)$



ภาพที่ 1(๗) ผลตอบสนองของตัวควบคุม PIDA ทั้งสามควบคุมกระบวนการชนิด type 0

ตารางที่ 1(๗) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการอันดับสาม type 0

controller	G_{c1}	G_{c2}	G_{c3}	Specification
R	2.1	2.9	5.9	--
K	26.3	27.1	30.1	--
P.O%	2.47	2.74	8.95	$\leq 5\%$
T_p (sec)	0.73	0.4	0.08	≤ 1.0
T_s (sec)	1.06	0.71	0.19	≤ 2.0
$\max = \frac{ c(t) }{ D(T) }$	0.0016	0.0011	0.0006	0 or < 0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาใหม่โดยให้กระบวนการเป็น type 1 และมี : $G(s) = \frac{1}{s(s+1)(s+7)}$

ขั้นตอนที่ 6. ใหม่จะได้สมการคุณลักษณะเป็น : $s^4 + (8+K)s^3 + (Kz+K(a+b)+7)s^2 + K(ab+z(a+b))s + Kabz = s^4 + 36.6s^3 + 206.23s^2 + 534.561s + 529.83$

ขั้นตอนที่ 7. แก้สมการเพื่อหาค่า a,b,z และ K จะได้ $a,b = 2.2965 \pm 1.5418j$, $z = 2.4471$ และ $K = 28.3$ เมื่อ $R = 2.1$ ดังนั้นจะได้สมการ $G_{c1}(s)$ เป็นดังนี้

$$G_{c1}(s) = 28.3 \frac{(s^2 + 4.5930s + 7.6511)(s + 2.4471)}{s} \quad (4.๗)$$

พิจารณาเมื่อเพิ่มค่า R ที่ตัวควบคุม $G_{c2}(s)$ เมื่อ $R = 6$ จะได้สมการเป็น

$$G_{c2}(s) = 32.2 \frac{(s^2 + 4.0396s + 7.4748)(s + 6.2896)}{s} \quad (5.๗)$$

ในทำนองเดียวกันที่ $G_{c3}(s)$ เมื่อ $R = 10$ จะได้สมการของ $G_{c3}(s)$ เป็น

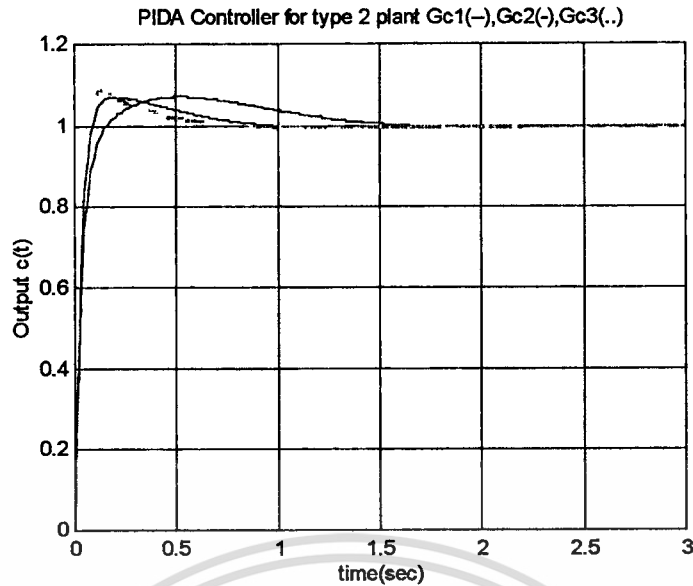
$$G_{c3}(s) = 36.2 \frac{(s^2 + 4.0796s + 7.8422)(s + 8.8872)}{s} \quad (6.๗)$$

เมื่อนำสมการที่ (4.๗), (5.๗) และ (6.๗) ไปทำการเขียนแบบจากโปรแกรม MATLAB สังเกตจะเห็นว่าค่า เปรอร์เซ็น โอเวอร์ชูท P.O. = 6.0, 7.16, 8.8 % , สำหรับตัวควบคุม $G_{c1}(s)$, $G_{c2}(s)$, $G_{c3}(s)$ ตามลำดับ ซึ่งมีคุณสมบัติที่ดีกว่าโดยการเพิ่มค่า K ของแต่ละตัวควบคุมเป็น 35, 50, 70 ของตัวควบคุม $G_{c1}(s)$, $G_{c2}(s)$, $G_{c3}(s)$ ตามลำดับ ซึ่งได้เปอร์เซ็นโอเวอร์ชูทเป็นที่น่าสนใจ และผลตอบสนองของตัวควบคุมทั้งสามแสดงได้ดังภาพที่ 2(๗) และบันทึกค่าต่างๆลงในตารางที่ 2(๗) และได้ตรงตามคุณลักษณะมากที่สุดในการณีของตัวควบคุม $G_{c3}(s)$ ซึ่งให้ผลตอบสนองตรงตามข้อกำหนด

ตารางที่ 2(๗) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการอันดับสาม type 1

controller	G_{c1}	G_{c2}	G_{c3}	Specification
R	2.1	6	10	--
K	35	50	75	--
P.O%	4.87	4.72	4.95	$\leq 5\%$
T_p (sec)	0.45	0.15	0.08	≤ 1.0
T_s (sec)	0.95	0.53	0.3	≤ 2.0
$\max \frac{ C(s) }{ D(s) }$	0.0013	0.0004	0.0002	0 or < 0.05

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับงานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2(ข) ผลของตัวควบคุม PIDA ควบคุมกระบวนการชนิด type 1 หลังจากเพิ่ม เกณฑ์ K

พิจารณาใหม่ให้กระบวนการเป็นชนิด type 2 และมี : $G(s) = \frac{1}{s^2(s+1)}$

ขั้นตอนที่ 6. จะได้สมการคุณลักษณะเป็น

$$s^4 + (1+K)s^3 + K(a+b+z)s^2 + K(ab+z(a+b))s + Kabz = s^4 + 36.6s^3 + 206.23s^2 + 534.561s + 529.83$$

ขั้นตอนที่ 7. แก้สมการเพื่อหาค่า a,b,z และ K จะได้ a,b = 1.9090 ± 1.9418j, z = 2.0242 และ

K = 35.3 เมื่อ R = 2.1 ดังนั้นจะได้สมการ $G_{c1}(s)$ เป็นดังนี้

$$G_{c1}(s) = 35.3 \frac{(s^2 + 3.8180s + 7.4149)(s + 2.0242)}{s} \quad (7.ข)$$

ขั้นตอนที่ 8. พล็อตกราฟผลตอบสนอง และคำนวณหาค่าตัวควบคุมที่สอง $G_{c2}(s)$ เมื่อ R = 0.1 จะ
ได้สมการเป็น

$$G_{c2}(s) = 33.3 \frac{(s^2 + 3.91s + 7.3568)(s + 0.1)}{s} \quad (8.ข)$$

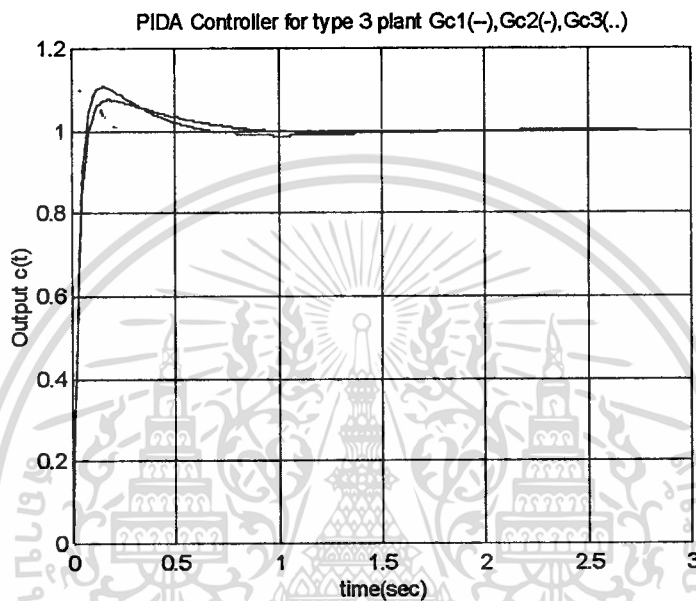
ในทำนองเดียวกันที่ตัวควบคุม $G_{c3}(s)$ เมื่อ R = 30 จะได้สมการของ $G_{c3}(s)$ เป็น

$$G_{c3}(s) = 63.2 \frac{(s^2 + 4.1870s + 8.4498)(s + 14.1739)}{s} \quad (9.ข)$$

นำสมการที่ (7.ข),(8.ข) และ (9.ข) ไปทำการเขียนแบบจากโปรแกรม MATLAB ในกรณีนี้เปอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น โอลเวอร์ฮูทไม่เป็นที่น่าพอใจนักคือมี P.O = 7.59, 10.74, 16.09 % , สำหรับตัวควบคุม $G_{c1}(s)$, $G_{c2}(s)$, $G_{c3}(s)$ ตามลำดับ โดยการเพิ่มค่าเกนซ์ K ต่อเลขเคด กับตัวควบคุมที่ K = 60, 85, 280 ของตัวควบคุม $G_{c1}(s)$, $G_{c2}(s)$, $G_{c3}(s)$ ตามลำดับ ซึ่งมีเปอร์เซ็นต์โอเวอร์ชูน้อยกว่า 5% ที่ทำให้ค่า $\max \frac{|C(t)|}{|D(t)|}$ ของตัวควบคุม $G_{c2}(s)$ ใช้เวลามากกว่าตัวควบคุมกระบวนการชนิดอื่น ซึ่งผลตอบสนองของตัวควบคุมทั้งสามแสดงได้ดังภาพที่ 3(ข) และบันทึกค่าต่างๆ ลงในตารางที่ 3(ข) ตามข้างล่างนี้



ภาพที่ 3(ข) ผลของตัวควบคุม PIDA ควบคุมกระบวนการชนิด type 2 หลังจากเพิ่มเกนซ์ K

ตารางที่ 3(ข) แสดงผลของตัวควบคุม PIDA สำหรับกระบวนการอันดับสาม type 2

controller	G_{c1}	G_{c2}	G_{c3}	Specification
R	2.1	0.1	30	--
K	60	85	280	--
P.O%	4.46	4.95	4.9	$\leq 5\%$
T_p (sec)	0.12	0.08	0.02	≤ 1.0
T_s (sec)	0.45	0.34	0.09	≤ 2.0
$\max = \frac{ C(t) }{ D(t) }$	0.0011	0.0145	0.00003	0 or < 0.05

สังเกตที่ตัวควบคุมที่มีค่า R คำน้อยๆพบว่าจะทำให้ค่าที่ครอบคลุมค่ารากแกนขวาด้านล่างเป็นที่น่าสนใจตรงตามวัตถุประสงค์ตามคุณสมบัติ ส่วนเปอร์เซ็นต์โอเวอร์ชูกำหนดโดยเลือกค่า R ได้ตามต้องการที่จะให้เวลาเข้าที่ (T_s) และเวลาขยับ (T_p) เร็วขึ้น แต่ว่าเปอร์เซ็นต์โอเวอร์ชูกจะมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นด้วย และมากกว่าตัวควบคุมสำหรับระบบชนิดที่สอง ซึ่งตัวควบคุมที่มีการต่อแชนแนลเกณฑ์ ได้ตามต้องการเพื่อนำไปเปรียบเทียบกับกระบวนการชนิดอื่นต่อไป

ดังนั้นการออกแบบตัวควบคุม PID สำหรับกระบวนการอันดับสามที่มีความยุ่งยากและมีข้อกำหนดที่ตายตัว สามารถทำให้ปัญหาเหล่านั้นหมดไปได้โดยใช้วิธีการออกแบบของตัวควบคุม PIDA ซึ่งมีข้อดีในการวิเคราะห์ และพิสูจน์ได้โดยการคำนวณหาค่าความแตกต่างกันแต่ละชนิดของกระบวนการอันดับ 3 , และพิสูจน์ในการนำเสนอวิธีการออกแบบตัวควบคุม PIDA ผลที่ได้รับจากการเลียนแบบโดยใช้วิธีการออกแบบตัวควบคุม PIDA ทำให้วิเคราะห์ง่ายเพียงแต่เพิ่มค่าเกณฑ์ที่ต่อแชนแนลกับตัวควบคุมบวกกับค่ารากในวงรอบปิด ดังนั้นขอเสนอสำหรับการออกแบบตัวควบคุม PIDA จึงเป็นอีกก้าวหนึ่งของผู้ที่ต้องการเริ่มออกแบบและวิเคราะห์ระบบที่สามารถพัฒนาได้ง่ายตามต้องการ และตัวควบคุมจะให้คุณสมบัติที่ดีเป็นที่พอใจสำหรับกระบวนการที่เป็นอันดับ 3. ซึ่งสามารถสร้างได้ไม่ยากเกินไป.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ค) โปรแกรมที่ในวิทยานิพนธ์ในการควบคุมมอเตอร์

```

SACL *                               SPLK #0FFFFH,*+                               SPLK #0000h,T3CNT
SUB #01Ah;0X01Ah is the max speed error  SPLK #029H,*+                               SPLK #0000h,T3CMP
BLEZ OK                               SPLK #029H,*+                               SPLK #0000h,COMCON
SPLK #01Ah,*                           SPLK #0,*                                     SPLK #0000h,DBTCON
OK                                       LAR AR2,#0300h                               SPLK #0000h,ACTR
LT SPEED_COUNT                          LAR AR1,#stack ;Init context save stack pointer  SPLK #0000h,SACTR
MPY *                                    LDP #0c0h                                     SPLK #0000h,CMPR1
PAC                                       SPLK #0006Fh, WD_CNTL                       SPLK #0000h,CMPR2
Assembly Code                            SPLK #05555h, WD_KEY                         SPLK #0000h,CMPR3
;*****                               SPLK #0AAAAh, WD_KEY                       SPLK #0000h,SCMPR1
;KEITANUN SUKCHU;                       ; initialize WDT registers                  SPLK #0000h,SCMPR2
; Description : This is the separately Excited DC  SPLK #06Fh, WD_CNTL ; clear WDFLAG,          SPLK #0000h,SCMPR3
motor drive code                          Disable WDT, set WDT                        SPLK #0000h,CAPCON ;no capture
;*****                               ; for 1 second overflow.                  SPLK #00fth,CAPFIFO
.include "c31.h"                           ; Set up CLKOUT to be SYSCLK p6-6         LACC FIFO1
;Speed regulator coefficients setting       SPLK #40C0h,SYSCR                            LACC FIFO2
Kp .set 015                               LACC SYSSR                                  LACC FIFO3
Ki .set 004                               AND #069FFh                                ;GP Timer 2 Setting start with GPT1 presc 128
;-----                               SACL SYSSR                                SPLK #0fffeh,T2PER
; Variable definitions                    ;set up PLL                               SPLK #00000h,T2CNT
;-----                               ;CPULOCK~20Mhz,SYSClk=10Mhz             SPLK #17C0h,T2CON
.bss CAPT,1 ; Capture indication          SPLK #0002h,CKCR0 ; PLL disabled          ;PWM Unit setting
.bss COMP,1 ;Duty cycle                   SPLK #00b1h,CKCR1                          SPLK #00500,T1PER
.bss Idc,1 ;Line current                  SPLK #0081h,CKCR0                          SPLK #0000h,T1CNT
.bss Idc_ref,1 ;Current reference         ;Set up zero Wait States for external memory  SPLK #0FFFh,ACTR
.bss SPEED_REF,1 ;Speed reference        ; MAR *,AR1                               SPLK #0508h,DBTCON
.bss SPEED_COUNT,1 ;Speed loop frequency ; LACC #0008h                              SPLK #00000,CMPR1
.bss stack,6 ;Context save stack         ; SACL *                                    SPLK #00000,CMPR2
;-----                               ; OUT *,WSGR                              SPLK #00000,CMPR3
Reset & interrupt vectors                ;I/O setting p11-11                       SPLK #0287h,COMCON
.sect "vectors"                           LDP #0c1h                                  SPLK #8287h,COMCON
RSVECT B _c_int0 ; PM 0 Reset Vector      SPLK #0000th, OCRA                          SPLK #2800h,T1CON
.space 16*6                               SPLK #0070h, OCRB                          SPLK #2840h,T1CON
INT4 B CAPINT                             SPLK #02800h, PBDATDIR ;not shut down  SPLK #0100h,GPTCON
.space 16*2                               clearing fault                               ;Capture Unit Setting
INT6 B ADCINT                             LACC PBDATDIR                              SPLK #0b0feh,CAPCON
.space 16*38                              ; Clear EV control registers              SPLK #00fth,CAPFIFO
.text                                       LDP #0c0h                                  ;Core Mask Setting
.global _c_int0                           SPLK #0000h,T1CON ;no timer enable          LDP #0
_c_int0                                   SPLK #0000h,T1PER                          LACC #028h
RC OVM ; Reset overflow mode              SPLK #0000h,T1CNT                          SACL IMR
SETC SXM ; Reset sign extension mode     SPLK #0000h,T1CMP                          LACC IFR
CLR XC XF                                 SPLK #0000h,T2CON                          SACL IFR
SETC INTM ; Set global interrupt mask     SPLK #0000h,T2PER                          ;EV Mask Setting, Vector & Flag reset p11-46
MAR *,AR2 ;Init speed variables stackLAR  SPLK #0000h,T2CNT                          LDP #0c0h
AR2,#0300h                               SPLK #0000h,T2CMP                          LACC IFRA
SPLK #0,*+                               SPLK #0000h,T3CON                          SACL IFRA
SPLK #0,*+                               SPLK #0000h,T3PER                          LACC IFRB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SACL IFRB	AND #010h	SACL CMPR2
LACC IFRC	BZ FAULT_CLEAR	SPLK #0000,CMPR1
SACL IFRC	SPLK #02820h,PBDATDIR	B END
SPLK #0,IMRA	LDP #0c0h	RISING1
SPLK #0,IMRB	LOOP ;Polling loop	LACC COMP
SPLK #7,IMRC	NOP	LDP #0c0h
LACC IVRA	NOP	SPLK #0F7Dh,ACTR
LACC IVRB	NOP	SACL CMPR1
LACC IVRC	B LOOP	SACL CMPR2
;ADC Unit setting p3-8	SEQUENCE	SPLK #0000,CMPR3
LDP #0c0h	LDP #0	B END
SPLK #0403h,ADCTRL2	LACC CAPT	FALLING1
SPLK #1b0ah,ADCTRL1;ADCIN5	SUB #1	LACC COMP
CLRC INTM	SFL	LDP #0c0h
LDP #0 ;Variables init	ADD #CAPT_DETER	SPLK #0FD7h,ACTR
SPLK #001FH,Idc_ref	BACC	SACL CMPR2
SPLK #040H,SPEED_REF	CAPT_DETER	SACL CMPR1
SPLK #0500,COMP	B FALLING3	SPLK #0000,CMPR3
SPLK #0000,SPEED_COUNT	B FALLING1	END
SPLK #0000H,CAPT	B RISING2	RET
*****	B FALLING2	SPEED_REG
STARTING PROCEDURE	B RISING1	MAR *,AR2
*****	RISING3	LAR AR2,#0302h
;Disable CAPT Input p2-75	LACC COMP	LDP #0
LDP #0c0h	LDP #0c0h	SPLK #1,SPEED_COUNT
SPLK #8000h,CAPCON	SPLK #0DF7h,ACTR	;Speed and speed error calculation
SPLK #00fh,CAPFIFO	SACL CMPR3	CLRC SXM
;Set I/O unit as I	SACL CMPR1	ZAC
LDP #00c1h	SPLK #0000,CMPR2	OR #0FFFFH
SPLK #0FF00h, OCRB	B END	RPT #15
SPLK #0000h, PCDATDIR	FALLING3	SUBC *
;WHICH SEQUENCE?	LACC COMP	AND #0FFFFH
LACC PCDATDIR	LDP #0c0h	SUB SPEED_REF
AND #0070H	SPLK #07FDh,ACTR	NEG
LDP #0	SACL CMPR1	SETC SXM
SACL CAPT	SACL CMPR3	;Speed regulation
LACC CAPT,12	SPLK #0000,CMPR2	LAR AR2,#0303h
SACH CAPT	B END	SACL *+
;Disable I/O Unit	RISING2	SUB *+
LDP #0c1h	LACC COMP	SACL *
SPLK #0FF70h, OCRB	LDP #0c0h	LT *-MPY
;Capture Unit Setting	SPLK #07DFh,ACTR	#Kp
LDP #0c0h	SACL CMPR2	LTP *-MPY
SPLK #0b0fch,CAPCON	SACL CMPR3	#Ki
SPLK #00fth,CAPFIFO	SPLK #0000,CMPR1	LTD *
*****	B END	ADD Idc_ref,4
END OF STARTING PROCEDURE	FALLING2	SFR
*****	LACC COMP	SFR
FAULT_CLEAR ;Init the pre-driver	LDP #0c0h	SFR
LDP #0c1h	SPLK #0D7Fh,ACTR	SFR
LACC PBDATDIR	SACL CMPR3	SACL Idc_ref

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;Reset Speed loop timer
SPLK #0,SPEED_COUNT
;Restore
LAR AR2,#0300H
RET
ADCINT
;save status registers
MAR *,AR1
MAR *+ ; skip one position
SST #1, *+ ; save ST1
SST #0, * ; save ST0
LDP #0
LACC SPEED_COUNT
SUB #1250
BNZ NO_SPEED_REG
CALL SPEED_REG
NO_SPEED_REG
LACC SPEED_COUNT
ADD #1
SACL SPEED_COUNT
LDP #0c0h
LACC SYSIVR
;current error calculation
CLRC SXM
LACC ADCFIFO1,10
LDP #0
SUB Idc_ref,16
SETC SXM
;current proportional regulator
ADD COMP,16
SACH COMP
;current regulator output limitation
LACC COMP
BGZ SUP_LIM
SPLK #0,COMP
B COMP_OK
SUP_LIM
SUB #0500
BLZ COMP_OK
SPLK #0500,COMP
COMP_OK
;output PWM
CALL SEQUENCE
;restore status registers
MAR *, AR1 ; make stack pointer active
LST #0, *- ; load ST0
LST #1, *- ; load ST1
CLRC INTM
RET
CAPINT
MAR *,AR1
MAR *+ ; skip one position
SST #1, *+ ; save ST1
SST #0, * ; save ST0
;Capture interrupt handling
LDP #0c0h
LACC IVRC ;FLAG RESET
;speed calculation
MAR *,AR2
LACL T2CNT
SACL *+
SUB *+
SACL *-ADD
*
SACL *-;
Disable CAPT Input p2-75
LDP #0c0h
SPLK #8000h,CAPCON
SPLK #00fh,CAPFIFO
;Set I/O unit as I
LDP #0c1h
SPLK #0FF0h, OCRB
SPLK #0000h, PCDATDIR
;WHICH SEQUENCE?
LACC PCDATDIR
AND #0070H
LDP #0
SACL CAPT
LACC CAPT,12
SACH CAPT
;Commutation
CALL SEQUENCE
;Disable I/O Unit
LDP #0c1h
SPLK #0FF70h, OCRB
;Capture Unit Setting
LDP #0c0h
SPLK #0b0fcb,CAPCON
SPLK #00fh,CAPFIFO
MAR *, AR1 ; make stack pointer active
LST #0, *- ; load ST0
LST #1, *- ; load ST1
CLRC INTM
RET
Linker File
/*-----*/
/* LINKER COMMAND FILE - MEMORY
SPECIFICATION for C31 */
/*-----*/
MEMORY
{
PAGE 0 : VECS : origin = 0h , length = 040h /*
PROGRAM */
PROG : origin = 40h , length = 0800h /* Ext
mem */
PAGE 1 : MMRS : origin = 0h , length = 05Fh
/* MMRS */
B2 : origin = 0060h , length = 020h /* DARAM
*/
B0 : origin = 0100h , length = 0200h /*
DARAM */
B1 : origin = 0300h , length = 0200h /*
DARAM */
/*-----*/
/* SECTIONS ALLOCATION */
/*-----*/
SECTIONS
{
.vectors : { } > VECS PAGE 0 /* INTERRUPT
VECTOR TABLE */
.text : { } > PROG PAGE 0 /* CODE */
.mmrms : { } > MMRS PAGE 1 /* Memory
Mapped Registers */
.blk0 : { } > B0 PAGE 1 /* Block B0 - page 4 */
.blss : { } > B2 PAGE 1 /* Block B2 - page 0 */
.blk1 : { } > B1 PAGE 1 /* Block B1 - page 7 */
}
}
โปรแกรมที่ใช้ในการควม
คุมตำแหน่งของความเร็ว
/*-----*/
=Name: ENCODER.C
project: keitanun Sukchu
-----
Function List: void Encoder_Init(unsigned
rollover)
void Encoder_ZeroPosition(void)
void Encoder_MatchIncrPhase(void)
void Encoder_CalcPhase(int sin,int ncos)
unsigned Encoder_SamplePosition(void)
void Encoder_CalePosition(unsigned
qep_incr_sample)
Status: OK
Target: TMS320C31
-----
/
/*-----*/
/* Header Files */
/*-----*/
#include <stdlib.h>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <math.h>
#include <q15_div.h> /* q15.lib */
#include <q15_atan.h> /* q15.lib */
#include "TLC32040.h"
/* TMS320C31 memory mapped registers */
#include "adc.h" /* ADC macros */
#include "evm_qep.h" /* QEP unit */
#include "monitor.h" /* test program & monitor */
#include "encoder.h" /* encoder definition */
/*-----*/
/* Variable Declaration */
/*-----*/
volatile unsigned encoder_position[2];
/*-----*/
/* void Encoder_Init(unsigned rollover) */
/*-----*/
/* Arguments: Incremental counts/revolution =
4-times line count */
/* */
/* Return value: None */
/*-----*/
void Encoder_Init(unsigned rollover)
{
/*-----*/
/* Initialize TLC32040 ADC */
/*-----*/
ADCTRL1 = 0x1800; /* enable both ADCs */
ADGTRL1 |= 0x0100; /* clear interrupt flag */
ADCTRL2 = 0x0003; /*ADC_CLOCK =
SYSCLK/10 = 1MHz*/
/*-----*/
/* Initialize TLC32040 QEP Circuit */
/*-----*/
QEP_Init(0x0,rollover); /* QEP counts Timer 2
*/
/*-----*/
/* Get zero incremental position (via index) and
*/
/* synchronize incremental count and phase */
/*-----*/
Encoder_MSG0; /* turn encoder into zero
position */
Encoder_ZeroPosition0; /* detect zero position
adjust counter */
Encoder_MatchIncrPhase0; /* match counter
(quadrant) and phase */
} /* end Encoder_Init */
/*-----*/
/* void Encoder_ZeroPosition(void) */
/*-----*/
/* Function: Modify/adjust qep_diff, so that the
incremental count */
/* of the encoder (using function QEP_GetIncr0)
matches with */
/* phase derived from the analog sin/-cos signal,
as shown */
/* below. */
/*-----*/
void Encoder_ZeroPosition(void)
{
/* Configure IOPC6/CAP3 to detect level */
OCRB &= 0xFFBF; /* clear bit 6 */
PCDATDIR &= 0xBFFF; /* clear bit 14 */
while ( !PCDATDIR & 0x0040); /* poll bit 6
*/
T2CNT = 0x0;
}
/*-----*/
/* void Encoder_MatchIncrPhase(void) */
/*-----*/
/* Function: Modify/adjust qep_diff, so that the
incremental count*/
/* of the encoder (using function QEP_GetIncr0)
matches with */
/*-----*/
void Encoder_MatchIncrPhase(void)
{
/*-----*/
/* local variables */
/*-----*/
char c;
volatile int buffer[2];
volatile unsigned ubuffer[2];
/*-----*/
/* Synchronize incremental count and encoder
phase */
/*-----*/
/* Assumption: ADCIN5 <-> QEP1/sin(x) */
/* ADCIN13 <-> QEP2/negcos(x) */
/* */
/* (T2CNT % QEP_ENCODER) % 4 + qep_diff
= Quadrant 0,1,2,3 */
/*-----*/
do
{
/*-----*/
/* aquire encoder signals */
ubuffer[1] = Encoder_SamplePosition0;
ADC_READ2(buffer[0],buffer[1]);
/* Correct U0 and U90 input signals offset */
buffer[0] = buffer[0] - ENC_U0_OFFSET;
buffer[1] = buffer[1] - ENC_U90_OFFSET;
/*-----*/
/* check, if nearly in the middle of a quadrant */
/*-----*/
} while ( abs(abs(buffer[0])-abs(buffer[1])) >
0x4000);
/*-----*/
ubuffer[0] = Encoder_CalcPhase(buffer[0],buffer
[1]);
ubuffer[0] = (ubuffer[0] >> 14) & 0x0003;
/* extract quadrant */
ubuffer[1] = ubuffer[1] & 0x0003;
qep_diff = qep_diff + ((int) ubuffer[1] - (int)
ubuffer[0]);
}
/*-----*/
/* unsigned Encoder_SamplePosition(void); */
/*-----*/
/* INCREMENTS --> Timer 2 counter (T2CNT)
*/
/* Arguments: Buffer for increments sample */
/* Return value: Incremental count */
/*-----*/
asm("ADCTRL1.set 7032h ");
asm("T2CNT.set 7405h ");
asm(".ref _qep_diff ");
asm(".ref_QEP_GetIncr ");
asm(".def_Encoder_SamplePosition ");
asm("_Encoder_SamplePosition: ");
asm(" ldp #ADCTRL1/128 ");
asm(" lacl ADCTRL1 ");
asm(" and #0FF81h ;clear channels ");
asm(" or #005Bh ;select channel 5 and 13");
asm(" sacl ADCTRL1 ;(1) start ADC's ");
#ifdef QEP_POWER2
asm(" ldp #_qep_diff ");
asm(" lacl #0 ");
asm(" sub _qep_diff ");
asm(" ldp #(T2CNT/128) ");
asm(" rpt #6 ");
asm(" nop ");
asm(" ;capture Timer 2 800ns after ADC start (1)
");
asm(" ;-----");
asm(" add T2CNT ");
asm(" and #(QEP_ROLLOVER-1) ");
asm(" ret ");

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#else
asm(" rpt #5 ");
asm(" nop ");
asm(" ;capture Timer 2 800ns after ADC start (1)
");
asm(" ;-----");
asm(" call _QEP_GetIncr; 500ns (NOP) +
200ns (call)
");
asm(" ; + 100ns in subroutine ");
asm(" ret ");
#endif
/*-----*/
/* void Encoder_CalcPosition(unsigned
qep_incr_sample); */
/*-----*/
/* Function: Calc encoder position */
/* increments (Timer 2) */
/* */
/* Arguments: increments sample */
/* */
/* Global var's: unsigned qep_position[2] */
/* */
/* Return value: None */
/*-----*/
void Encoder_CalcPosition(unsigned
qep_incr_sample)
{
/*-----*/
/* locals var's */
/*-----*/
volatile int sin_sample;
volatile int ncos_sample;
volatile unsigned incr;
volatile unsigned buffer;
incr = qep_incr_sample;
/*-----*/
/* read converted sin and -cos encoder signals */
/*-----*/
ADC_READ2(sin_sample,ncos_sample);
/* Correct U0 (sin_sample and U90
(ncos_sample) offset */
sin_sample = sin_sample - ENC_U0_OFFSET;
ncos_sample = ncos_sample -
ENC_U90_OFFSET;
/*-----*/
/* phase calculation */
/*-----*/
encoder_position[0] = Encoder_CalcPhase
(sin_sample,ncos_sample);
}
/*-----*/
/*
*/
/* correct incremental steps according to phase
information */
/*-----*/
/*
*/
buffer = ((encoder_position[0] >> 14) &
0x0003);
switch (buffer)
{
case 0: if ((incr & 0x0003) == 3)
incr = (incr + 1) & (qep_rollover-1);
break;
case 3: if ((incr & 0x0003) == 0)
incr = (incr - 1) & (qep_rollover-1);
break;
} /* switch */
/* remove (redundant) quadrant information (last
two bits) */
encoder_position[1] = (incr >> 2);
}
/*-----*/
/* int Encoder_CalcPhase(int qep_sin,int
qep_negcos); */
/*-----*/
/* Function: Incremental encoder initialization */
/* */
/* Arguments: None */
/* */
/* Return value: None */
/*-----*/
int Encoder_CalcPhase(int qep_sin, int
qep_negcos)
{
int phase;
int buffer;
/*-----*/
/* general calculation, within 1st quadrant */
/*-----*/
if (abs(qep_sin) == abs(qep_negcos))
phase = (PI/4);
else if (abs(qep_sin) < abs(qep_negcos))
{
buffer = q15_div(abs(qep_sin),abs
(qep_negcos));
/* phase = q15_atan(buffer); */
phase = q15p_atan(buffer);
}
else
buffer = q15_div(abs(qep_negcos),abs
(qep_sin));
/* phase = (PI/2) - q15_atan(buffer); */
phase = (PI/2) - q15p_atan(buffer);
}
/*-----*/
/*
*/
/* get 2nd, 3rd to 4th quadrant */
/*-----*/
if (qep_sin >= 0)
{
if (qep_negcos > 0)
phase = PI - phase; /* 2nd quadrant */
} /* end if */
else
{
if (qep_negcos > 0)
phase = PI + phase; /* 3rd quadrant */
else
phase = -phase; /* 4th quadrant */
} /* end else */
return phase;
}
/*-----*/
Name: EVM_QEP.C
Project: TLC240.LIB
Description: Initializes the QEP circuit module
Provides functions for serving module
/*-----*/
Function List: void QEP_Init(unsigned
zero_position)
unsigned QEP_GetIncr(void)
Target: TMS320C31
/*-----*/
/* Include Files */
/*-----*/
#include "c240.h"
#include "evm_qep.h"
/*-----*/
/* Global Var's */
/*-----*/
asm(".bss _qep_rollover,1 "); /* align on ONE
PAGE */
asm(".bss _qep_diff,1 ");
asm(".bss _qep_temp,1 ");
/*-----*/
/* void QEP_Init(unsigned zero_position,
unsigned rollover) */
/*-----*/
/*Function: Incremental encoder (QEP Circuit)
initialization */
/* based on timer 2 */

```

เอกสารนี้เป็นลิขสิทธิ์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* */
/* Arguments: unsigned zero_position */
/* unsigned rollover */
/* Return value: none */
/*-----*/
void QEP_Init(unsigned zero_position, unsigned
rollover)
{
OCRBR |= 0x30; /* enable qep1/2 mux'd inputs */
T2CON = 0xD83A; /* enable qep circuit using
timer2 */
/* bit 15,14: timer 2 doesn't on emulation
suspend */
/* bit 13-10: directional up/down */
/* bit 5-4: qep pulse circuit triggers timer 2 */
/* bit 3-2: compare no shadowed */
/* bit 1: enable compare */
T2CON |= 0x0040; /* enable timer 2 */
CAPCON &= 0x1D0F;
CAPCON |= 0xE000; /* enable qep decoder
circuit */
/* init rollover value and zero position for qep
timer 2 */
T2CNT = zero_position;
qep_rollover = rollover;
qep_diff = 0;
}
/*-----*/
/* unsigned QEP_GetIncr(void) */
/*-----*/
/* Funtion Returns incremental step of encoder
*/
/* 4-times resolution */
/* Arguments: None */
/* Return value: unsigned incremental value
module qep_rollover */
/*-----*/
asm(" .text ");
asm(" .def _qep_rollover ");
asm(" .def _qep_diff ");
asm(" .def _qep_temp ");
asm(" .def _QEP_GetIncr ");
asm("T2CNT .set 7405h");
asm("_QEP_GetIncr:");
asm(" ;read timer 2 counter ");
asm(" ;----- ");
asm(" ldp #T2CNT/128 ");
asm(" lacc T2CNT ");
asm(" ldp #_qep_temp ");
asm(" sacl _qep_temp ;save temporarily ");
asm(" ");
asm(" ;modulo calculation ");
asm(" ;----- ");
asm(" check: lacc _qep_temp,16 ");
asm(" sub _qep_diff,16 ");
asm(" bcmd Nega,lt ; ACC negative ");
asm(" sub _qep_rollover,16 ");
asm(" bcmd OK,lt ; ACC is OK ");
asm(" lacc _qep_diff,16 ");
asm(" add _qep_rollover,16 ");
asm(" sach _qep_diff ");
asm(" b check; ");
asm(" ");
asm(" Nega: lacc _qep_diff,16 ");
asm(" sub _qep_rollover,16 ");
asm(" sach _qep_diff ");
asm(" b check; ");
asm(" ");
asm(" OK: add _qep_rollover,16 ");
asm(" ");
asm(" EPI: sach _qep_temp ");
asm(" lacc _qep_temp ;ACC = Return Value ");
asm(" ret ");
;-----
; Name: Q15_DIV.ASM
; Project: Q15.LIB
;-----
; int q15_atan(int);
; unsigned q15p_atan(unsigned)
; Status:
;
; Target: TMS320C31
;-----
; Name: Q15_DIV.ASM
; Project: Q15.LIB
;-----
include "q15_atan.inc";lookup table
;-----
; int q15_atan(int);
;-----
; unsigned _q15_div(unsigned nom,unsigned
denom)
;-----
; Function: arcus tanges, for fractional q15
format
;
; Function: positive fractional q15 division
; range from 0000h .. 7FFFF
; min: -1.0 <--> 8000h
; max: 0.9999 <--> 7FFFh
;
; Arguments: unsigned nominator: q15 [0000h -
7FFFFh]
; unsigned denominator: q15 > nominator
; --> !!! check beforehand !!!
;
; Return value: dividend q15 <--> 0000h - 7FFFh
;-----
;-----
.def _q15_atan ;define global
.text
_q15_atan:
;context save
;-----
popd *+ ;push return address
sar AR0, *+ ;push old frame pointer
sar AR1, *
lar AR0, * ;init new frame pointer
lar AR2, * ,AR2
sbrk #2 ;AR2 points to first parameter
;division for POSITIV fractional numbers
;-----
lacc *- ,16
rpt #14
subc *
;ACCL = quotient
;context restore
;-----
mar *_ ,AR1
sbrk #1 ;pop local var's + 1 from stack
pshd * ;restore return address
ret
;-----
; Name: Q15_ATAN.ASM
; Project: Q15.LIB
; Function: int q15_atan(int)
; unsigned q15p_atan(unsigned)
; Status:
;
; Target: TMS320C31
;-----
; Name: Q15_DIV.ASM
; Project: Q15.LIB
;-----
include "q15_atan.inc";lookup table
;-----
; int q15_atan(int);
;-----
; unsigned _q15_div(unsigned nom,unsigned
denom)
;-----
; Function: arcus tanges, for fractional q15
format
;
; Function: positive fractional q15 division
; range from 0000h .. 7FFFF
; min: -1.0 <--> 8000h
; max: 0.9999 <--> 7FFFh
;
; Arguments: unsigned nominator: q15 [0000h -
7FFFFh]
; unsigned denominator: q15 > nominator
; --> !!! check beforehand !!!
;
; Return value: dividend q15 <--> 0000h - 7FFFh
;-----
;-----
.def _q15_atan ;define global
.text
_q15_atan:
;context save
;-----
popd *+ ;push return address
sar AR0, *+ ;push old frame pointer
sar AR1, *
lar AR0, * ;init new frame pointer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกิจกรรมงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาตจากฝ่ายวิชาการ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

adrk #3 ;allocate space for two
;local variables
mar *,AR2
lar AR2,#-3
mar *0+ ;AR2 = &parameter
;check if negative or -1
;-----
lacc *
adrk #3
sac1 * ;local #1 = abs(parameter)
bend OK,GEQ
sub #8000h
bend MINUS_1,EQ
lacc *
neg
sac1 * ;local #1 = abs(parameter)
;calcuete atan for POSITIV fractional numbers
;-----
OK: lacc *+,tablen_lg2+1
;lookup table length = 2^tablen_lg2
sach * ;local #2 = first table address
lacc #table
add *
tblr *+ ;local #2 = first value
add #1
tblr * ;local #3 = second value
lacc *-
sub *+ ;ACC = difference = local #3 - #2
sac1 * ;local #3 = difference
lt * ;T = difference
sbrk #2 ;AR2 points to local #1
lacc *,tablen_lg2 ;ACC = local #1 << tablen_lg2
and #7FFFh ;make distance positiv value
sac1 * ;local #1 = distance
mpy *+ ;diffrence * distance
spm 1
lacc *-,16 ;ACCH = local #2 = first value
apac ;ACC += distance * difference
sach * ;local #1 = 'positiv' result
lacc * ;put into lower ACC
;context restore
;-----
spm 0 ;default 'C' setting
mar *,AR1
sbrk #(3+1) ;pop local var's+1 from stack
lar ar0, *- ;restore old frame pointer
pshd * ;restore return address
ret
;-----
; unsigned q15p_atan(unsigned);
;-----
-
; Function: arcus tanges, for positiv fractional
q15 format
;
; Arguments: fractional q15
; min: 9.0 <-> 0000h
; max: 0.9999 <-> 7FFFh
; Return value: scaled angle (0 .. PI/4)
; scaling: PI (e.g. atan(1.0) = 0.25 or 2000h)
; Error: < 2 LSB (128 point lookup table)
;-----
Q15_ATAN.INC
def_q15p_atan ;define global
;-----
.text
;atan(x), with 0 <= x <= 1
;-----
; output scalingfactor = PI
;-----
tblen_lg2 .set 7
.text
;-----
; atan lookup table
;-----
table:
.word 0
.word 81
.word 163
.word 244
.word 326
.word 407
.word 489
.word 570
.word 651
.word 732
.word 813
.word 894
.word 975
adrk #3 ;allocate space for three local variables
mar *,AR2
lar AR2,#-3
mar *0+ ;AR2 = &parameter
;local #1 = parameter
;-----
lacc *
adrk #3
sac1 * ;local #1 = parameter
;calcuete atan for POSITIV fractional numbers
;-----
lacc *+,tablen_lg2+1
;lookup table length = 2^tablen_lg2
sach * ;local #2 = first table address
lacc #table

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.word 1056	.word 4836	.word 7635
.word 1136	.word 4901	.word 7679
.word 1217	.word 4966	.word 7724
.word 1297	.word 5030	.word 7768
.word 1377	.word 5094	.word 7812
.word 1457	.word 5157	.word 7856
.word 1537	.word 5220	.word 7899
.word 1617	.word 5282	.word 7942
.word 1696	.word 5344	.word 7984
.word 1775	.word 5406	.word 8026
.word 1854	.word 5467	.word 8068
.word 1933	.word 5528	.word 8110
.word 2012	.word 5589	.word 8151
.word 2090	.word 5649	.word 8192
.word 2168	.word 5708	ENCODER.H
.word 2246	.word 5768	/*=====*/
.word 2324	.word 5826	/* Name: ENCODER.H */
.word 2401	.word 5885	/* Project: ENCODER */
.word 2478	.word 5943	/* Originator: Keitanun Sukchu */
.word 2555	.word 6000	/* Description: Header File for ENCODER.C */
.word 2632	.word 6058	/*=====*/
.word 2708	.word 6114	#ifndef __ENCODER_H
.word 2784	.word 6171	#define __ENCODER_H
.word 2860	.word 6227	/*=====*/
.word 2935	.word 6282	/* USER SETTABLE ENCODER
.word 3010	.word 6337	CONSTANTS */
.word 3085	.word 6392	/*=====*/
.word 3159	.word 6446	/*=====*/
.word 3233	.word 6500	/* 1) Incremental resolution */
.word 3307	.word 6554	/*=====*/
.word 3380	.word 6607	#define QEP_ROLLOVER 8192 /* incremental
.word 3453	.word 6660	counts */
.word 3526	.word 6712	asm("QEP_ROLLOVER .set 8192 "); /* dito,
.word 3599	.word 6764	assembler support */
.word 3670	.word 6815	/* #define QEP_POWER2 */ /* undefine if
.word 3742	.word 6867	incremental */
.word 3813	.word 6917	/* count isn't a power of 2 */
.word 3884	.word 6968	/*=====*/
.word 3955	.word 7018	/* Pin ADCIN5 (ADC0, channel 5) selected for
.word 4025	.word 7068	U0 */
.word 4095	.word 7117	/* Pin ADCIN13 (ADC1, channel 5) selected for
.word 4164	.word 7166	U90 */
.word 4233	.word 7214	#define ENC_U0_OFFSET (0x8000 + 0) /* U0
.word 4302	.word 7262	(sin) offset */
.word 4370	.word 7310	#define ENC_U90_OFFSET (0x8000 + 300) /*
.word 4438	.word 7358	U90 (-cos) offset */
.word 4505	.word 7405	/*=====*/
.word 4572	.word 7451	/* OTHER CONSTANTS */
.word 4639	.word 7498	/*=====*/
.word 4705	.word 7544	#define PI 32768
.word 4771	.word 7589	#define ENABLE_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define DISABLE 0
/*-----*/
/* Global Variables */
/*-----*/
extern volatile unsigned encoder_position[2];
/*-----*/
/* Function Declaration */
/*-----*/
extern void Encoder_Init(unsigned rollover);
extern void Encoder_ZeroPosition(void);
extern void Encoder_MatchIncrPhase(void);
extern int Encoder_CalcPhase(int sin,int ncos);
extern unsigned Encoder_SamplePosition(void);
extern void Encoder_CalcPosition(unsigned
qep_incr_sample);
#endif
EVM_QEP.H
/*-----*/
/* Name: EVM_QEP.H */
/* Project: ENCODER */
/* Originator: Keitanun Sukchu */
#ifndef __EVM_QEP_H__
#define __EVM_QEP_H__
extern volatile unsigned qep_rollover;
extern volatile int qep_diff;
unsigned QEP_GetIncr(void);
void QEP_Init(unsigned zero_position, unsigned
rollover);
#endif
/* Name: Q15_ATAN.H */
/*-----*/
int q15_atan(int);
/*-----*/
---*/
Function: arcus tanges, for unsigned fractional
q15 format
Arguments: unsigned fractional q15
min: 0 <--> 0000h
max: 0.9999 <--> 7FFFh
Return value: scaled angle (0.. PI/4)
scaling: PI (e.g. atan(1.0) = 0.25 or 2000h)
Error: < 2 LSB (128 point lookup table)
/*-----*/
*/
#ifndef __Q15_ATAN_H__
#define __Q15_ATAN_H__
unsigned q15p_atan(unsigned value); /* dito, but
only positive q15 */
#endif
Q15_DIV.H
/*-----*/
/* Name: Q15_DIV.H */
/* Project: ENCODER */
/* Originator: Keitanun Sukchu */
/* Description: ADC Converter 'C' Marcos */
/*-----*/
#ifndef __ADC_H__
#define __ADC_H__
/*-----*/
/* Include files */
#include "TLC32040.h"
/* Macros */
/* select ADC0/1 channel and start ADC's */
#define ADC_START(ch_0_7,ch_8_15) \
{ \
ADCTRL1 &= 0xFF81; \
ADCTRL1 |= ((ch_0_7<<1) | ((ch_8_15-8)<<4) \
| 0x0001); \
}
/* read ADC0 & 1 value */
#define ADC_READ2(val0,val1) \
{ \
while(!(ADCTRL1 & 0x0100)); \
val0 = ADCFIFO1; \
val1 = ADCFIFO2; \
ADCTRL1 |= 0x0100; } ___an(unsigned value);
/* dito,
*/
#endif
TLC32040.H
/* Name: TLC32040.H */
/* Project: TLC32040.LIB */
/* Description: TMS320C31 memory mapped
registers */
#ifndef __TLC32040_H__
#define __TLC32040_H__
/* Memory Mapped Register */
#define IMR *(volatile unsigned int*) 0x0004
#define IFR *(volatile unsigned int*) 0x0006
/* Watchdog and Real time Interrupt Control
Registers */
#define RTICNTR *(volatile unsigned int*)
0x7021
#define WDCNTR *(volatile unsigned int*)
0x7023
#define WDKEY *(volatile unsigned int*)
0x7025
#define RTICR *(volatile unsigned int*) 0x7027
#define WDCR *(volatile unsigned int*) 0x7029
/* PLL Clock Registers */
#define CKCR0 *(volatile unsigned int*)
0x702B
#define CKCR1 *(volatile unsigned int*)
0x702D
/* Output Logic */
#define OCRA *(volatile unsigned int*) 0x7090
#define OCRB *(volatile unsigned int*) 0x7092
#define PADATDIR *(volatile unsigned int*)
0x7098
#define PBDATDIR *(volatile unsigned int*)
0x709A
#define PCDATDIR *(volatile unsigned int*)
0x709C
/* Definitions for SCI Module */
#define SCICCR *(volatile unsigned int*)0x7050
/* SCI comms.ctrl. reg */#define SCICTL1 *
(volatile unsigned int*)0x7051 /* SCI control
register */
#define SCIHBAUD *(volatile unsigned int*)
0x7052 /* Baud rateselect MSB */
#define SCILBAUD *(volatile unsigned int*)
0x7053 /* Baud rateselect LSB */
#define SCICTL2 *(volatile unsigned int*)
0x7054 /* Xmit int. ctrl& status reg*/
#define SCIRXST *(volatile unsigned int*)
0x7055 /* RCV ctrl andstatus reg*/
#define SCIRXEMU *(volatile unsigned int*)
0x7056 /* Receiver databuffer */
#define SCIRXBUF *(volatile unsigned int*)
0x7057 /* Transmit data buffer */
#define SCITXBUF *(volatile unsigned int*)
0x7059 /* Transmit data buffer*/

```

เอกสารที่สงวนไว้สำหรับนักศึกษาเท่านั้น ไม่

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define SCIPC2 *(volatile unsigned int*)0x705E   #define SCMPR3 *(volatile unsigned int*)
/* Port ctrl register #2 */                   0x741C
#define SCIPRI *(volatile unsigned int*)0x705F
/* Int. priority ctrl reg */                  #define CAPCON *(volatile unsigned int*)
/* Definitions for ADC Module */              0x7420
/* structures are computed uncorrectly by the C
Compiler v6.60 ! */                           #define CAPFIFO*(volatile unsigned int*)
/* therefore following approach has to be used */ 0x7422
#define ADCTRL1 *(volatile unsigned int*)
0x7032 /* ADC Control reg 1 */                #define CAP1FIFO*(volatile unsigned int*)
#define ADCTRL2 *(volatile unsigned int*)       0x7423
0x7034 /* ADC Control reg 2 */
#define ADCFIFO1 *(volatile unsigned int*)
0x7036 /* ADC1 result FIFO */                 #define CAP2FIFO*(volatile unsigned
int*0x7424
#define ADCFIFO2 *(volatile unsigned int*)
0x7038 /* ADC2 result module */               #define CAP3FIFO*(volatile unsigned
int*0x7425
/* structures are computed uncorrectly by the C
Compiler v6.60 ! */                           #define CAP4FIFO*(volatile unsigned
int*0x7426
#define GPTCON *(volatile unsigned
int*0x7400
#define T1CNT *(volatile unsigned int*) 0x7401 #define EVIMRA *(volatile unsigned
int*0x742C
#define T1CMPR *(volatile unsigned int*)
0x7402
#define T1PR *(volatile unsigned int*) 0x7403   fine EVIMRB *(volatile unsigned int*0x742D
#define T1CON *(volatile unsigned int*) 0x7404
#define T2CNT *(volatile unsigned int*) 0x7405 #define EVIMRC *(volatile unsigned
int*0x742E
#define T2CMPR *(volatile unsigned int*)
0x7406
#define T2PR *(volatile unsigned int*) 0x7407   #define EVIFRA *(volatile unsigned int*)
0x742F
#define T2CON *(volatile unsigned int*) 0x7408
#define T3CNT *(volatile unsigned int*) 0x7409 #define EVIFRB *(volatile unsigned int*)
0x7430
#define T3CMPR *(volatile unsigned
int*0x740A
T3CON *(volatile unsigned int*) 0x740C
#define COMCON*(volatile unsigned
int*0x7411
#define ACTR *(volatile unsigned int*) 0x7413   #define EVTIVRA *(volatile unsigned int*)
0x7432
#define SACTR *(volatile unsigned int*) 0x7414
#define DBTCON *(volatile unsigned
int*0x7415
#define CMPR1 *(volatile unsigned int*)
0x7417
#define CMPR2 *(volatile unsigned int*)
0x7418
#define CMPR3 *(volatile unsigned int*)
0x7419
#define SCMPR1 *(volatile unsigned int*)
0x741A
#define SCMPR2 *(volatile unsigned int*)
0x741B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลงานทางวิชาการขณะศึกษาระดับบัณฑิตวิทยาลัย

- (1) เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร “ การสร้างสัญญาณด้วยสัญญาณเสียงที่มีการสูญเสียไปขึ้นมาใหม่” คอมพิวเตอร์ ISSN 0125-2100 ปีที่ 21, ฉบับที่ 114, กรกฎาคม-สิงหาคม 2538. หน้า 49-55.
- (2) เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร, “ การสร้างเครื่องตรวจสอบคุณภาพของน้ำอย่างต่อเนื่อง” การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร , พฤศจิกายน 2538, หน้า 826 –813
- (3) เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร “ การออกแบบตัวควบคุม Invert Pendulum สำหรับระบบที่เป็นเชิงเส้น ” วิศวกรรม ลาดกระบัง ปีที่ 13, ฉบับ 1, กรกฎาคม 2539, หน้า 131-136
- (4) เกียรติอนันต์ สุขชู, ปราโมทย์ พานกุลลาบ, วิทยา ทิพย์สุวรรณพร, วิริยะ กองรัตน์ “การประยุกต์ใช้ตัวควบคุม H_{∞} สำหรับ Inverted Pendulum โดยแรงเสียดทานที่ไม่แน่นอน” วิศวกรรม ลาดกระบัง ปีที่ 14, ฉบับที่ 1, กรกฎาคม 2541, หน้า 35 –34
- (5) วิทยา ทิพย์สุวรรณพร, สุพรรณ กุลพาณิชย์, เกียรติอนันต์ สุขชู “ การสร้างตัวควบคุม PIDA สำหรับควบคุมความเร็วมอเตอร์กระแสตรงแบบแยกตัวกระตุ้น” วิศวกรรมลาดกระบัง ปีที่ 14, ฉบับที่ 2, มิถุนายน 2541. หน้า 49 –57.
- (6) เกียรติอนันต์ สุขชู, วิทยา ทิพย์สุวรรณพร, ชัยณรงค์ ไพรพฤษ์, กฤษชัย วิถีพานิช, “การออกแบบตัวควบคุม PIDA สำหรับการควบคุมตำแหน่งมอเตอร์เหนี่ยวนำกระแสสลับ”การประชุมนววิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 21 วันที่ 12 –13, พฤศจิกายน 2541. CT016 หน้า 413 -416
ณ. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ถ.สุขสวัสดิ์ 48 เขตทุ่งครุ กรุงเทพมหานคร

ประวัติผู้เขียน

นายเกียรติอนันต์ สุขชู เกิดวันที่ 3 ธันวาคม 2508 ที่จังหวัดนครปฐม สำเร็จการศึกษาอุตสาหกรรมศาสตรบัณฑิต (โทรคมนาคม) จากมหาวิทยาลัยสยาม เมื่อ 24 พฤษภาคม 2536 และประกาศนียบัตร วิชาชีพชั้นสูง (ช่างไฟฟ้า) จากวิทยาลัยเทคนิคนครปฐม เมื่อ 23 มีนาคม 2533

ปี 2539 เป็นอาจารย์พิเศษ สอนวิชาคอมพิวเตอร์ และการประยุกต์ใช้งาน ที่โรงเรียนเทคโนโลยีกรุงเทพฯ จนถึงปัจจุบัน และขณะศึกษาเมื่อปี 2539-2540 เป็นผู้ช่วยนักวิจัย ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมคณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์ทำงานที่บริษัท คอมพิวเตอร์ซิสเต็มคอนเนกชั่น (C & C CO.LTD.) เมื่อปี 2533-2536 ตำแหน่งช่างเทคนิค

