

การชดเชยการเสียดทานและสภาพไม่เป็นเชิงเส้นในฮาร์ดดิสก์ไดรฟ์ขนาดจิ๋ว^{*}
ด้วยฟิสิกส์ลอจิก

นางสาววิไลพร เงินบาท

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2551

**FUZZY LOGIC COMPENSATION FOR NONLINEARITIES
AND FRICTION IN A MICRO
HARD DISK DRIVE**

Wilaiporn Ngernbaht

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Electrical Engineering**

Suranaree University of Technology

Academic Year 2008

การขออนุญาตเสียดทานและสภาพไม่เป็นเชิงเส้นในฮาร์ดดิสก์ไดรฟ์ขนาดจิ๋ว
ด้วยฟิสิกส์ลอจิก

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรปริญญาโทบริหารธุรกิจ

คณะกรรมการสอบวิทยานิพนธ์

(รศ. ดร.กิตติ อรรถกิจมงคล)

ประธานกรรมการ

(ศ. น.ท. ดร.สราวุฒิ สุจิตกร)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

(อ. ดร.ทองพล อารีรักษ์)

กรรมการ

(ศ. ดร.ไพโรจน์ สัตยธรรม)

รองอธิการบดีฝ่ายวิชาการ

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

วิไลพร เงินบาท : การชดเชยการเสียดทานและสภาพไม่เป็นเชิงเส้นในฮาร์ดดิสก์ไดรฟ์ขนาดเล็ก
ด้วยฟัซซีลอจิก (FUZZY LOGIC COMPENSATION FOR NONLINEARITIES
AND FRICTION IN A MICRO HARD DISK DRIVE) อาจารย์ที่ปรึกษา :
ศาสตราจารย์ นาวาอากาศโท ดร. สราวุธ สุจิตจร, 108 หน้า

งานวิจัยวิทยานิพนธ์นี้ มีความมุ่งหมายในการลดค่าผิดพลาดทางการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ขนาดเล็ก เนื่องจากสภาพไม่เป็นเชิงเส้นและความเสียดทานของตัวกระตุ้นชนิด voice-coil-motor (VCM) ซึ่งทำหน้าที่เป็นตัวกระตุ้นหัวอ่าน การปรับปรุงสมรรถนะให้ดีขึ้นได้ใช้ตัวควบคุมแบบฟัซซีลอจิก นอกจากนั้น สภาวะรีโซแนนซ์ในย่านความถี่สูงของหัวอ่านได้รับการแก้ปัญห ด้วยการชดเชยเชิงเส้น เนื้อหาของวิทยานิพนธ์ประกอบด้วย การศึกษาพลวัตและแบบจำลองทางคณิตศาสตร์ของหัวอ่านที่กระตุ้นด้วย VCM ในย่านความถี่ต่ำและความถี่สูง การจำลองสถานการณ์ การออกแบบตัวควบคุมฟัซซีลอจิกเพื่อชดเชยความไม่เป็นเชิงเส้น การออกแบบการชดเชยเชิงเส้นเพื่อสร้างเสถียรภาพและปรับปรุงสมรรถนะการทำงานของหัวอ่านฮาร์ดดิสก์ไดรฟ์ในย่านความถี่สูง ตลอดจนทำการออฟไลน์สมรรถนะเชิงเส้นด้วยการค้นหาแบบตามูเชิงปรับตัว สำหรับการทำงานของหัวอ่านในย่านความถี่สูง

WILAIORN NGERNBAHT : FUZZY LOGIC COMPENSATION FOR
NONLINEARITIES AND FRICTION IN A MICRO HARD DISK DRIVE.

THESIS ADVISOR : PROF. WNG. CMDR. SARAWUT SUJITJORN, Ph.D.,
108 PP.

R/W HEAD/MOTION/FUZZY COMPENSATION/ADAPTIVE TABU SEARCH

This research thesis aims to reduce the motion errors of a read/write head of a micro hard drive. The errors are due to nonlinearity and friction in the voice-coil-motor (VCM) operating as an actuator of the head. Performance improvement is achieved by using a fuzzy logic controller. Moreover, the high-frequency resonance of the head are resolved by the linear compensation approach. This thesis includes studies of dynamic and mathematical model of the head actuated by a VCM in low- and high-frequency ranges, simulations, design of a fuzzy logic controller compensating for the nonlinearity, as well as linear compensation design for head stabilization and performance enhancement in high-frequency range. In addition, performance optimization is achieved via the adaptive tabu search (ATS) for high-frequency operation of the head.

กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณบุคคล และกลุ่มบุคคลต่อไปนี้ ที่ได้กรุณาให้คำปรึกษา แนะนำและช่วยเหลืออย่างยิ่ง ทั้งในด้านวิชาการและด้านการดำเนินงานวิจัย

ศาสตราจารย์ นาวาอากาศโท ดร.สรายุทธ สุจิตจร อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่เมตตาให้การอบรม สั่งสอน ชี้แนะแนวทางอันเป็นประโยชน์ยิ่งต่องานวิจัย รวมถึงได้ช่วยตรวจทาน และแก้ไขรายงานวิทยานิพนธ์เล่มนี้จนทำให้มีความสมบูรณ์ยิ่งขึ้น ช่วยเหลือในการทำการศึกษาวิจัย ตลอดจนให้คำแนะนำในการเขียน และตรวจแก้ไขวิทยานิพนธ์จนเสร็จสมบูรณ์

อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารีทุกท่าน รวมถึงผู้ช่วยศาสตราจารย์เดชา พวงดาวเรือง ที่ได้กรุณาให้คำปรึกษา และแนะนำความรู้ทางวิชาการอย่างดียิ่งมาโดยตลอด

ขอบคุณคุณอัญชลี รักด่านกลาง และคุณภัทรวรรณ สิริทิกวินกุล ที่ให้ความช่วยเหลือ ดูแลเรื่องเอกสาร และอำนวยความสะดวกให้กับผู้วิจัยตลอดระยะเวลาที่ศึกษา

ขอบคุณ พี่ ๆ เพื่อน ๆ น้อง ๆ บัณฑิตศึกษาทุกท่าน ที่ให้คำแนะนำในการทำวิจัย ให้ข้อมูลที่เป็นประโยชน์ต่อผู้วิจัย และขอบคุณ

ท้ายนี้ผู้วิจัยขอกราบขอบพระคุณคุณพ่อสำราญ คุณแม่หลีฟ้า พี่วลี พี่สุภารัตน์ รวมถึงญาติพี่น้องของผู้วิจัยทุกท่านที่ได้ให้ความรัก ความอบอุ่น ความห่วงใย การอบรมเลี้ยงดู ให้กำลังใจในยามที่ผู้วิจัยท้อแท้ และให้การสนับสนุนทางด้านการศึกษายิ่งมาโดยตลอด

วิไลพร เงินบาท

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช

บทที่

1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของเบื้องต้น.....	2
1.4 ขอบเขตของการวิจัย.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.7 การจัดทำรูปเล่มวิทยานิพนธ์.....	3
2 แบบจำลองทางคณิตศาสตร์ของหัวอ่านฮาร์ดดิสก์ไครฟ์.....	4
2.1 บทนำ.....	4
2.2 แบบจำลองของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่ต่ำ.....	5
2.3 แบบจำลองของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่สูง.....	7
2.4 สรุป.....	9
3 การปรับปรุงเสถียรภาพและสมรรถนะของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่ต่ำ.....	10
3.1 บทนำ.....	10
3.2 ความรู้เบื้องต้นเกี่ยวกับฟิสิกส์ลอจิก.....	11
3.3 การออกแบบตัวควบคุมฟิสิกส์แบบกฎสองอินพุต.....	14
3.4 การอนุมานกฎฟิสิกส์.....	16
3.5 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ ด้วยตัวควบคุมฟิสิกส์.....	19

สารบัญ (ต่อ)

หน้า

3.6	การปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก.....	20
3.7	การเพิ่มการแบ่งส่วนตัวแปรฟuzzy.....	23
3.8	การออกแบบตัวควบคุมฟuzzyแบบกฎสองชุด.....	25
3.9	สรุป.....	30
4	การปรับปรุงเสถียรภาพและสมรรถนะของหัวอ่านฮาร์ดดิสก์ไครฟ์ย่านความถี่สูง.....	32
4.1	บทนำ.....	32
4.2	การใช้ตัวชดเชยแบบเฟสล้ำหน้า.....	32
4.3	การใช้ตัวชดเชยแบบเฟสล้ำหน้าล้ำหลังเชิงซ้อน.....	37
4.4	การใช้ตัวชดเชยแบบเฟสล้ำหน้าล้ำหลังเชิงซ้อน ตามวิธีของ Messner และคณะเมื่อใช้ค่า $\zeta > 1$	40
4.5	สรุป.....	44
5	การหาค่าเหมาะที่สุดด้วยการค้นหาแบบตามูเชิงปรับตัวสำหรับตัวชดเชยรีโซแนนซ์.....	45
5.1	บทนำ.....	45
5.2	การค้นหาแบบตามูเชิงปรับตัว.....	45
5.3	การหาค่าเหมาะที่สุดสำหรับตัวชดเชยรีโซแนนซ์.....	50
5.4	สรุป.....	53
6	บทสรุปและข้อเสนอแนะ.....	54
6.1	สรุป.....	54
6.2	ข้อเสนอแนะ.....	55
	รายการอ้างอิง.....	56
	ภาคผนวก	
	ภาคผนวก ก. รายละเอียดโปรแกรมสำหรับแบบจำลองของระบบการเคลื่อนที่ ของหัวอ่านฮาร์ดดิสก์ไครฟ์.....	58
	ภาคผนวก ข. รายละเอียดโปรแกรมตัวควบคุมฟuzzyแบบกฎสองอินพุต.....	62
	ภาคผนวก ค. รายละเอียดโปรแกรม ATS.....	100
	ภาคผนวก ง. การเผยแพร่ผลงานขณะศึกษา.....	106
	ประวัติผู้เขียน.....	108

สารบัญตาราง

ตารางที่	หน้า
3.1	กฎฟิสิกส์.....14
3.2	กฎฟิสิกส์.....24
4.1	ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับค่า K กรณีมีการชดเชยด้วยเฟสล้ำหน้า.....36
4.2	ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับค่า K กรณีมีการชดเชยด้วยเฟสล้ำหน้าและล้ำหลังเชิงซ้อน.....40
4.3	ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับค่า K กรณีมีการชดเชยด้วยเฟสล้ำหน้าและล้ำหลังอันดับสี่แบบห่วงเงิน.....43

สารบัญรูป

รูปที่	หน้า
1.1 ส่วนประกอบฮาร์ดดิस्कไดรฟ์.....	1
2.1 ส่วนประกอบฮาร์ดดิस्कไดรฟ์.....	4
2.2 โครงสร้างทางกลของ VCM.....	5
2.3 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน โดยมีอินพุตค่าต่าง ๆ.....	7
2.4 ผลตอบสนองทางโดเมนเวลาเมื่ออินพุตเป็นขั้นบันไดหนึ่งหน่วย.....	8
2.5 ผลตอบสนองทางโดเมนความถี่ (bode plot).....	9
3.1 ความแตกต่างของความเป็นสมาชิกระหว่างเซตธรรมดา กับ ฟัซซีเซต.....	12
3.2 โครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซี.....	12
3.3 ฟังก์ชันความเป็นสมาชิกของอินพุต.....	15
3.4 การรวมกฎแบบ Sugeno.....	17
3.5 การอนุมานกฎฟัซซีแบบ Sugeno.....	18
3.6 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน กรณีมีการควบคุมด้วยวิธีต่าง ๆ และกรณีไม่มีการควบคุม.....	20
3.7 ฟังก์ชันความเป็นสมาชิกของอินพุต.....	21
3.8 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน.....	22
3.9 ฟังก์ชันความเป็นสมาชิกของอินพุต.....	23
3.10 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน.....	24
3.11 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านกรณีมีการควบคุมแบบต่าง ๆ.....	26
3.12 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านเมื่อเคลื่อนที่ไปครั้งละ $1\ \mu m$	28
3.13 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านเมื่อเคลื่อนที่ไปครั้งละ $1\ \mu m$	29
3.14 สัญญาณการควบคุม.....	30
4.1 โครงสร้างของระบบที่มีการชดเชย.....	33
4.2 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วยเฟสล้ำหน้า เมื่อปรับค่า K	35
4.3 แผนภาพโบดของระบบวงเปิดกรณีมีการชดเชยด้วยเฟสล้ำหน้า เมื่อปรับค่า K	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วยเฟสล้ำหน้า เมื่อปรับจูนค่า K.....	36
4.5 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วย เฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K.....	38
4.6 แผนภาพโพลของระบบวงเปิดกรณีมีการชดเชยด้วย เฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K.....	39
4.7 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วย เฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K.....	39
4.8 ผลการตอบสนองทางเวลาของระบบของระบบกรณีมีการชดเชยแบบ เฟสล้ำหน้าและล้ำหลังอันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K.....	42
4.9 แผนภาพโพลของระบบวงเปิดกรณีมีการชดเชยแบบเฟสล้ำหน้าและล้ำหลัง อันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K.....	42
4.10 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยแบบเฟสล้ำหน้า และล้ำหลังอันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K.....	43
5.1 แผนภูมิการดำเนินงานของของการค้นหาแบบตามูเชิงปรับตัว.....	48
5.2 แผนภูมิการทำงานของกลไกย้อนรอยการค้นหา.....	49
5.3 แผนภูมิการทำงานของกลไกปรับรัศมีการค้นหา.....	49
5.4 การคำนวณค่า cost.....	50
5.5 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วยตัวชดเชยจาก การค้นหาแบบตามูเชิงปรับตัว.....	51
5.6 แผนภาพโพลของระบบวงเปิดกรณีมีการชดเชยด้วยตัวชดเชยจาก การค้นหาแบบตามูเชิงปรับตัว.....	52
5.7 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วยตัวชดเชยจาก การค้นหาแบบตามูเชิงปรับตัว.....	52

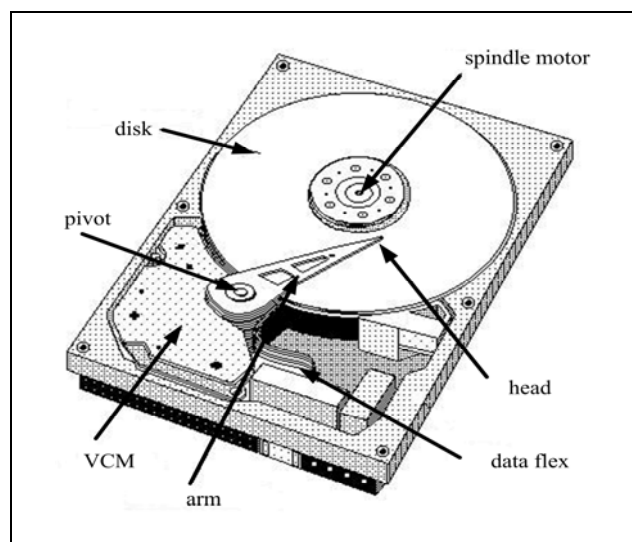
บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อุตสาหกรรมฮาร์ดดิสก์ ในปัจจุบันมีความต้องการที่จะผลิตฮาร์ดดิสก์ไดรฟ์ให้มีขนาดเล็กลง และอ่านข้อมูลได้เร็วขึ้น อีกทั้งราคาของฮาร์ดดิสก์ไดรฟ์ต้องถูกลงด้วย วิธีการหนึ่งสำหรับเพิ่มอัตราการอ่านข้อมูลของฮาร์ดดิสก์ไดรฟ์ให้เร็วขึ้น คือ การลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านฮาร์ดดิสก์ไดรฟ์เมื่อเคลื่อนตัวไปตามแทร็ก (track) ต่าง ๆ เพื่ออ่านข้อมูล (Peng, Chen, Cheng, and Lee, 2005) โดยความผิดพลาดดังกล่าวมีสาเหตุมาจากสภาพไม่เป็นเชิงเส้นและความเสียดทานของตัวกระตุ้นชนิด voice-coil-motor (VCM) ซึ่งทำหน้าที่เป็นตัวขับเคลื่อนหัวอ่านฮาร์ดดิสก์ไดรฟ์

VCM เป็นมอเตอร์กระแสตรงชนิดหนึ่ง ประกอบด้วย ส่วนที่เคลื่อนที่ได้ เรียกว่าโรเตอร์ และส่วนที่อยู่นิ่งกับที่เรียกว่า สเตเตอร์ สเตเตอร์นั้นประกอบด้วยแม่เหล็กถาวรเหมือนกับมอเตอร์กระแสตรง ส่วนโรเตอร์นั้นประกอบด้วยขดลวด แกนหมุน และแขนโลหะ มีสายเคเบิลข้อมูลเชื่อมต่อระหว่างหัวอ่านกับแกนโลหะ ส่วนประกอบฮาร์ดดิสก์ไดรฟ์แสดงในรูปที่ 1.1



รูปที่ 1.1 ส่วนประกอบฮาร์ดดิสก์ไดรฟ์

เนื่องจากการควบคุมหัวอ่านฮาร์ดดิสก์โดยปกติดำเนินงานทั้งในย่านความถี่ต่ำและความถี่สูง การควบคุมในย่านความถี่ต่ำมีประเด็นสำคัญเป็นการแก้ปัญหา สภาพไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ เนื่องจากปารองเพลตรองแกนหมุนกับแกนหมุนเกิดการเสียดทานเมื่อหัวอ่านเคลื่อนตัวไปตามแทร็กต่าง ๆ

เพื่ออ่านข้อมูลซึ่งเป็นปัจจัยหลักที่ทำให้เกิดสภาพไม่เป็นเชิงเส้นนอกจากนั้น ยังเกิดจากสายเคเบิลข้อมูลที่เชื่อมต่อระหว่างหัวอ่านกับแขนโลหะอีกด้วย ส่วนในย่านความถี่สูง การควบคุมให้หัวอ่านติดตามแทรกข้อมูล ประสพปัญหาสำคัญจากสภาวะเรโซแนนซ์ และการขาดเสถียรภาพ ผลงานวิจัยที่น่าเสนอในวิทยานิพนธ์นี้ ให้ความสนใจการควบคุมในย่านความถี่สูงของหัวอ่าน ซึ่งมุ่งเน้นการลดทอนผลกระทบจากสภาวะเรโซแนนซ์ และการชดเชยทางเสถียรภาพ และในย่านความถี่ต่ำให้ความสนใจระบบควบคุมเพื่อแก้ปัญหาสภาวะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ

1.2 วัตถุประสงค์ของการวิจัย

- 1) เพื่อพัฒนาวิธีการแก้ปัญหาเนื่องมาจากสภาพไม่เป็นเชิงเส้นและการเสียดทานในฮาร์ดดิสก์ไครฟ์ขนาดจิ๋ว ด้วยตัวควบคุมฟuzzyลอจิก
- 2) เพื่อพัฒนาวิธีการสร้างเสถียรภาพและชดเชยสภาวะเรโซแนนซ์ในย่านความถี่สูงของการทำงานของฮาร์ดดิสก์ไครฟ์ขนาดจิ๋ว

1.3 ข้อตกลงเบื้องต้น

การคำนวณ และการจำลองสถานการณ์ ที่มีความซับซ้อนในงานวิจัยนี้ อาศัยโปรแกรม MATLAB เป็นเครื่องมือในการคำนวณ

1.4 ขอบเขตของการวิจัย

- 1) การออกแบบระบบควบคุมเชิงเส้นเพื่อสร้างเสถียรภาพและปรับปรุงสมรรถนะการทำงานของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่สูง มีข้อกำหนดสมรรถนะดังนี้ การพุ่งเกินไม่เกิน 10% ค่าผิดพลาดในสถานะอยู่ตัวอยู่ในช่วง $\pm 1\%$ ส่วนเฟสมากกว่า 30° และส่วนเฟสอัตรายาย มากกว่า 6dB
- 2) การออกแบบตัวชดเชยป้อนกลับแบบฟuzzyลอจิกแก้ปัญหาค่าไม่เป็นเชิงเส้นตลอดจนความเสียดทานของการทำงานของฮาร์ดดิสก์ไครฟ์ในย่านความถี่ต่ำ โดยมีข้อกำหนดสมรรถนะดังนี้ การพุ่งเกินไม่เกิน 5% ค่าผิดพลาดในสถานะอยู่ตัวอยู่ในช่วง $\pm 1\%$
- 3) การออกแบบเพื่อสร้างเสถียรภาพ จะได้รับการอพติไมซ์ด้วยการค้นหาแบบตามูเชิงปรับตัว

1.5 ขั้นตอนการดำเนินงาน

- 1) ศึกษาและค้นคว้าแบบจำลองของฮาร์ดดิสก์ไครฟ์ขนาดจิ๋วที่ใช้ VCM ทั้งแบบเชิงเส้น และที่ปรากฏสภาวะไม่เป็นเชิงเส้นตลอดจนความเสียดทาน
- 2) จำลองสถานการณ์เพื่อศึกษาพลวัตของฮาร์ดดิสก์ไครฟ์ขนาดจิ๋วที่ใช้ VCM ทั้งแบบเชิงเส้น และที่ปรากฏสภาวะไม่เป็นเชิงเส้นตลอดจนความเสียดทาน

- 3) สืบค้นปริทัศน์วรรณกรรมที่เกี่ยวข้องกับการออกแบบตัวควบคุมฟัซซี
- 4) ศึกษาวิธีการออกแบบตัวควบคุมฟัซซี และดำเนินการออกแบบ
- 5) ออกแบบกฎฟัซซีเพื่อการควบคุม
- 6) จำลองสถานการณ์ที่ได้ออกแบบตัวควบคุมฟัซซีเพื่อศึกษาประสิทธิผลและทำการปรับปรุงในขั้นต้น
- 7) ศึกษาวิธีการชดเชยแบบเชิงเส้น และดำเนินการออกแบบ
- 8) จำลองสถานการณ์ระบบเชิงเส้น เพื่อศึกษาเปรียบเทียบประสิทธิผลของตัวชดเชย
- 9) ศึกษาผลงานวิจัยที่มีปรากฏมาก่อนแล้วซึ่งใช้แนวทาง Composite Nonlinear Feedback (CNF) ชดเชยสถานะไม่เชิงเส้น
- 10) ดำเนินงานอพติไมเซชันด้วยการค้นหาตามูเชิงปรับตัว หรือ ATS

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้ตัวควบคุมแบบฟัซซีที่แก้ปัญหาเนื่องมาจากสภาพไม่เป็นเชิงเส้นและการเลียดทานในฮาร์ดดิสก์ไครฟ์ขนาดเล็ก
- 2) ได้ตัวชดเชยเพื่อสร้างเสถียรภาพและชดเชยสถานะเรโซแนนซ์ในย่านความถี่สูงของการทำงานของฮาร์ดดิสก์ไครฟ์ขนาดเล็ก

1.7 การจัดทำรูปเล่มวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ประกอบด้วย 6 บท บทที่ 1 เป็นบทนำกล่าวถึงความสำคัญของปัญหาวัตถุประสงค์ ข้อตกลงเบื้องต้น ขอบเขตของการวิจัย ขั้นตอนการดำเนินงาน และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย รวมทั้งแนะนำเนื้อหาพอสังเขปที่เป็นองค์ประกอบของวิทยานิพนธ์ฉบับนี้

บทที่ 2 กล่าวถึงการแบบจำลองทางคณิตศาสตร์ของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่ต่ำและความถี่สูง

บทที่ 3 กล่าวถึงระบบควบคุมแบบฟัซซี ทฤษฎีฟัซซีเซต การออกแบบระบบควบคุมแบบฟัซซี

บทที่ 4 กล่าวถึง การออกแบบตัวชดเชยแบบเฟสล้ำหน้า เฟสล้ำหน้าล้ำหลังเชิงซ้อน ตัวชดเชยแบบเฟสล้ำหน้าล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$

บทที่ 5 กล่าวถึงการหาค่าเหมาะที่สุดด้วยการค้นหาแบบตามูเชิงปรับตัว

บทที่ 6 กล่าวถึงบทสรุปและข้อเสนอแนะ

ภาคผนวก ก. รายละเอียดของโปรแกรมสำหรับแบบจำลองของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ใน

ภาคผนวก ข. รายละเอียดของโปรแกรมสำหรับตัวควบคุมฟัซซีแบบกฎสองอินพุต

ภาคผนวก ค. รายละเอียดของโปรแกรม ATS

ภาคผนวก ง. การเผยแพร่ผลงานขณะศึกษา

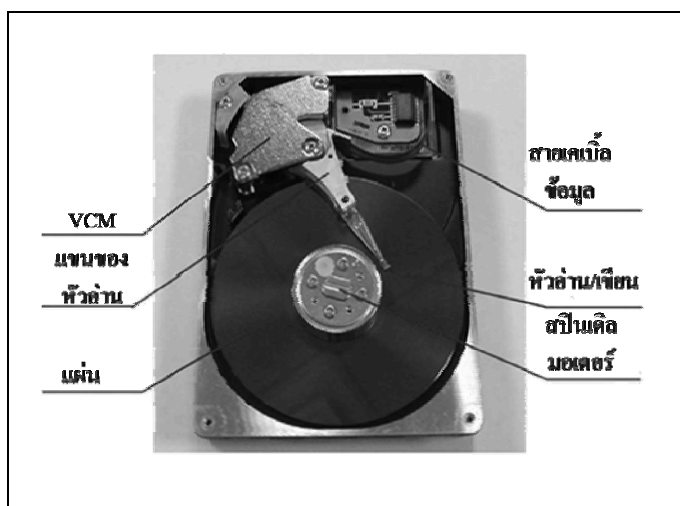
บทที่ 2

แบบจำลองทางคณิตศาสตร์ของหัวอ่านฮาร์ดดิสก์ไดรฟ์

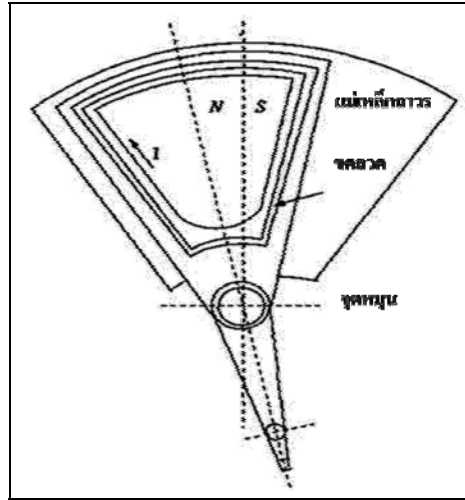
2.1 บทนำ

ในการดำเนินงานวิเคราะห์และออกแบบเพื่อควบคุมระบบ มีความจำเป็นที่ต้องการแบบจำลองทางคณิตศาสตร์ของระบบเพื่อใช้อธิบายพลวัตของกระบวนการที่ควบคุม และใช้ในการออกแบบตัวควบคุมเพื่อให้ได้ผลตอบสนองตามที่ต้องการ ดังนั้นกระบวนการหาแบบจำลองทางคณิตศาสตร์ของระบบจึงเป็นปัจจัยที่สำคัญประการหนึ่ง ในบทที่ 2 นี้จะได้กล่าวถึงระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไดรฟ์ เนื่องจากการควบคุมหัวอ่านฮาร์ดดิสก์โดยปกติดำเนินงานทั้งในย่านความถี่ต่ำและความถี่สูง การควบคุมในย่านความถี่ต่ำมีประเด็นสำคัญเป็นการแก้ปัญหา สภาวะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ ส่วนในย่านความถี่สูง การควบคุมให้หัวอ่านติดตามแทร็กข้อมูล ประสบปัญหาสำคัญจากสภาวะเรโซแนนซ์ และการขาดเสถียรภาพ

การเคลื่อนที่ของหัวอ่านนั้นใช้ตัวกระตุ้นชนิด voice-coil-motor (VCM) ทำหน้าที่เป็นตัวขับเคลื่อนหัวอ่านฮาร์ดดิสก์ไดรฟ์ VCM เป็นมอเตอร์กระแสตรงประกอบด้วย โรเตอร์และสเตเตอร์ โดยสเตเตอร์จะประกอบด้วยแม่เหล็กถาวร ส่วนโรเตอร์ จะประกอบด้วย ขดลวด แกนหมุนและแขนโลหะมีสายเคเบิลข้อมูลเชื่อมต่อระหว่างหัวอ่านกับแขนโลหะ ดังส่วนประกอบที่แสดงในรูปที่ 2.1 และในส่วนโครงสร้างทางกลของ VCM ได้แสดงไว้ในรูปที่ 2.2



รูปที่ 2.1 ส่วนประกอบฮาร์ดดิสก์ไดรฟ์



รูปที่ 2.2 โครงสร้างทางกลของ VCM

2.2 แบบจำลองของหัวอ่านฮาร์ดดิสก์ไดรฟ์ในย่านความถี่ต่ำ

การเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไดรฟ์ ที่ปรากฏสถานะไม่เป็นเชิงเส้นและสภาพความเสียดทาน พบว่าเมื่อหัวอ่านเคลื่อนที่ไปตามแทร็กต่าง ๆ เพื่ออ่านข้อมูลนั้นเกิดการเสียดทานขึ้นระหว่างบารองเพลาตรงแกนหมุนกับแกนหมุน (Chen, Lee, Peng, and Venkataramanan, 2006) โดยแรงเสียดทานที่เกิดขึ้นพิจารณาเป็น แรงเสียดทานคูลอมบ์และแรงเสียดทานสถิตซึ่งแรงเสียดทานที่เกิดขึ้นเป็นปัจจัยสำคัญที่ทำให้เกิดสภาพไม่เป็นเชิงเส้น นอกจากนั้นยังเกิดจากสายเคเบิลข้อมูลที่เชื่อมต่อระหว่างหัวอ่านกับแขนโลหะอีกด้วย พลวัตการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์อาจแทนได้ด้วยแบบจำลองทางคณิตศาสตร์ (Chen, Lee, Peng, and Venkataramanan, 2006) ดังสมการที่ (2-1) ซึ่งฟังก์ชัน T_f ได้อธิบายไว้ในสมการที่ (2-2) ถึงสมการที่ (2-4) ตามลำดับ

$$\ddot{y} = 2.35 \times 10^8 u - 6.7844 \times 10^6 \arctan(0.5886y) + T_f \quad (2-1)$$

$$T_f = \begin{cases} -\left[\left| 1.175 \times 10^6 uy + 0.01(\dot{y})^2 \right| + 15000 \right] \\ \quad \times \text{sgn}(\dot{y}) - 282.6\dot{y}, & \dot{y} \neq 0 \\ -T_e, & \dot{y} = 0, |T_e| \leq T_s \\ T_s \text{sgn}(T_e), & \dot{y} = 0, |T_e| > T_s \end{cases} \quad (2-2)$$

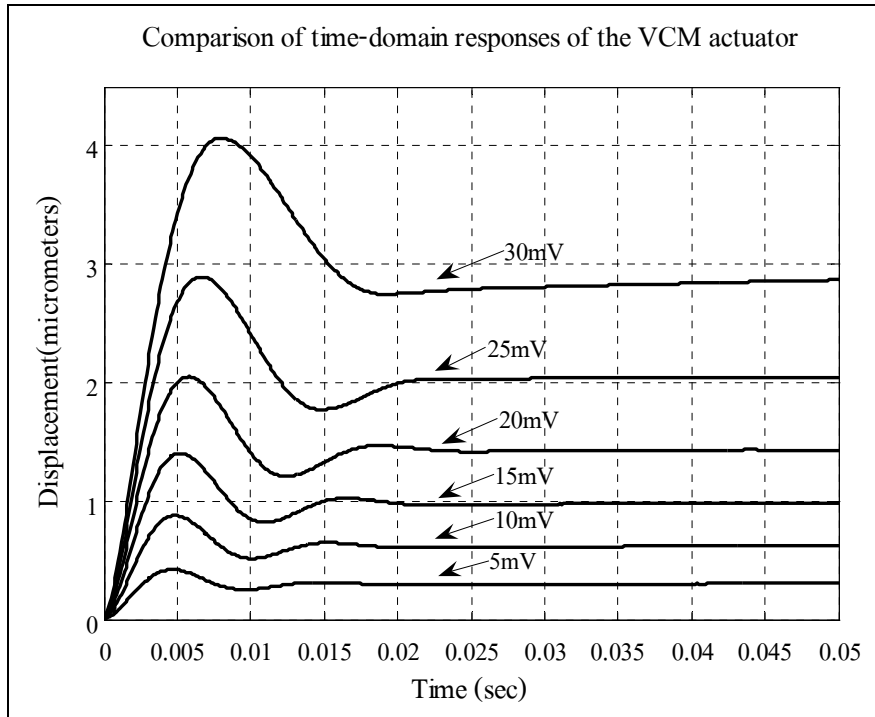
$$T_e = 2.35 \times 10^8 [-0.02887 \arctan(0.5886y) + u] \quad (2-3)$$

$$T_s = 1.293 \times 10^6 |u_0 y_0| + 1.65 \times 10^4 \quad (2-4)$$

โดยที่

T_f	คือ	แรงบิดที่เกิดจากแรงเสียดทาน (Nm)
T_e	คือ	แรงบิดที่เกิดจากภายนอก (external torque) (Nm)
T_s	คือ	แรงบิดที่เกิดจากแรงเสียดทานสถิต (Nm)
u	คือ	แรงดันอินพุต (V)
u_0	คือ	แรงดันอินพุตที่เวลาเริ่มต้น (V)
y	คือ	เอาต์พุตซึ่งเป็นระยะการเคลื่อนที่ของหัวอ่าน (m)
y_0	คือ	ระยะการเคลื่อนที่ของหัวอ่านที่เวลาเริ่มต้น (m)

ดำเนินการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์โดยใช้แบบจำลองดังสมการ (2-1) ป้อนอินพุตเป็นแรงดันไฟฟ้ากระแสตรงที่ขนาด 5 มิลลิโวลต์ 10 มิลลิโวลต์ 15 มิลลิโวลต์ 20 มิลลิโวลต์ 25 มิลลิโวลต์ และ 30 มิลลิโวลต์ ได้ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านดังแสดงในรูปที่ 2.3 จะเห็นว่าเมื่อป้อนอินพุตด้วยขนาดที่มากขึ้น ทำให้หัวอ่านเคลื่อนที่ในระยะที่เพิ่มขึ้นด้วย แต่การเข้าถึงตำแหน่งของหัวอ่านนั้นมีค่าผิดพลาดเกิดขึ้นมาก และมีการพุ่งเกินมากขึ้นสัมพันธ์กับขนาดของสัญญาณอินพุต ซึ่งทำให้อัตราเร็วในการเข้าถึงข้อมูลนั้นลดลง สาเหตุสำคัญที่ทำให้เกิดค่าผิดพลาดดังกล่าวนี้ คือ สภาพไม่เป็นเชิงเส้นจากความอ่อนตัวของสายเคเบิลข้อมูล และความเสียดทานของบารองเพลตาของ VCM



รูปที่ 2.3 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านโดยมีอินพุตค่าต่าง ๆ

2.3 แบบจำลองของหัวอ่านฮาร์ดดิสก์ไดรฟ์ในย่านความถี่สูง

การควบคุมหัวอ่านให้ติดตามแทรกข้อมูล เป็นการปฏิบัติงานในย่านความถี่สูงซึ่งหัวอ่านฮาร์ดดิสก์มีรูปแบบการทำงานอนุมานได้ว่าเป็นเชิงเส้น (Chen, Lee, Peng, and Venkataramanan, 2006) อาจสรุปได้ว่า หัวอ่านฮาร์ดดิสก์มีแบบจำลองทางคณิตศาสตร์ที่เป็นฟังก์ชันถ่ายโอนดังสมการที่ (2-5) อินพุตของฟังก์ชันถ่ายโอนดังกล่าว คือ แรงดันที่ป้อนให้กับ VCM ในขณะที่เอาต์พุต คือ ระยะการเคลื่อนที่ของหัวอ่าน ฟังก์ชัน $G_{r.m.1}(s)$ $G_{r.m.2}(s)$ $G_{r.m.3}(s)$ $G_{r.m.4}(s)$ และ $G_{r.m.5}(s)$ ในสมการที่ (2-5) ได้แสดงไว้ในสมการที่ (2-6) ถึงสมการที่ (2-10) (Chen, Lee, Peng, and Venkataramanan, 2006) ตามลำดับ

$$G(s) = \frac{2.35 \times 10^8}{s^2} G_{r.m.1}(s) G_{r.m.2}(s) G_{r.m.3}(s) G_{r.m.4}(s) G_{r.m.5}(s) \quad (2-5)$$

$$G_{r.m.1}(s) = \frac{0.8709s^2 + 1726s + 1.369 \times 10^9}{s^2 + 1480s + 1.369 \times 10^9} \quad (2-6)$$

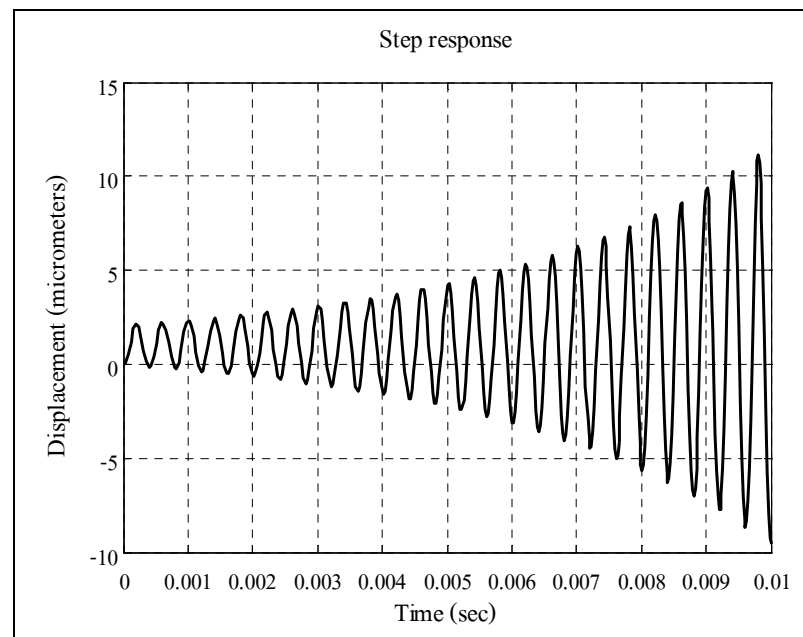
$$G_{r.m.2}(s) = \frac{0.9332s^2 - 805.8s + 1.739 \times 10^9}{s^2 + 125.1s + 1.739 \times 10^9} \quad (2-7)$$

$$G_{r.m.3}(s) = \frac{1.072s^2 + 925.1s + 1.997 \times 10^9}{s^2 + 536.2s + 1.997 \times 10^9} \quad (2-8)$$

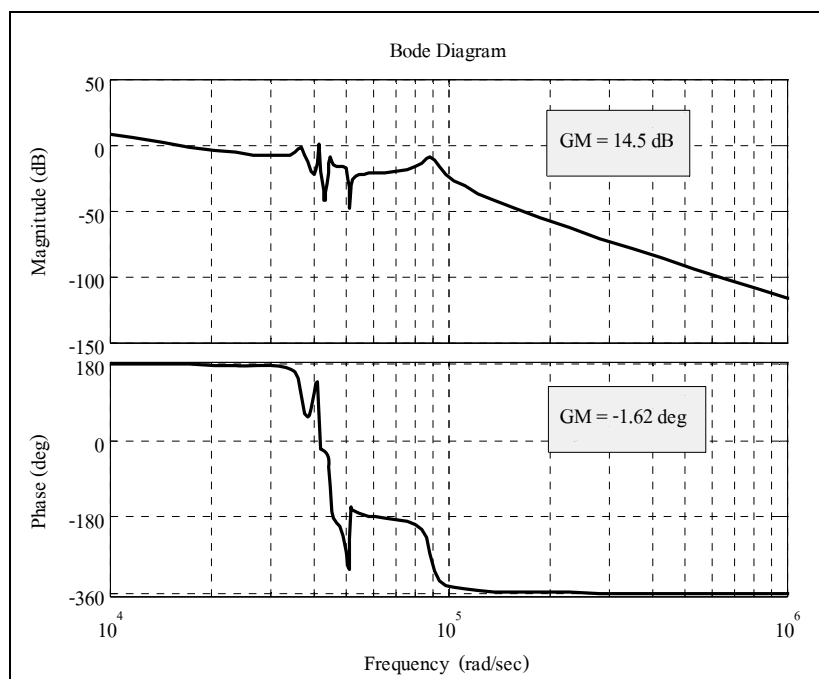
$$G_{r.m.4}(s) = \frac{0.9594s^2 + 98.22s + 2.514 \times 10^9}{s^2 + 1805s + 2.514 \times 10^9} \quad (2-9)$$

$$G_{r.m.5}(s) = \frac{7.877 \times 10^9}{s^2 + 6212s + 7.877 \times 10^9} \quad (2-10)$$

ดำเนินการจำลองสถานการณ์โดยใช้ฟังก์ชันถ่ายโอนดังสมการที่ (2-5) เมื่ออินพุตเป็นขั้นบันไดหนึ่งหน่วย สามารถแสดงผลตอบสนองในโดเมนเวลาได้ด้วยรูปที่ 2.4 ในรูปที่ 2.5 เป็นแผนภาพโพล จะพบว่าห้วงอ่านฮาร์ดดิสก์มี ส่วนเผื่ออัตราขยาย (GM) เท่ากับ 14.5 dB และมีส่วนเฟส (PM) เท่ากับ -1.62° เมื่อพิจารณาที่กราฟในโดเมนเวลาจะสามารถเห็นการขาดเสถียรภาพของระบบได้อย่างชัดเจน



รูปที่ 2.4 ผลตอบสนองทางโดเมนเวลาเมื่ออินพุตเป็นขั้นบันไดหนึ่งหน่วย



รูปที่ 2.5 ผลตอบสนองทางโดเมนความถี่ (bode plot)

2.4 สรุป

ในการหาแบบจำลองทางคณิตศาสตร์ของหัวอ่านฮาร์ดดิสก์ไครฟ์นั้นพิจารณาได้เป็น 2 ขั้นตอนการทำงานก็คือ ในย่านความถี่ต่ำซึ่งแสดงได้ด้วยแบบจำลองทางคณิตศาสตร์แบบไม่เป็นเชิงเส้น และย่านความถี่สูงแสดงได้ด้วยแบบจำลองทางคณิตศาสตร์ที่เป็นฟังก์ชันถ่ายโอนมีอันดับ 12 จากการจำลองสถานการณ์ในย่านความถี่ต่ำพบว่า เมื่อป้อนอินพุตด้วยขนาดที่มากขึ้น ทำให้หัวอ่านเคลื่อนที่ในระยะทางที่เพิ่มขึ้นด้วย แต่การเข้าถึงตำแหน่งของหัวอ่านนั้นมีค่าผิดพลาดเกิดขึ้นมาก ซึ่งเป็นสาเหตุมาจากสภาวะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ ดังนั้นการควบคุมหัวอ่านในย่านความถี่ต่ำจึงมีประเด็นสำคัญเป็นการแก้ปัญหาสภาวะไม่เป็นเชิงเส้นของแรงเสียดทาน ส่วนในย่านความถี่สูง การควบคุมให้หัวอ่านติดตามแทร็กข้อมูล ประสบปัญหาสำคัญจากสภาวะเรโซแนนซ์ และการขาดเสถียรภาพพิจารณาได้จากการจำลองสถานการณ์โดยป้อนอินพุตขั้นบันไดหนึ่งหน่วยซึ่งจากการจำลองสถานการณ์ในโดเมนเวลาจะเห็นการขาดเสถียรภาพได้อย่างชัดเจน และในโดเมนความถี่พบว่ามีสภาวะเรโซแนนซ์เกิดขึ้น ส่วนเพื่ออัตราขยาย (GM) เท่ากับ 14.5 dB และมีส่วนเฟส (PM) เท่ากับ -1.62° ดังนั้นการควบคุมหัวอ่านในย่านความถี่สูงจะมุ่งเน้นการลดทอนผลกระทบจากสภาวะเรโซแนนซ์ และการชดเชยทางเสถียรภาพ การออกแบบระบบควบคุมให้กับหัวอ่านจะดำเนินการทั้งย่านความถี่ต่ำและความถี่สูง ดังจะได้กล่าวถึงรายละเอียดการออกแบบในบทต่อ ๆ ไป

บทที่ 3

การปรับปรุงเสถียรภาพและสมรรถนะของหัวอ่านฮาร์ดดิสก์ไครฟ์ ย่านความถี่ต่ำ

3.1 บทนำ

การวิเคราะห์และออกแบบเพื่อควบคุมระบบด้วยวิธีการแบบดั้งเดิม (conventional control) เช่น การควบคุมแบบอมตะ (classical control) หรือ การควบคุมสมัยใหม่ (modern control) ต้องการแบบจำลองทางคณิตศาสตร์ของระบบที่แม่นยำของกระบวนการที่ควบคุมรวมทั้งต้องมีความรู้ความเข้าใจเกี่ยวกับทฤษฎีการควบคุมเป็นอย่างดี ในกรณีที่ระบบมีความซับซ้อนมาก ๆ การพัฒนาแบบจำลองทางคณิตศาสตร์ต้องใช้เวลานานและยุ่งยาก อาจทำให้ความถูกต้องเที่ยงตรงลดลง เนื่องจากแบบจำลองทางคณิตศาสตร์เป็นหัวใจของการวิเคราะห์และออกแบบระบบด้วยวิธีการแบบอมตะ ดังนั้นระบบที่ขาดแบบจำลองทางคณิตศาสตร์ที่แม่นยำย่อมทำให้วิธีการแบบอมตะมีประสิทธิภาพลดลงไปด้วย การแก้ปัญหาคือการหาข้อจำกัดของวิธีการแบบดั้งเดิมแนวทางหนึ่ง คือ การใช้วิธีการควบคุมแบบชาญฉลาด (intelligent control) (King, 1999) ตัวอย่างเช่น ตัวควบคุมฟัซซีลอจิก (fuzzy logic controller) เป็นต้น ปัจจุบันได้มีการนำวิธีการควบคุมแบบชาญฉลาดมาประยุกต์ใช้อย่างกว้างขวาง เช่น ระบบขนส่งมวลชน (Sujitjorn, Mellitt, and Rabukwella, 1987) ระบบเครื่องปรับอากาศ (Iokibe, Tobi, and Araki, 1999) หุ่นยนต์ (Hagras and Sobh, 2002) ระบบสายส่งกำลังไฟฟ้า (Mohagheghi, Venayagamoorthy and Harley, 2005) ตัวควบคุมฟัซซีลอจิก (fuzzy logic controller) เกิดจากแนวคิดและทฤษฎีฟัซซีเซตของ Lotfi A. Zadeh (Zadeh, 1968) (Zadeh, 1973) (Zadeh, 1976) วิธีการนี้อาศัยข้อมูลความรู้และประสบการณ์การทำงานของมนุษย์เป็นพื้นฐานในการควบคุม ไม่จำเป็นต้องพึ่งพาแบบจำลองทางคณิตศาสตร์ที่แม่นยำของระบบที่ควบคุม (Meireles, Almeida, and Simoes, 2003) ทำให้สามารถประยุกต์วิธีการดังกล่าวเพื่อควบคุมระบบที่มีความซับซ้อนหรือคลุมเครือ และมีความไม่เป็นเชิงเส้นได้อย่างมีประสิทธิภาพ ในการวิเคราะห์และออกแบบเพื่อควบคุมระบบที่มีความซับซ้อนด้วยตัวควบคุมแบบฟัซซีจะแตกต่างจากวิธีการควบคุมแบบดั้งเดิม กล่าวคือ จะใช้ตัวแปรทางภาษา (linguistic variables) แทนการใช้ตัวแปรเชิงเลข (numerical variables) และบ่งบอกลักษณะความสัมพันธ์ระหว่างตัวแปรต่างๆ ด้วยประโยคเงื่อนไขแบบฟัซซี (fuzzy conditional statements) เป็นต้น

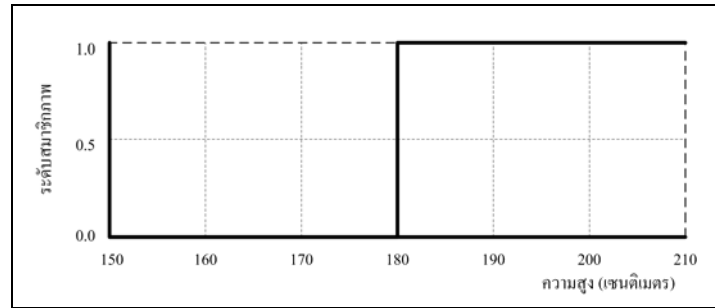
เนื้อหาในบทที่ 3 นี้กล่าวถึง การออกแบบตัวควบคุมแบบกฎสองอินพุต เพื่อลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านฮาร์ดดิสก์ไครฟ์ที่ปรากฏสถานะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ อันดับแรกได้กล่าวถึงหลักการการทำงานของตัวควบคุมฟัซซีแบบกฎสองอินพุต การออกแบบตัวควบคุมดังกล่าวให้กับระบบเพื่อลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านรวมทั้งดำเนินการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ด้วยตัวควบคุมฟัซซี และได้ปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก และเพิ่มจำนวนการแบ่งส่วนตัวแปรฟัซซี เพื่อศึกษาถึงผลกระทบต่อระบบการ

เคลื่อนที่ของหัวอ่าน โดยดำเนินการจำลองสถานการณ์เปรียบเทียบกับผลเดิมที่ได้ออกแบบไว้ก่อนหน้านี้แล้ว นอกจากนี้ได้ดำเนินการออกแบบตัวควบคุมฟัซซีอีกหนึ่งชุดเพื่อช่วยปรับปรุงในสถานะชั่วคราวให้ดีขึ้น และกฎชุดที่สองเพื่อรักษาเสถียรภาพในสถานะอยู่ตัวและนำผลการจำลองสถานการณ์มาเปรียบเทียบกับกรณีที่มีการควบคุมแบบพีไอดีและกรณีที่มีการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) ซึ่งเป็นผลงานวิจัยของ Chen, Lee, Peng, and Venkataramanan, 2006

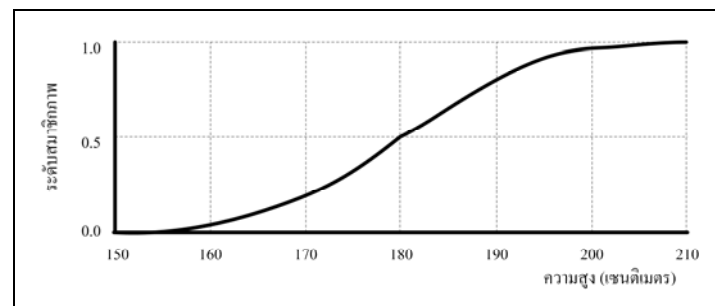
3.2 ความรู้เบื้องต้นเกี่ยวกับฟัซซีลอจิก

ฟัซซีลอจิกถูกพัฒนาขึ้นโดย Lotfi A. Zadeh เมื่อปี ค.ศ. 1965 ซึ่งมีรากฐานอยู่บนทฤษฎีฟัซซีเซต ซึ่งฟัซซีเซตเป็นกลุ่มสิ่งของหรือวัตถุที่มีการเปลี่ยนแปลงความเป็นสมาชิกในเซตอย่างค่อยเป็นค่อยไป หรืออาจกล่าวได้ว่าค่าที่แสดงความเป็นสมาชิกมีลักษณะต่อเนื่อง (สราวุธ สุจิตจร, 2538) เมื่อ 0 หมายถึงการไม่เป็นสมาชิกในเซต และ 1 หมายถึงการเป็นสมาชิกในเซตอย่างสมบูรณ์ โดยจะมีการกำหนดค่าความเป็นสมาชิกผ่านฟังก์ชันความเป็นสมาชิก (membership function) เพื่อระบุค่าความเป็นสมาชิก (degree of membership) ของสมาชิกใด ๆ ในเซตที่อยู่ในขอบเขตเอกภพสัมพัทธ์ (universe of discourse) ที่พิจารณา ในขณะที่เซตปกติธรรมดาจะพิจารณาสิ่งของหรือวัตถุใด ๆ ว่าเป็นสมาชิกหรือไม่เป็นสมาชิกของเซตเท่านั้น ค่าที่ใช้แสดงลักษณะสมบัติดังกล่าวมีเพียง 0 หรือ 1 ซึ่ง 0 หมายถึงการไม่เป็นสมาชิกในเซต และ 1 หมายถึงการเป็นสมาชิกในเซต ดังนั้นจะเห็นได้ว่าฟัซซีเซตมีความแตกต่างไปจากเซตธรรมดาในเรื่องการให้ค่าที่แสดงลักษณะสมบัติความเป็นสมาชิกและไม่เป็นสมาชิกในเซต ซึ่งแสดงในรูปที่ 3.1 โดยเป็นเซตของช่วงความสูงของมนุษย์จาก 150 เซ็นติเมตร เมื่อพิจารณาเซตปกติธรรมดาจะเห็นได้ว่าช่วงความสูง 180 เซ็นติเมตร ถึง 210 เซ็นติเมตร เป็นสมาชิกของเซตอย่างสมบูรณ์ ส่วนที่เหลือนอกย่านดังกล่าวไม่เป็นสมาชิกอย่างสิ้นเชิง เมื่อพิจารณาฟัซซีเซต พบว่า ช่วงความสูงจาก 150 เซ็นติเมตร ถึง 210 เซ็นติเมตร แสดงความเป็นสมาชิกอย่างค่อยเป็นค่อยไป โดยมีระดับความเป็นสมาชิกอยู่ในช่วง 0 ถึง 1

โครงสร้างพื้นฐานของระบบควบคุมฟัซซีแสดงได้ด้วยแผนภาพในรูปที่ 3.2 โดยสัญญาณความคลาดเคลื่อนระหว่างอินพุตกับเอาต์พุตจะเข้าสู่ตัวควบคุมฟัซซี จากนั้นสัญญาณเอาต์พุตจากตัวควบคุมฟัซซีจะถูกป้อนเข้าสู่ระบบที่ต้องการควบคุม จนกระทั่งได้สัญญาณเอาต์พุตของระบบ ซึ่งตัวควบคุมประกอบด้วยชุดปฏิบัติการฟัซซีฟิเคชัน (fuzzification unit) ฐานความรู้ (knowledge) ลอจิกเพื่อการตัดสินใจ (decision making logic) และชุดปฏิบัติการดีฟัซซีฟิเคชัน (defuzzification unit) ดังมีรายละเอียดต่อไปนี้

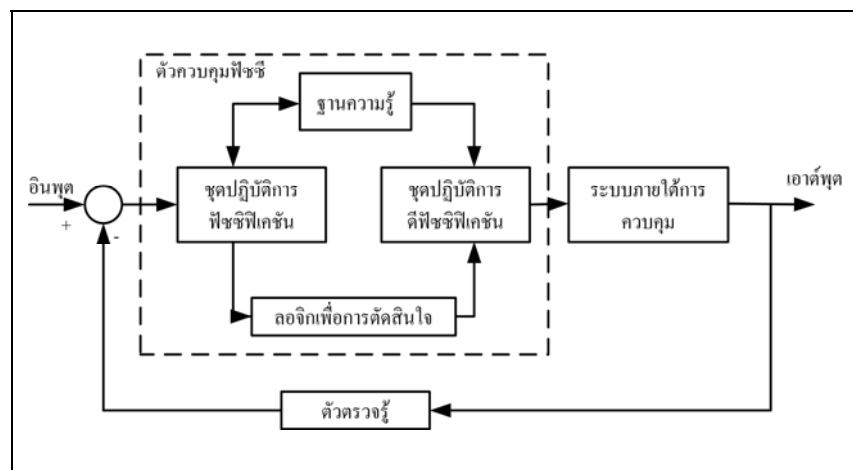


(ก)



(ข)

รูปที่ 3.1 ความแตกต่างของความเป็นสมาชิกระหว่างเซตธรรมดากับฟัซซีเซต
(ก) เซตธรรมดา (ข) ฟัซซีเซต



รูปที่ 3.2 โครงสร้างพื้นฐานของระบบควบคุมแบบฟัซซี

ชุดปฏิบัติการฟัซซีฟิเคชัน ทำหน้าที่แปลงสัญญาณข้อมูลทางด้านอินพุตแบบธรรมดาเป็นตัวแปรภาษาหรือตัวแปรฟัซซี โดยอยู่ในรูปของค่าระดับความเป็นสมาชิกแต่ละฟัซซีเซตบางครั้งอาจรวมชุดวัดสัญญาณด้วยทรานสดิวเซอร์และชุดเปรียบเทียบสัญญาณไว้ในส่วนที่เป็นชุดปฏิบัติการฟัซซีฟิเคชันนี้ด้วย เนื่องจากตัวแปรจากตัวแปรอินพุตแต่ละตัวที่วัดได้จากเครื่องมือวัดอาจอยู่นอกขอบเขตเอกภพสัมพัทธ์ที่กำหนด ดังนั้นจึงต้องดำเนินการแปลงค่าตัวแปรอินพุตเหล่านี้ให้อยู่ในรูปขอบเขตเอกภพสัมพัทธ์ที่กำหนด ด้วยค่าสเกล (scaling) หรือค่าเกน (gain factor) เรียกขั้นตอนนี้ว่า การทำนอร์มัลไลซ์ (normalization) ตัวอย่างเช่น สัญญาณความเร็วจากมอเตอร์กระแสตรงที่วัดได้อยู่ในช่วง $[-1000, +1000]$ รอบต่อนาที ถ้ากำหนดให้เอกภพสัมพัทธ์อยู่ระหว่าง $[-1, +1]$ ดังนั้นต้องทำการคูณค่าแรงดันที่วัดได้ด้วย $(1/1000)$ รอบต่อนาที เป็นต้น

ฐานความรู้ ประกอบด้วยส่วนที่เป็นฐานข้อมูล (data base) และฐานกฎการควบคุม (control rule base) ส่วนที่เป็นฐานข้อมูลนั้นให้นิยามต่าง ๆ ที่ใช้กำหนดกฎการควบคุมนั้น รวมทั้งเป็นศูนย์รวมของกฎการควบคุมแบบตัวแปรทางภาษา ซึ่งแสดงถึงแนวทางการควบคุมให้พฤติกรรมตามที่ต้องการ

ลอจิกเพื่อการตัดสินใจ เป็นส่วนที่ทำให้การเลียนแบบการตัดสินใจของมนุษย์ โดยอาศัยแนวคิดทางคณิตศาสตร์ และการอนุมานแบบฟัซซี

ชุดปฏิบัติการดีฟัซซีฟิเคชัน ทำหน้าที่แปลงข้อมูลแบบฟัซซีไปเป็นตัวแปรธรรมดา (conventional variable) หรือคริปส์เอาต์พุตที่สมนัยกัน เพื่อให้ระบบภายใต้การควบคุมสามารถเข้าใจสัญญาณดังกล่าวได้ ข้อมูลแบบฟัซซีนี้เป็นข้อมูลที่ได้มาจากการอนุมานกฎการควบคุมแบบฟัซซี เนื่องจากขอบเขตเอกภพสัมพัทธ์ของเอาต์พุตในฟังก์ชันความเป็นสมาชิกอาจยังไม่อยู่ในขอบเขตที่น่าไปใช้งานได้ ดังนั้นจึงต้องทำการแปลงค่าเอาต์พุตในเอกภพสัมพัทธ์ให้อยู่ในขอบเขตเอาต์พุตจริงที่ใช้ควบคุมระบบ เรียกขั้นตอนนี้ว่าการดีนอร์มัลไลซ์ (demoralization) ตัวอย่างเช่น เอกภพสัมพัทธ์ของเอาต์พุตในฟังก์ชันความเป็นสมาชิกอยู่ในช่วง $[-1, +1]$ แต่สัญญาณแรงดันที่ป้อนให้กับมอเตอร์กระแสตรงมีขอบเขตอยู่ในช่วง $[-12, +12]$ โวลต์ ดังนั้นต้องทำการคูณค่าเอาต์พุตเอกภพสัมพัทธ์ด้วย 12 ก่อนส่งไปควบคุมระบบจริง เป็นต้น

การควบคุมระบบแบบฟัซซี เป็นการวิเคราะห์และแก้ไขปัญหาเกี่ยวกับระบบที่มีความซับซ้อน โดยอาศัยคณิตศาสตร์แบบฟัซซี อาศัยกระบวนการตัดสินใจและความคิดอ่านของมนุษย์เป็นพื้นฐานในการควบคุม โดยแสดงความสัมพันธ์ระหว่างสถานะต่าง ๆ ในระบบพลวัตในรูปประโยคฟัซซี ซึ่งมีลักษณะเป็นประโยคเงื่อนไข ถ้า... แล้ว... (if... then) เรียกประโยคเงื่อนไขแบบนี้ว่า กฎฟัซซี (fuzzy rule) ในการดำเนินงานตัวแปรทางภาษาแทนการใช้ตัวแปรเชิงเลข ซึ่งตัวแปรทางภาษาเป็นตัวแปรที่มีการให้ค่าเป็นคำหรือประโยค เพื่อบ่งบอกลักษณะของสิ่งของใด ๆ ซึ่งมีค่าตามค่าให้ค่าของฟัซซีเซต ตัวอย่างเช่น ถ้าความเร็วของมอเตอร์มีค่ามาก แล้วลดแรงดันที่ให้กับมอเตอร์ ซึ่งตัวแปรทางภาษาหรือตัวแปรฟัซซี คือความเร็วของมอเตอร์ และแรงดันที่ให้กับมอเตอร์ เป็นต้น

กฎฟัซซีในประโยคเงื่อนไข ถ้า... แล้ว... ประกอบด้วยส่วนของตัวแปรเงื่อนไข (antecedent) และส่วนของตัวแปรผลลัพธ์ (consequent) ซึ่งในแต่ละส่วนอาจมีได้หลายตัวแปร โดยเชื่อมประโยคด้วยโอเพอร์เรเตอร์ and หรือ or เพื่อใช้ในการแก้ปัญหาสำหรับควบคุมระบบอินพุตเดียวเอาต์พุตเดียว (SISO)

และระบบหลายอินพุต (MIMO) กฎการควบคุมฟัซซีสามารถกำหนดขึ้นด้วยการเขียนเป็นประโยคแสดงอาการของความพยายามควบคุมกับผลที่เกิดขึ้นกับการตอบสนองของระบบ โดยอาศัยความรู้ความชำนาญและประสบการณ์ของมนุษย์ในการกำหนดกฎฟัซซีเพื่อควบคุมระบบ

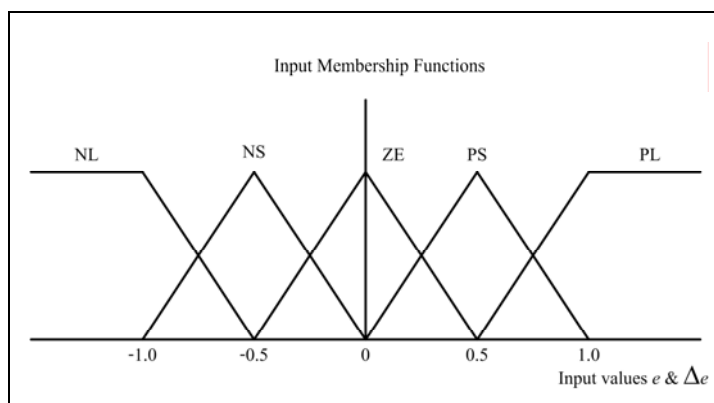
3.3 การออกแบบตัวควบคุมฟัซซีแบบกฎสองอินพุต

การออกแบบตัวควบคุมฟัซซีแบบกฎสองอินพุตเพื่อลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านฮาร์ดดิสก์ไดรฟ์ที่ปรากฏสถานะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ ในงานวิจัยวิทยานิพนธ์นี้ พิจารณาการควบคุมตำแหน่งของหัวอ่านเมื่อเคลื่อนตัวไปตามแทรก (track) ต่าง ๆ เพื่ออ่านข้อมูล ให้หัวอ่านเคลื่อนที่ไปด้วยระยะทาง $1\ \mu\text{m}$ ทั้งนี้จะพิจารณาเวลาในการเข้าสู่สถานะอยู่ตัวและการพุ่งเกินเป็นปัจจัยบ่งชี้สมรรถนะของระบบ ตัวควบคุมแบบฟัซซีที่ใช้ในงานวิจัยนี้เป็นแบบระบบเดี่ยว กฎการควบคุมมี 2 อินพุต และ 1 เอาต์พุต ในส่วนของตัวแปรอินพุตคือ ความคลาดเคลื่อนของตำแหน่งของหัวอ่านฮาร์ดดิสก์ (error, e) และ ค่าปริมาณความคลาดเคลื่อนที่เปลี่ยนแปลงของหัวอ่านฮาร์ดดิสก์ (change in error, Δe) ส่วนตัวแปรเอาต์พุต คือ ปริมาณที่เปลี่ยนแปลงในสัญญาณการควบคุม (Δu)

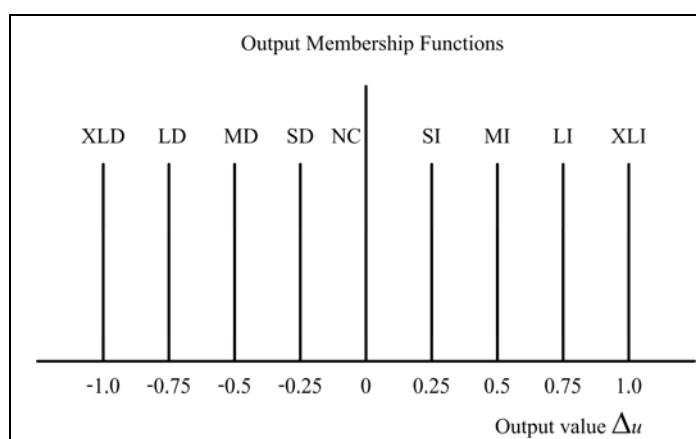
จากการวิเคราะห์พฤติกรรมของระบบโดยอาศัยผลการจำลองสถานการณ์ ดังปรากฏในบทที่ 2 สามารถกำหนดกฎฟัซซีได้ดังตารางที่ 3.1 และฟังก์ชันความเป็นสมาชิกในส่วนของตัวแปรอินพุต ซึ่งกำหนดให้เป็นรูปสามเหลี่ยม และรูปสี่เหลี่ยมคางหมูแสดงไว้ดังรูปที่ 3.3 ประกอบไปด้วย 5 ฟัซซีเซตคือ NL (negative large), NS (negative small), ZE (zero), PS (positive small) และ PL (positive large) โดยฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตในส่วนผลลัพธ์ของกฎฟัซซีแสดงไว้ดังรูปที่ 3.3 กำหนดเป็นแบบ Sugeno ประกอบไปด้วย 9 ฟัซซีเซตคือ XLD (extra largely decrease) LD (largely decrease) MD (moderately decrease) SD (slightly decrease) NC (no change) SI (slightly increase) MI (moderately increase) LI (largely increase) และ XLI (extra largely increase) ทั้งนี้การอนุมานกฎฟัซซีใช้วิธี Sugeno (T. Takagi and M. Sugeno, 1985)

ตารางที่ 3.1 กฎฟัซซี

		Change in errors (Δe)				
		NL	NS	ZE	PS	PL
errors (e)	NL	XLI	LI	MI	SI	NC
	NS	LI	MI	SI	NC	SD
	ZE	MI	SI	NC	SD	MD
	PS	SI	NC	SD	MD	LD
	PL	NC	SD	MD	LD	XLD



(ก)



(ข)

รูปที่ 3.3 (ก) ฟังก์ชันความเป็นสมาชิกของอินพุต

(ข) ฟังก์ชันความเป็นสมาชิกของเอาต์พุต

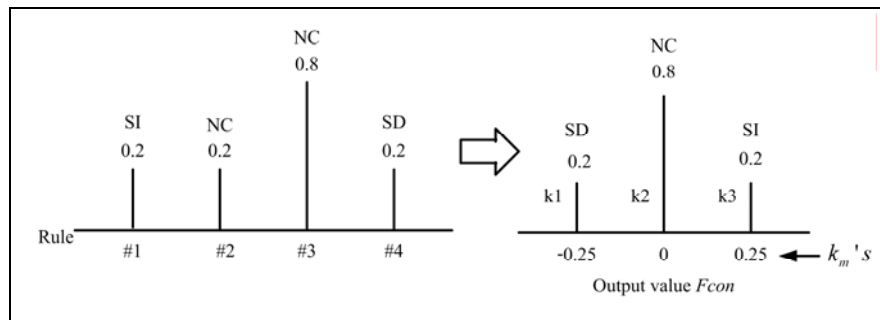
จากการกำหนดกฎฟัซซีตามตารางที่ 3.1 และการกำหนดฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตและตัวแปรเอาต์พุตดังแสดงในรูปที่ 3.3 เมื่อนำมาเขียนกฎฟัซซีจะได้ทั้งสิ้นจำนวน 25 กฎดังต่อไปนี้

กฎที่ 1	ถ้า	$e = \text{NL}$	และ	$\Delta e = \text{NL}$	แล้ว	$\Delta u = \text{XLI}$
กฎที่ 2	ถ้า	$e = \text{NS}$	และ	$\Delta e = \text{NL}$	แล้ว	$\Delta u = \text{LI}$
กฎที่ 3	ถ้า	$e = \text{ZE}$	และ	$\Delta e = \text{NL}$	แล้ว	$\Delta u = \text{SI}$
กฎที่ 4	ถ้า	$e = \text{PS}$	และ	$\Delta e = \text{NL}$	แล้ว	$\Delta u = \text{SI}$

กฎที่ 5	ถ้า	$e = PL$	และ	$\Delta e = NL$	แล้ว	$\Delta u = NC$
กฎที่ 6	ถ้า	$e = NL$	และ	$\Delta e = NS$	แล้ว	$\Delta u = XLI$
กฎที่ 7	ถ้า	$e = NS$	และ	$\Delta e = NS$	แล้ว	$\Delta u = MI$
กฎที่ 8	ถ้า	$e = ZE$	และ	$\Delta e = NS$	แล้ว	$\Delta u = SI$
กฎที่ 9	ถ้า	$e = PS$	และ	$\Delta e = NS$	แล้ว	$\Delta u = NC$
กฎที่ 10	ถ้า	$e = PL$	และ	$\Delta e = NS$	แล้ว	$\Delta u = SP$
กฎที่ 11	ถ้า	$e = NL$	และ	$\Delta e = ZE$	แล้ว	$\Delta u = MI$
กฎที่ 12	ถ้า	$e = NS$	และ	$\Delta e = ZE$	แล้ว	$\Delta u = SI$
กฎที่ 13	ถ้า	$e = ZE$	และ	$\Delta e = ZE$	แล้ว	$\Delta u = NC$
กฎที่ 14	ถ้า	$e = PS$	และ	$\Delta e = ZE$	แล้ว	$\Delta u = SD$
กฎที่ 15	ถ้า	$e = PL$	และ	$\Delta e = ZE$	แล้ว	$\Delta u = MD$
กฎที่ 16	ถ้า	$e = NL$	และ	$\Delta e = PS$	แล้ว	$\Delta u = SI$
กฎที่ 17	ถ้า	$e = NS$	และ	$\Delta e = PS$	แล้ว	$\Delta u = NC$
กฎที่ 18	ถ้า	$e = ZE$	และ	$\Delta e = PS$	แล้ว	$\Delta u = SD$
กฎที่ 19	ถ้า	$e = PS$	และ	$\Delta e = PS$	แล้ว	$\Delta u = MD$
กฎที่ 20	ถ้า	$e = PL$	และ	$\Delta e = PS$	แล้ว	$\Delta u = XLD$
กฎที่ 21	ถ้า	$e = NL$	และ	$\Delta e = PL$	แล้ว	$\Delta u = NC$
กฎที่ 22	ถ้า	$e = NS$	และ	$\Delta e = PL$	แล้ว	$\Delta u = SD$
กฎที่ 23	ถ้า	$e = ZE$	และ	$\Delta e = PL$	แล้ว	$\Delta u = SD$
กฎที่ 24	ถ้า	$e = PS$	และ	$\Delta e = PL$	แล้ว	$\Delta u = LD$
กฎที่ 25	ถ้า	$e = PL$	และ	$\Delta e = PL$	แล้ว	$\Delta u = XLD$

3.4 การอนุมานกฎฟัซซี

รูปแบบการอนุมานกฎฟัซซีดำเนินการตามวิธี Sugeno ซึ่ง Michio Sugeno, 1985 ได้เสนอวิธีการอนุมานที่แตกต่างไปจากวิธีของแมมดานี (Mamdani and Assilian, 1975) ด้วยการลดขั้นตอนการคำนวณหาจุดศูนย์กลางในการทำดีฟัซซี Sugeno ได้ใช้เส้นตรงโทนในการแทนฟังก์ชันสมาชิกในส่วนของตัวเองแปรเอาต์พุต ในที่นี้ค่าเอาต์พุตของแต่ละกฎจะเป็นเส้นตรงโทนซึ่งทำให้สะดวกสำหรับขั้นตอนการรวมกฎ จะเห็นได้ว่าความแตกต่างของระบบฟัซซีแบบ Sugeno มีเพียงส่วนเอาต์พุตเท่านั้น การทำดีฟัซซีของระบบแสดงได้ดังรูปที่ 3.4



รูปที่ 3.4 การรวมกฎแบบ Sugeno

สามารถคำนวณค่าเอาต์พุตที่ได้จากการทำฟัซซี่ ได้ดังสมการที่ (3-1) ค่าเอาต์พุตที่ได้จากการทำฟัซซี่นี้เรียกว่าค่าถ่วงน้ำหนักเฉลี่ย (weighted average หรือ WA)

$$WA = \frac{\sum_{m=1}^L u(k_m) \times k_m}{\sum_{m=1}^L u(k_m)} \quad (3-1)$$

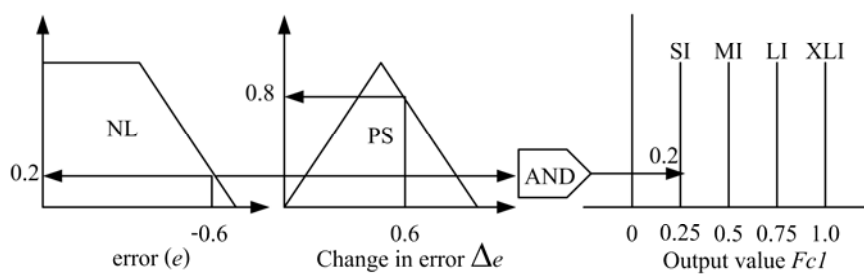
โดยที่ k_m คือ เส้นตรงโทน

$u(k_m)$ คือ ค่าระดับความเป็นสมาชิกของเส้นตรงโทน k_m

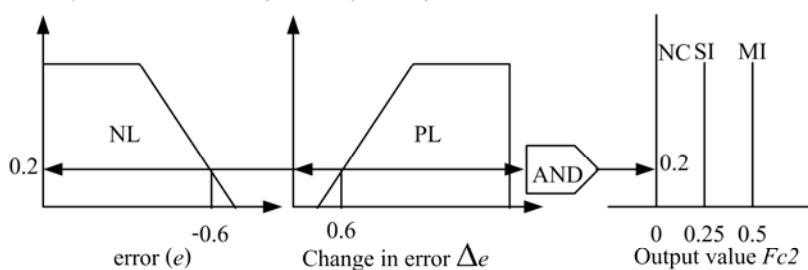
ในการอนุมานกฎฟัซซี่ แต่ละครั้ง มีจำนวนกฎที่เป็นไปได้จาก 25 กฎ อยู่เพียง 1-4 กฎ ดังนั้นการอนุมานกฎฟัซซี่ดำเนินการดังตัวอย่างต่อไปนี้

กำหนดให้ error (e) = -0.6 change in error (Δe) = 0.6 พิจารณาความคลาดเคลื่อนของตำแหน่งของหัวอ่านฮาร์ดดิสก์ (error, e) ที่ -0.6 ซึ่งเป็นค่าที่อยู่ในเซต NL และเซต NS โดยมีระดับความเป็นสมาชิกของเซต NL เท่ากับ 0.2 และมีระดับความเป็นสมาชิกของเซต NS เท่ากับ 0.8 พิจารณาค่าปริมาณความคลาดเคลื่อนที่เปลี่ยนแปลงของหัวอ่านฮาร์ดดิสก์ (change in error, Δe) ที่ 0.6 ซึ่งเป็นค่าที่อยู่ในเซต PS และเซต PL โดยมีระดับความเป็นสมาชิก เท่ากับ 0.8 และ 0.2 ตามลำดับ ดังนั้นจะได้กฎในส่วนเงื่อนไข 4 ข้อดังแสดงรายละเอียดตามรูปที่ 3.5

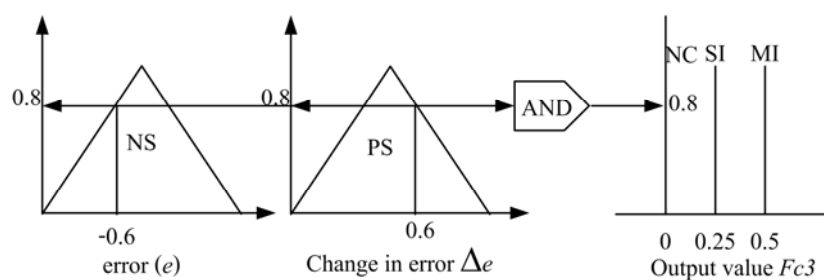
1. IF (e is NL AND e is PS) THEN (u is SI)



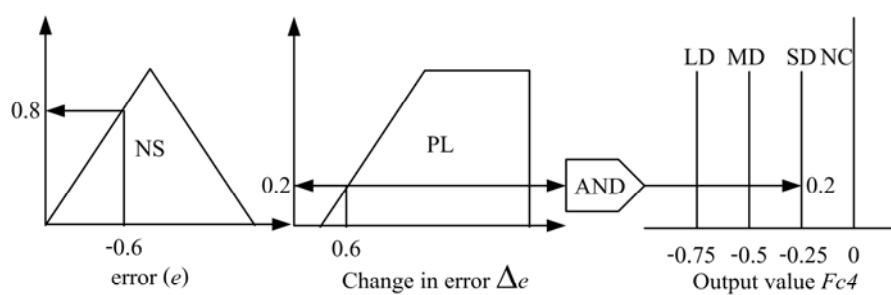
2. IF (e is NL AND e is PL) THEN (u is NC)



3. IF (e is NS AND e is PS) THEN (u is NC)



4. IF (e is NS AND e is PL) THEN (u is SD)



รูปที่ 3.5 การอนุมานกฎฟัซซีแบบ Sugeno

จากรูป 3.4 เมื่อรวมผลจาก 4 กฎ ทั้ง 4 เข้าด้วยกันแบบ Sugeno และคำนวณผลการอนุมานกฎฟัซซีแบบ Sugeno ตามสมการ ที่ (3-1) จะได้ดังสมการที่ (3-2)

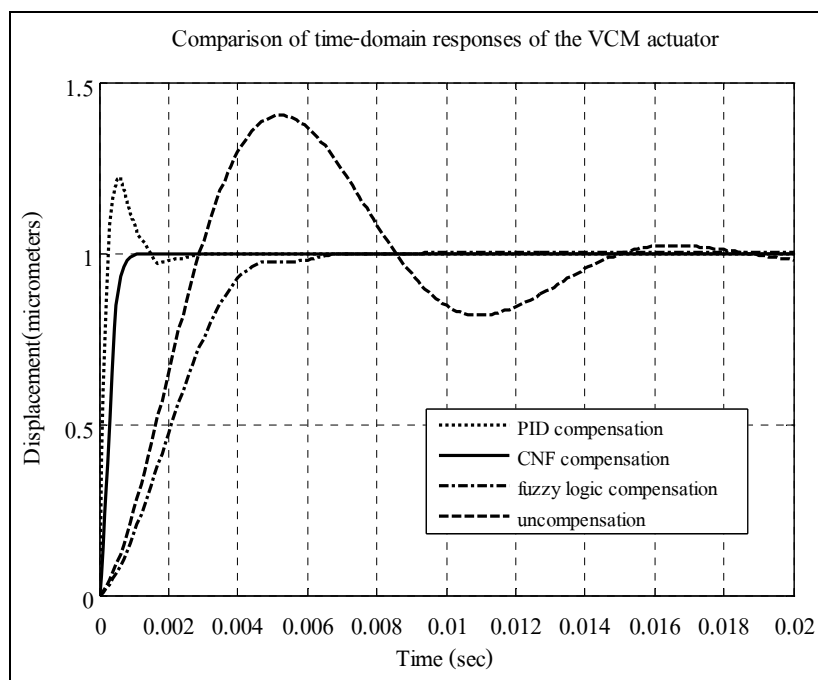
$$WA = \frac{0.2 \times (-0.25) + 0.8 \times 0 + 0.2 \times 0.25}{0.2 + 0.8 + 0.2} = 0 \quad (3-2)$$

จะเห็นได้ว่าการอนุมานกฎฟัซซีแบบ Sugeno เป็นวิธีที่ง่าย ไม่ยุ่งยากซับซ้อน

3.5 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ด้วยตัวควบคุมฟัซซี

เพื่อทดสอบประสิทธิภาพของตัวควบคุมฟัซซีแบบกฎสองอินพุต ได้ดำเนินการจำลองสถานการณ์โดยใช้แบบจำลองคณิตศาสตร์ของระบบที่ได้แสดงไว้ในสมการที่ (2-1) ถึง (2-4) ซึ่งค่าพารามิเตอร์ต่าง ๆ ของระบบการเคลื่อนที่ของหัวอ่านที่ปรากฏในสมการดังกล่าว กำหนดให้ แรงดันอินพุตที่เวลาเริ่มต้น (u_0) มีค่าเป็น 0 โวลต์ ระยะการเคลื่อนที่ของหัวอ่านที่เวลาเริ่มต้น (y_0) เป็น 0 เมตร แรงดันอินพุต (u) เป็น 10 มิลลิโวลต์ และ เอาต์พุตซึ่งเป็นระยะการเคลื่อนที่ของหัวอ่าน (y) มีค่าเป็น 0 เมตร ผลการจำลองสถานการณ์แสดงในรูปที่ 3.6 แสดงกรณีที่มีการควบคุมด้วยฟัซซี กรณีที่มีการควบคุมแบบพีไอดี และกรณีที่มีการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) ซึ่งเป็นผลงานวิจัยของ Chen และคณะ (Chen, Lee, Peng, and Venkataramanan, 2006) และ กรณี ที่ ไม่ มีการ ควบคุม ใด ๆ เพื่อเปรียบเทียบสมรรถนะของตัวควบคุม

เมื่อพิจารณาผลการตอบสนองจากการจำลองสถานการณ์กรณีที่มีการควบคุมด้วยฟัซซีและไม่มี การควบคุมใด ๆ ซึ่งแสดงในรูปที่ 3.6 จะเห็นว่าเวลาในการลู่เข้าสู่สถานะอยู่ตัวกรณีที่มีการควบคุมด้วยฟัซซีนั้นเท่ากับ 7 มิลลิวินาทีและไม่มีการพุ่งเกินเกิดขึ้นเมื่อเปรียบเทียบกับกรณีที่ไม่มีการควบคุมซึ่งมีการพุ่งเกินถึง 43 % และใช้เวลาในการเข้าสู่สถานะอยู่ตัวถึง 20 มิลลิวินาที จะเห็นว่าการควบคุมแบบฟัซซีนั้นได้ลดทอนการพุ่งเกินลงไปและยังช่วยลดเวลาในการเข้าสู่สถานะอยู่ตัวได้ดีขึ้น แต่เมื่อเปรียบเทียบกับผลงานวิจัยที่มีมาก่อนของ Chen และคณะ ที่ใช้วิธีการควบคุมแบบพีไอดี และการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) นั้นกรณีที่มีการควบคุมแบบพีไอดีใช้เวลาในการเข้าสู่สถานะอยู่ตัวเพียง 1.7 มิลลิวินาที แต่มีการพุ่งเกินเกิดขึ้นมากกว่า 20 % ส่วนกรณีที่มีการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) นั้นใช้เวลาในการสู่สถานะอยู่ตัวที่รวดเร็วเพียง 0.8 มิลลิวินาที และยังไม่มีการพุ่งเกินเกิดขึ้น

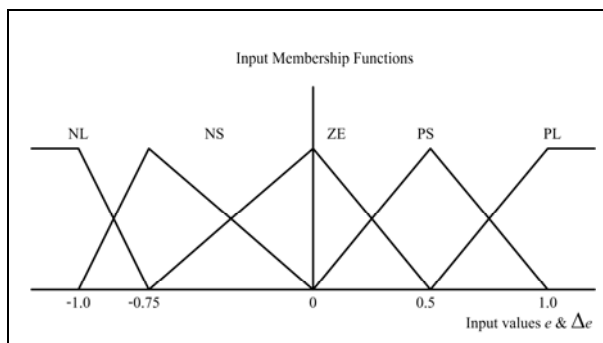


รูปที่ 3.6 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านกรณีมีการควบคุม
ด้วยวิธีต่าง ๆ และกรณีไม่มีการควบคุม

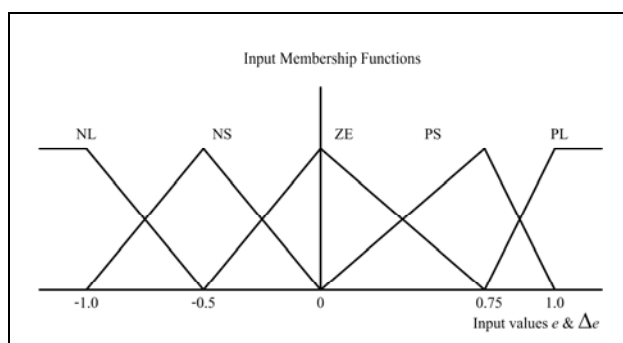
3.6 การปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิก

เพื่อปรับปรุงเวลาในการเข้าสู่เสถียรภาพของหัวอ่านฮาร์ดดิสก์ให้ดีขึ้นนั้น ได้ดำเนินการปรับรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตของตัวควบคุมฟัซซี่แบบกฎสองอินพุต ให้มีฟังก์ชันความเป็นสมาชิกเป็นรูปสามเหลี่ยม และสี่เหลี่ยมคางหมูแบบไม่สมมาตร ทั้งนี้ในส่วนของฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตยังคงรูปแบบเป็นแบบ Sugeno เช่นเดิม รวมถึงการตั้งกฎการควบคุม กระบวนการฟัซซี่ฟิเคชันและดีฟัซซี่ฟิเคชัน นอกจากนี้ค่าพารามิเตอร์ต่าง ๆ ของตัวควบคุมแบบกฎสองอินพุตยังคงเดิม เพียงแต่ปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตในกฎฟัซซี่เท่านั้น

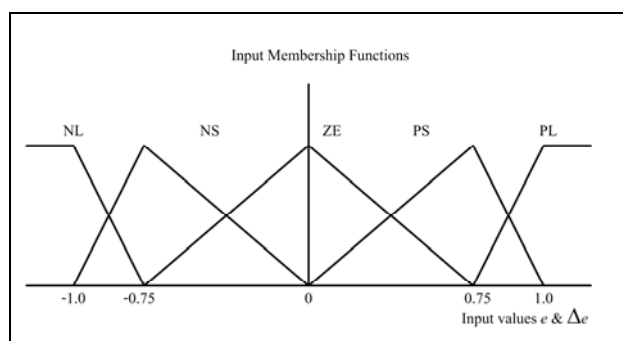
ดำเนินการปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตเป็นแบบรูปสามเหลี่ยม และสี่เหลี่ยมคางหมูแบบไม่สมมาตรทั้งหมด 3 กรณี ดังแสดงในรูปที่ 3.7 ซึ่งแสดงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตแต่ละตัว คือ ความคลาดเคลื่อนของตำแหน่งของหัวอ่านฮาร์ดดิสก์ (error, e) และ ค่าปริมาณความคลาดเคลื่อนที่เปลี่ยนแปลงของหัวอ่านฮาร์ดดิสก์ (change in error, Δe) โดยการแบ่งส่วนตัวแปรฟัซซี่ประกอบด้วย 5 ฟัซซี่เซตคือ NL (negative large) NS (negative small) ZE (zero) PS (positive small) และ PL (positive large) ซึ่งเป็นตัวแปรภาษา



(ก)



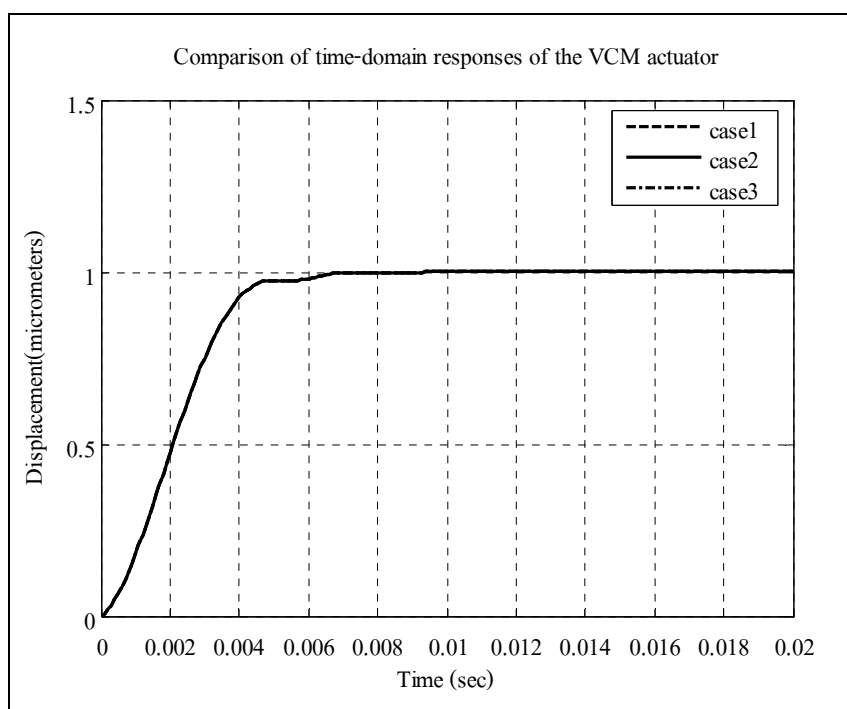
(ข)



(ค)

- รูปที่ 3.7 (ก) ฟังก์ชันความเป็นสมาชิกของอินพุตกรณีที่ 1
 (ข) ฟังก์ชันความเป็นสมาชิกของอินพุตกรณีที่ 2
 (ค) ฟังก์ชันความเป็นสมาชิกของอินพุตกรณีที่ 3

ดำเนินการจำลองสถานการณ์หลังจากที่ปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตทั้งสามกรณีที่ได้แสดงในรูปที่ 3.7 (ก) (ข) และ (ค) และแสดงผลการจำลองสถานการณ์ไว้ในรูปที่ 3.8



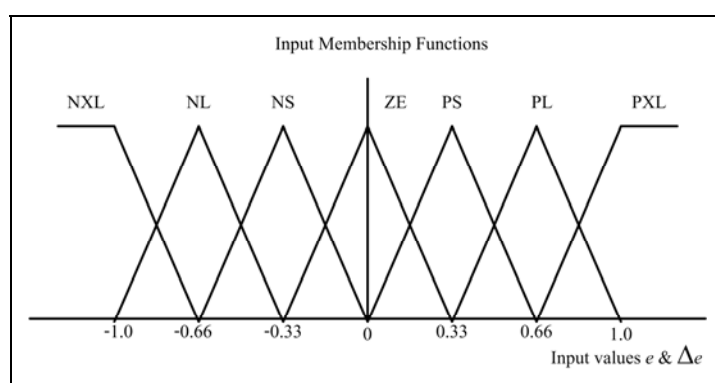
รูปที่ 3.8 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน

เมื่อพิจารณาผลการจำลองสถานการณ์ที่ได้แสดงไว้ในรูปที่ 3.8 หลังจากปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกในส่วนตัวแปรอินพุตเป็นสามกรณีคือ แบบ สามเหลี่ยม และสี่เหลี่ยมคางหมูแบบไม่สมมาตร ซึ่งแสดงในรูปที่ 3.7 จะเห็นได้ว่า การเคลื่อนที่ของหัวอ่านเมื่อปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกทั้งสามกรณีนั้น หัวอ่านเคลื่อนที่ไปในตำแหน่งเดียวกันดูได้จากเส้นกราฟที่แสดงในรูปที่ 3.8 ซึ่งทับกันพอดี และใช้เวลาในการเข้าสู่สถานะอยู่ตัวเท่ากันคือ 7 มิลลิวินาทีและไม่มีการพุ่งเกินเกิดขึ้น ซึ่งเมื่อเปรียบเทียบกับกรณีที่ยังไม่ได้ปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุต ซึ่งเดิมเป็นกรณีรูปทรงฟังก์ชันความเป็นสมาชิกเป็นแบบ สามเหลี่ยม และสี่เหลี่ยมคางหมูแบบสมมาตรนั้นมีค่าเท่ากันคือ 7 มิลลิวินาที ซึ่งอาจกล่าวได้ว่าการปรับเปลี่ยนรูปทรงฟังก์ชันความเป็นสมาชิกของตัวควบคุมฟัซซีที่ใช้ควบคุมการเคลื่อนที่ของหัวอ่านนั้น ไม่มีผลใด ๆ กับการตอบสนองของหัวอ่านฮาร์ดดิสก์ไครฟ์เลย

3.7 การเพิ่มการแบ่งส่วนตัวแปรฟัซซี

เพื่อปรับปรุงเวลาในการเข้าสู่เสถียรภาพของหัวอ่านฮาร์ดดิสก์ให้ดีขึ้นนั้น ได้ดำเนินการเพิ่มการแบ่งส่วนตัวแปรฟัซซีโดยประกอบด้วย 7 ฟัซซีเซต ซึ่งเดิมได้ออกแบบไว้ 5 ฟัซซีเซต ทั้งนี้ได้ดำเนินการทดสอบโดยกำหนดให้รูปแบบฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตเป็นรูปสามเหลี่ยมและสี่เหลี่ยมคางหมูเช่นเดิม โดยกำหนดพารามิเตอร์ต่าง ๆ ของตัวควบคุมฟัซซีแบบกฎสองอินพุตมีค่าเท่าเดิม

ฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุตประกอบด้วยตัวแปรทางภาษา NXL (negative extra largely) NL (negative large) NS (negative small) ZE (zero) PS (positive small) PL (positive large) และ PXL (positive extra largely) ดังแสดงในรูปที่ 3.9 ทั้งนี้ในส่วนของฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตยังคงรูปแบบเป็นแบบ Sugeno เช่นเดิม คือประกอบไปด้วย 9 ฟัซซีเซตคือ XLD (extra largely decrease) LD (largely decrease) MD (moderately decrease) SD (slightly decrease) NC (no change) SI (slightly increase) MI (moderately increase) LI (largely increase) และ XLI (extra largely increase) การตั้งกฎการควบคุม กระบวนการฟัซซีฟิเคชันและดีฟัซซีฟิเคชันเป็นแบบ Sugeno ซึ่งค่าพารามิเตอร์ต่าง ๆ ของระบบการเคลื่อนที่ของหัวอ่านยังคงเหมือนเดิม นอกจากนี้ค่าพารามิเตอร์ต่าง ๆ ของตัวควบคุมแบบกฎสองอินพุตยังคงเดิม สามารถกำหนดกฎฟัซซีทั้งสิ้นจำนวน 49 กฎ ดังตารางที่ 3.2

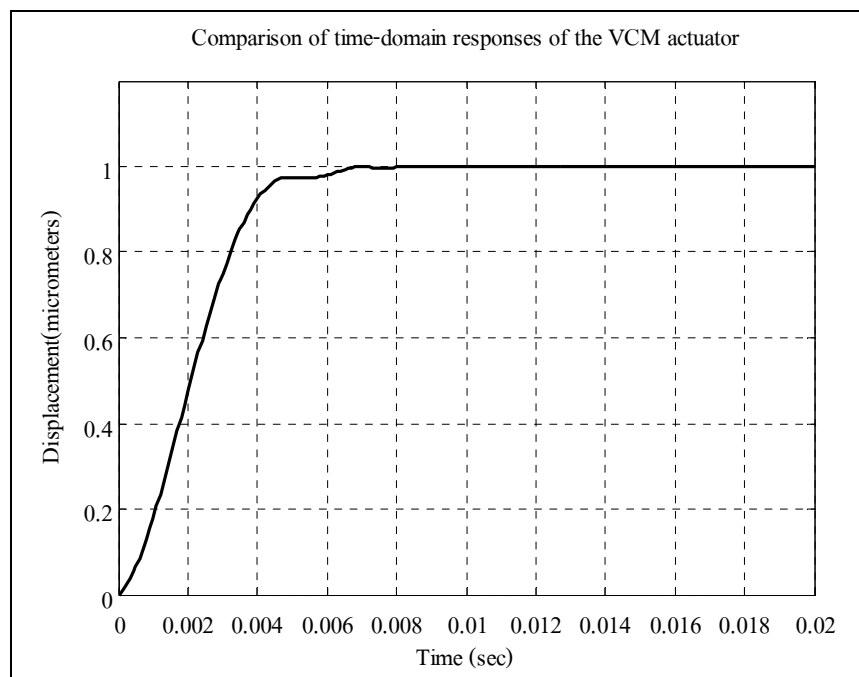


รูปที่ 3.9 ฟังก์ชันความเป็นสมาชิกของอินพุต

ตารางที่ 3.2 แสดงกฎฟัซซี่

		Change in error (Δe)						
		NXL	NL	NS	ZE	PS	PL	XPL
error (e)	NXL	XLI	XLI	XLI	LI	MI	SI	NC
	NL	XLI	XLI	LI	MI	SI	NC	SD
	NS	XLI	LI	MI	SI	NC	SD	MD
	ZE	LI	MI	SI	NC	SD	MD	LD
	PS	MI	SI	NC	SD	MD	LD	XLD
	PL	SI	NC	SD	MD	LD	XLD	XLD
	XPL	NC	SD	MD	LD	XLD	XLD	XLD

เพื่อทดสอบประสิทธิภาพของตัวควบคุมฟัซซี่แบบกฎสองอินพุต ได้ดำเนินการจำลองสถานการณ์หลังจากที่ได้เพิ่มการแบ่งส่วนตัวแปรฟัซซี่โดยประกอบด้วย 7 ฟัซซี่เซต ซึ่งเดิมได้ออกแบบไว้ 5 ฟัซซี่เซต ผลการจำลองสถานการณ์แสดงในรูปที่ 3.10



รูปที่ 3.10 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่าน

เมื่อพิจารณาผลการจำลองสถานการณ์ที่แสดงไว้ในรูปที่ 3.10 จะเห็นได้ว่า การเคลื่อนที่ของหัวอ่านเมื่อเพิ่มการแบ่งส่วนตัวแปรฟัซซีโดยประกอบด้วย 7 ฟัซซีเซต ซึ่งเดิมได้ออกแบบไว้ 5 ฟัซซีเซต ใช้เวลาในการเข้าสู่สถานะอยู่ตัวเท่ากับ 7 มิลลิวินาทีและไม่มีการพุ่งเกินเกิดขึ้น ซึ่งมีค่าเหมือนกับการจำลองสถานการณ์ที่แสดงในรูปที่ 3.6 ด้วยโดยเป็นกรณีที่มีการแบ่งส่วนตัวแปรฟัซซีเป็น 5 ฟัซซีเซต จะมีจำนวนกฎฟัซซีทั้งสิ้น 25 กฎ ในขณะที่เมื่อทำการเพิ่มการแบ่งส่วนตัวแปรฟัซซีเป็น 7 ฟัซซีเซต พบว่าจำนวนกฎเพิ่มขึ้นเป็น 49 กฎ ดังนั้นอาจกล่าวได้ว่าการเพิ่มการแบ่งส่วนตัวแปรฟัซซีจะส่งผลให้จำนวนกฎฟัซซีเพิ่มขึ้นโดยการเพิ่มการแบ่งส่วนตัวแปรฟัซซีที่ใช้ควบคุมการเคลื่อนที่ของหัวอ่านนั้น ไม่มีผลใด ๆ กับระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์เลย

3.8 การออกแบบตัวควบคุมฟัซซีแบบกฎสองชุด

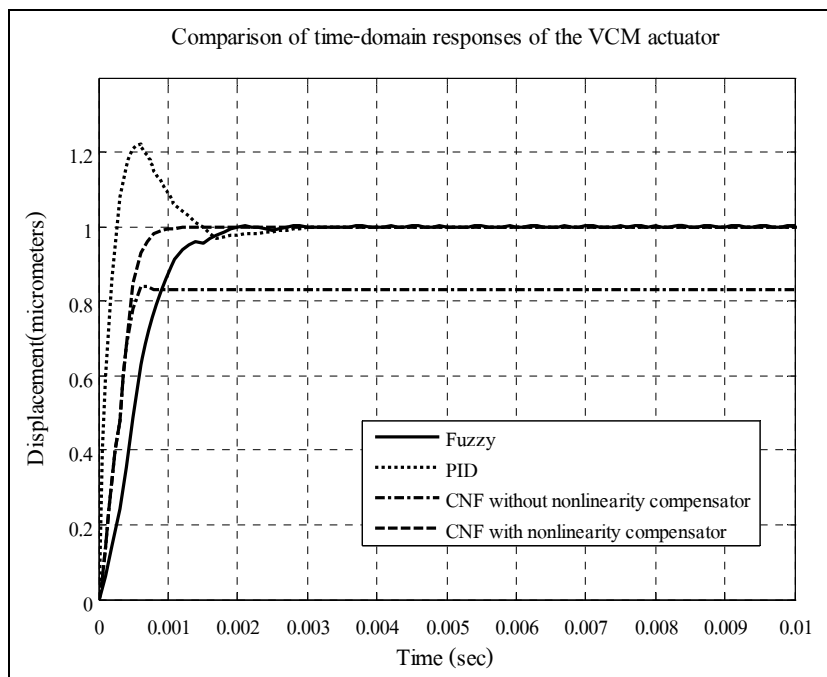
การควบคุมการเคลื่อนที่ของหัวอ่านเพื่อปรับปรุงเวลาในการเข้าสู่สถานะอยู่ตัวของหัวอ่านฮาร์ดดิสก์ให้ดีขึ้นนั้น จะเห็นว่ากฎฟัซซีที่ได้ออกแบบไว้ข้างต้นนั้นสามารถทำให้หัวอ่านสามารถเข้าสู่เสถียรภาพได้แต่ใช้เวลาในการเข้าสู่สถานะอยู่ตัวนั้นค่อนข้างมาก ซึ่งเมื่อมีการปรับเปลี่ยนฟังก์ชันความเป็นสมาชิกของตัวแปรอินพุต เป็นรูปสามเหลี่ยม และสี่เหลี่ยมคางหมูแบบไม่สมมาตรแล้ว พบว่าไม่สามารถช่วยลดเวลาในการเข้าสู่สถานะอยู่ตัวลงได้ รวมทั้งได้ดำเนินการเพิ่มจำนวนการแบ่งส่วนตัวแปรฟัซซีแล้ว ซึ่งจากผลการจำลองสถานการณ์แสดงให้เห็นว่า การเพิ่มจำนวนการแบ่งส่วนตัวแปรฟัซซีนั้นไม่ส่งผลกระทบใด ๆ กับระบบเลย ดังนั้นเพื่อปรับปรุงเวลาในการเข้าสู่สถานะอยู่ตัวของระบบการเคลื่อนที่ของหัวอ่าน ได้ดำเนินการออกแบบตัวควบคุมฟัซซีอีกหนึ่งชุดเพื่อช่วยปรับปรุงในสถานะชั่วคราวให้ดีขึ้น และกฎชุดที่สองยังคงเป็นกฎชุดเดิมที่ได้ออกแบบไว้ก่อนหน้านี้แล้วในหัวข้อที่ 3.3 เพื่อรักษาเสถียรภาพในสถานะอยู่ตัว จากการวิเคราะห์พฤติกรรมของระบบสามารถกำหนดกฎการควบคุมชุดแรกได้ดังนี้

Begin: $u = 0.08 \text{ V}$

IF ($x \geq 0.6 \text{ } \mu\text{m}$) THEN ($u = -0.02 \text{ V}$)

IF ($t \geq 0.0015 \text{ sec}$ AND $t \leq 0.0016 \text{ sec}$) THEN ($u = 0.01 \text{ V}$).

เมื่อดำเนินการควบคุมด้วยกฎชุดแรกเรียบร้อยแล้วก็จะสวิตช์กฎชุดที่สองให้ร่วมทำงาน ทั้งนี้ในส่วนของฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตยังคงรูปแบบเป็นแบบ Sugeno เช่นเดิม การตั้งกฎการควบคุม กระบวนการฟัซซีฟิเคชันและดีฟัซซีฟิเคชันเป็นแบบ Sugeno เช่นเดิม นอกจากนี้ค่าพารามิเตอร์ต่าง ๆ ของตัวควบคุมแบบกฎสองอินพุตยังคงเป็นเช่นเดิม ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านหลังจากได้เพิ่มกฎการควบคุมเป็นสองชุดแสดงได้ดังรูปที่ 3.11



รูปที่ 3.11 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านกรณีมีการควบคุมแบบต่าง ๆ

เมื่อพิจารณาผลการตอบสนองจากการจำลองสถานการณ์กรณีที่มีการควบคุมด้วยฟัซซีแบบกฎฟัซซีสองชุดและไม่มีการควบคุมใด ๆ ซึ่งแสดงในรูปที่ 3.11 จะเห็นว่าเวลาในการเข้าสู่สถานะอยู่ตัวกรณีที่มีการควบคุมแบบฟัซซีนี้นั้นเท่ากับ 1.6 มิลลิวินาทีและไม่มีการพุ่งเกินเกิดขึ้น จะเห็นว่า การควบคุมแบบฟัซซีนี้นั้นได้ลดทอนการพุ่งเกินลงไปและยังช่วยลดเวลาในการเข้าสู่สถานะอยู่ตัวได้ดีขึ้น และเมื่อเปรียบเทียบกับผลงานวิจัยที่มีมาก่อนของ B.M. Chen และคณะที่ใช้วิธีการควบคุมแบบพีไอดี และการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) นั้นจะเห็นว่า การควบคุมด้วยฟัซซีแบบกฎฟัซซีสองชุดมีประสิทธิภาพดีกว่าการควบคุมแบบพีไอดี โดยมีเวลาในการเข้าสู่สถานะอยู่ตัวน้อยกว่าและไม่เกิดการพุ่งเกิน ส่วนกรณีที่มีการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) นั้นใช้เวลาในการสู่สถานะอยู่ตัวดีที่สุดเพียง 0.8 มิลลิวินาที และยังไม่มีการพุ่งเกินเกิดขึ้น

เมื่อได้ออกแบบตัวควบคุมไว้กับการเคลื่อนที่ของหัวอ่านเพื่อเคลื่อนที่ไปในตำแหน่ง $1\ \mu\text{m}$ แล้วนั้นพบว่าหัวอ่านสามารถเคลื่อนที่ไปได้และใช้เวลาเคลื่อนที่น้อยลง ดังนั้นจึงได้ออกแบบกฎการควบคุมชุดแรกเพิ่มเติมเพื่อให้หัวอ่านเคลื่อนที่ไปครั้งละ $1\ \mu\text{m}$ ให้เคลื่อนที่ไปในตำแหน่ง $2\ \mu\text{m}$ $3\ \mu\text{m}$ $4\ \mu\text{m}$ และ $5\ \mu\text{m}$ ตามลำดับ โดยกฎชุดที่สองนี้ยังคงเป็นกฎชุดเดียวกับที่แสดงในหัวข้อที่ 3.3 ทั้งนี้ในส่วนของฟังก์ชันความเป็นสมาชิกของตัวแปรเอาต์พุตยังคงรูปแบบเป็นแบบ Sugeno การตั้งกฎการควบคุมกระบวนการฟัซซีฟิเคชันและดีฟัซซีฟิเคชันเป็นแบบ Sugeno เช่นเดิม โดยกฎการควบคุมชุดแรกแสดงได้ดังนี้ และผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านแสดงได้ดังรูปที่ 3.12

การเคลื่อนที่ไปที่ตำแหน่ง $1 \mu m$

Begin: $u = 0.08 \text{ V}$

IF ($x \geq 0.6 \mu m$) THEN ($u = -0.02 \text{ V}$)

IF ($t \geq 0.0015 \text{ sec}$ AND $t \leq 0.0016 \text{ sec}$) THEN ($u = 0.01 \text{ V}$).

การเคลื่อนที่ไปที่ตำแหน่ง $2 \mu m$

Begin: $u = 0.016 \text{ V}$

IF ($x \geq 1.4 \mu m$) THEN ($u = 0 \text{ V}$)

IF ($t \geq 0.0115 \text{ sec}$ AND $t \leq 0.0116 \text{ sec}$) THEN ($u = 0.02 \text{ V}$).

การเคลื่อนที่ไปที่ตำแหน่ง $3 \mu m$

Begin: $u = 0.24 \text{ V}$

IF ($x \geq 2.3 \mu m$) THEN ($u = 0.008 \text{ V}$)

IF ($t \geq 0.0215 \text{ sec}$ AND $t \leq 0.0216 \text{ sec}$) THEN ($u = 0.025 \text{ V}$).

การเคลื่อนที่ไปที่ตำแหน่ง $4 \mu m$

Begin: $u = 0.4 \text{ V}$

IF ($x \geq 3.2 \mu m$) THEN ($u = 0.02 \text{ V}$)

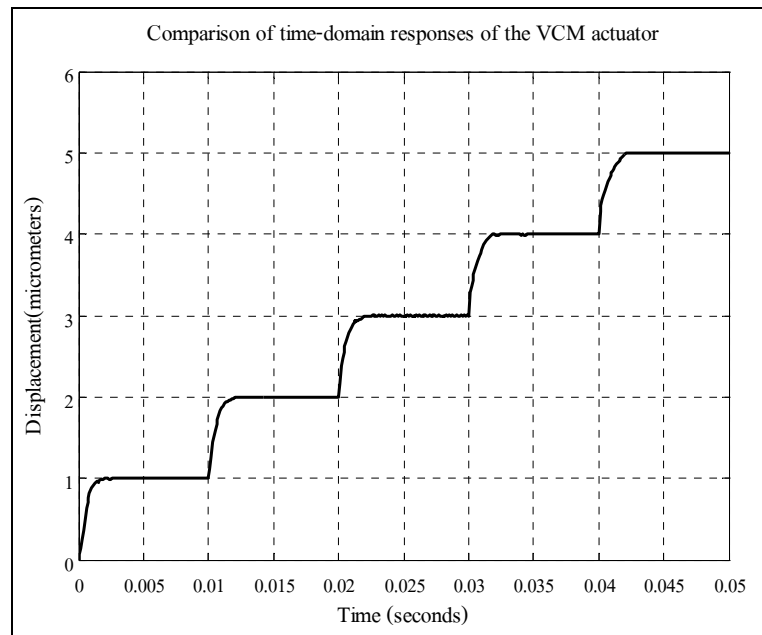
IF ($t \geq 0.0315 \text{ sec}$ AND $t \leq 0.0316 \text{ sec}$) THEN ($u = 0.03 \text{ V}$).

การเคลื่อนที่ไปที่ตำแหน่ง $5 \mu m$

Begin: $u = 0.4 \text{ V}$

IF ($x \geq 4.25 \mu m$) THEN ($u = 0.02 \text{ V}$)

IF ($t \geq 0.0411 \text{ sec}$ AND $t \leq 0.0412 \text{ sec}$) THEN ($u = 0.03 \text{ V}$).



รูปที่ 3.12 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านเมื่อเคลื่อนที่ไปครั้งละ $1 \mu m$

เมื่อออกแบบกฎฟัซซี่ให้หัวอ่านเคลื่อนที่ไปข้างหน้าครั้งละ $1 \mu m$ ไปจนถึงตำแหน่ง $5 \mu m$ จึงได้ออกแบบกฎฟัซซี่ชุดแรกเพิ่มเติมเพื่อให้หัวอ่านเคลื่อนที่กลับไปครั้งละ $1 \mu m$ ในตำแหน่ง $4 \mu m$ $3 \mu m$ $2 \mu m$ และ $1 \mu m$ ตามลำดับ โดยกฎชุดที่สองนี้ยังคงเป็นกฎชุดเดียวกับที่แสดงในหัวข้อที่ 3.3 กฎการควบคุมชุดแรกแสดงได้ดังต่อไปนี้

การเคลื่อนที่กลับไปตำแหน่ง $4 \mu m$

Begin: $u = 0.1 V$

IF ($x \leq 4.8 \mu m$) THEN ($u = 0.02V$)

IF ($t \geq 0.0515 \text{ sec}$ AND $t \leq 0.0516 \text{ sec}$) THEN ($u = 0.025 V$).

การเคลื่อนที่กลับไปตำแหน่ง $3 \mu m$

Begin: $u = 0.1 V$

IF ($x \leq 3.8 \mu m$) THEN ($u = 0.02V$)

IF ($t \geq 0.0615 \text{ sec}$ AND $t \leq 0.0616 \text{ sec}$) THEN ($u = 0.025 V$).

การเคลื่อนที่กลับไปตำแหน่ง $2 \mu m$

Begin: $u = 0.06 V$

IF ($x \leq 2.4 \mu m$) THEN ($u = 0.008V$)

IF ($t \geq 0.0515 \text{ sec}$ AND $t \leq 0.0516 \text{ sec}$) THEN ($u = 0.025 V$).

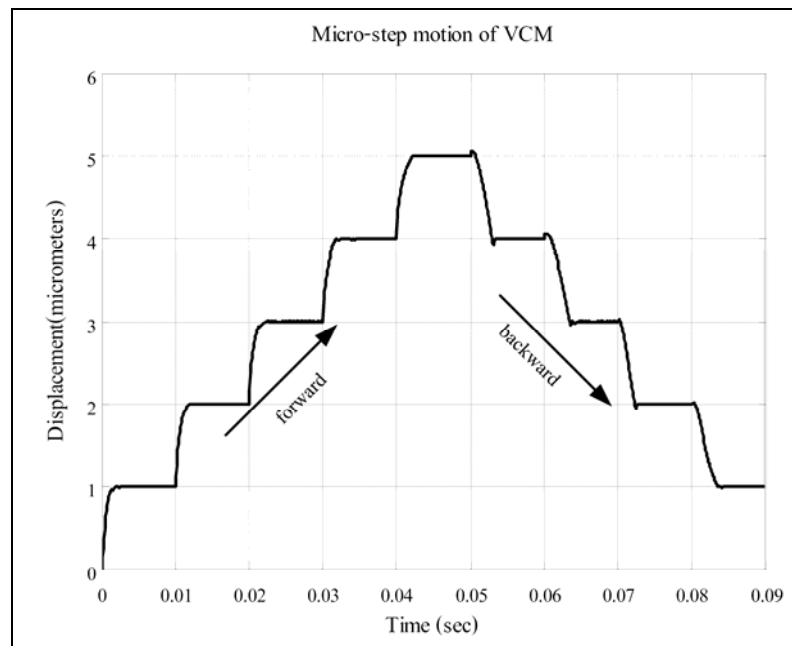
การเคลื่อนที่กลับไปตำแหน่ง $1 \mu m$

Begin: $u = 0.06 \text{ V}$

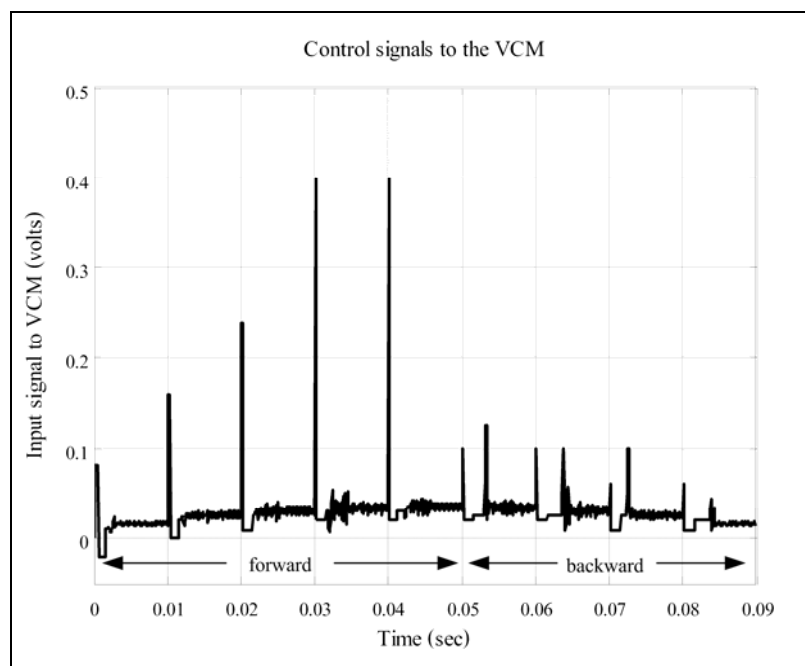
IF ($x \leq 1.6 \text{ } \mu\text{m}$) THEN ($u = 0.008\text{V}$)

IF ($t \geq 0.0515 \text{ sec AND } t \leq 0.0516 \text{ sec}$) THEN ($u = 0.02 \text{ V}$).

ดำเนินการจำลองสถานการณ์โดยให้หัวอ่านเคลื่อนที่ไปครั้งละ $1 \text{ } \mu\text{m}$ ไปถึงตำแหน่ง $5 \text{ } \mu\text{m}$ และให้เคลื่อนที่กลับในทำนองเดียวกัน ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านแสดงได้ดังรูปที่ 3.13 และรูปที่ 3.14 แสดงสัญญาณการควบคุม จากผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านจะเห็นกฎฟuzzyที่ออกแบบนั้นสามารถทำให้หัวอ่านเคลื่อนที่ไปในตำแหน่งที่ต้องการได้ และเวลาในการเข้าสู่สถานะอยู่ตัวคงที่ด้วย



รูปที่ 3.13 ผลการจำลองสถานการณ์การเคลื่อนที่ของหัวอ่านเมื่อเคลื่อนที่ไปครั้งละ $1 \text{ } \mu\text{m}$



รูปที่ 3.14 สัญญาณการควบคุม

3.9 สรุป

การออกแบบตัวควบคุมฟิซซีแบบกฎสองอินพุต เพื่อลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านฮาร์ดดิสก์ไครฟ์ที่ปรากฏสถานะไม่เป็นเชิงเส้นของแรงเสียดทานต่าง ๆ นั้น จากผลการจำลองสถานการณ์จะเห็นได้ว่าตัวควบคุมฟิซซีแบบกฎสองอินพุตสามารถทำให้หัวอ่านเข้าสู่เสถียรภาพได้ และยังลดเวลาในการเข้าสู่สถานะอยู่ตัว อีกทั้งยังไม่มีอาการฟุ้งเกินเกิดขึ้น แต่เมื่อเปรียบเทียบกับผลงานวิจัยของ Chen และคณะที่ใช้วิธีการควบคุมแบบฟิโอดี และการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) ซึ่งแสดงในรูปที่ 3.6 นั้นใช้เวลาในการเข้าสู่สถานะอยู่ตัวน้อยกว่าตัวควบคุมฟิซซีแบบกฎสองอินพุต ดังนั้นเพื่อลดเวลาในการเข้าสู่สถานะอยู่ตัว จึงได้ปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิกเป็นรูป สามเหลี่ยมและสี่เหลี่ยมคางหมูแบบไม่สมมาตร พบว่าการปรับเปลี่ยนรูปแบบฟังก์ชันความเป็นสมาชิกไม่สามารถช่วยลดเวลาในการเข้าสู่สถานะอยู่ตัวได้ รวมทั้งได้ดำเนินการเพิ่มจำนวนการแบ่งส่วนตัวแปรฟิซซี ซึ่งผลจากการจำลองสถานการณ์แสดงให้เห็นว่าการเพิ่มจำนวนการแบ่งส่วนตัวแปรฟิซซีไม่ส่งผลต่อเวลาในการเข้าสู่สถานะอยู่ตัวเลย ดังนั้นเพื่อปรับปรุงเวลาในการเข้าสู่สถานะอยู่ตัวของระบบการเคลื่อนที่ของหัวอ่าน ได้ดำเนินการออกแบบกฎการควบคุมอีกหนึ่งชุดเพื่อช่วยปรับปรุงในสถานะชั่วคราวให้ดีขึ้น และใช้กฎชุดที่สองซึ่งเป็นกฎแบบฟิซซีเพื่อรักษาเสถียรภาพในสถานะอยู่ตัว จากผลการจำลองสถานการณ์จะเห็นว่าตัวควบคุมนี้ทำให้หัวอ่านเคลื่อนที่สู่สถานะอยู่ตัวในเวลาเพียง 1.6 มิลลิวินาที และไม่มีอาการฟุ้งเกินเกิดขึ้นซึ่งมีสมรรถนะดีกว่า เมื่อนำมาเปรียบเทียบกับวิธีการควบคุมแบบ ฟิโอดี ส่วนการควบคุมแบบ Composite Nonlinear Feedback Control (CNF) นั้นใช้เวลาในการเข้าสู่สถานะอยู่ตัวเพียง 0.8

มิลลิวินาที แต่เมื่อศึกษาถึงวิธีการออกแบบตัวควบคุม CNF แล้ววิธีการออกแบบนั้นยุ่งยากซับซ้อนไม่สามารถดำเนินการได้โดยง่าย

บทที่ 4

การปรับปรุงเสถียรภาพและสมรรถนะของหัวอ่านฮาร์ดดิสก์ไครฟ์ผ่านความถี่สูง

4.1 บทนำ

เมื่อพิจารณาผลตอบแทนทางพลวัตของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่สูง พบว่ามีรีโซแนนซ์เกิดขึ้นกับระบบทำให้ระบบขาดเสถียรภาพดังที่ได้อธิบายไว้ในบทที่ 2 ดังนั้นจึงจำเป็นต้องออกแบบตัวชดเชยเพื่อลดทอนผลกระทบจากสภาวะรีโซแนนซ์ และการชดเชยทางเสถียรภาพให้กับระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ ในการปฏิบัติการดังกล่าวได้อาศัยแบบจำลองทางคณิตศาสตร์ของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ ดังแสดงรายละเอียดไว้ในบทที่ 2 สำหรับใช้ในการออกแบบตัวชดเชย แนวทางหนึ่งในการออกแบบตัวชดเชยได้แก่ การใช้ตัวชดเชยแบบเฟสล้ำหน้า (สราวุฒิ สุจิตจร, 2546) ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อน (Messner, Bedillion, Xia, and Kams, 2007) และตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$

เนื้อหาในบทนี้จะได้กล่าวถึงวิธีการออกแบบตัวชดเชยแบบ การใช้ตัวชดเชยแบบเฟสล้ำหน้า ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อน และตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$ เพื่อสร้างเสถียรภาพให้กับระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ที่ปรากฏสภาวะรีโซแนนซ์ ซึ่งจะดำเนินการจำลองสถานการณ์ให้กับระบบโดยใช้แบบจำลองทางคณิตศาสตร์ของระบบในการออกแบบตัวชดเชย ทั้งนี้จะทำการพิจารณาทั้งผลตอบแทนทางเวลาและทางความถี่ของระบบ เพื่อพิจารณาประสิทธิผลของตัวชดเชยที่ได้ออกแบบ โดยมีข้อกำหนดทางเทคนิคในการออกแบบเป็น การพุ่งเกินไม่เกิน 10% ค่าผิดพลาดในสถานะอยู่ตัวอยู่ในช่วง $\pm 1\%$ ส่วนเฟื่อเฟสมากกว่า 30° และส่วนเฟื่ออัตราขยาย มากกว่า 6dB

4.2 การใช้ตัวชดเชยแบบเฟสล้ำหน้า

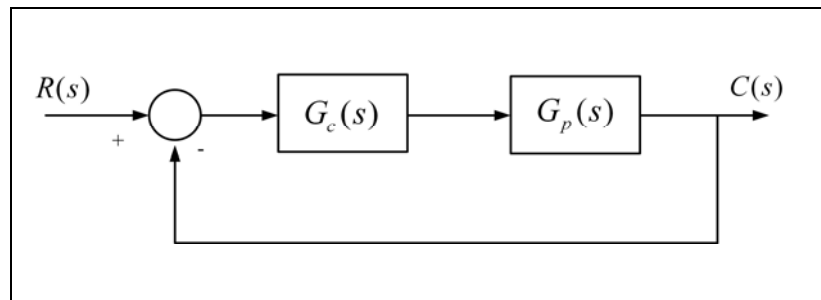
การชดเชยแบบเฟสล้ำหน้า (phase lead compensation) หมายถึงการชดเชยที่กระทำกับระบบ โดยอาศัยตัวชดเชยที่มีลักษณะสมบัติทางเฟสเป็นบวก โครงสร้างของระบบที่ใช้ตัวชดเชยแบบการชดเชยแบบเฟสล้ำหน้า สามารถแสดงได้ดังรูปที่ 4.1 ซึ่งตัวชดเชยมีฟังก์ชันถ่ายโอนดังสมการที่ (4-1)

$$G_c(s) = \frac{K(s+z)}{(s+p)} \quad (4-1)$$

โดยที่ K คือ อัตราขยาย

z คือ ซีโร

p คือ โพล



รูปที่ 4.1 โครงสร้างของระบบที่มีการชดเชย

โดยที่ $R(s)$ คือ สัญญาณอินพุตอ้างอิง

$C(s)$ คือ สัญญาณเอาต์พุต

$G_c(s)$ คือ ตัวชดเชย (compensator)

$G_p(s)$ คือ ระบบการเคลื่อนที่ของหัวอ่าน

การออกแบบตัวชดเชยแบบเฟสล้ำหน้า จะทำการออกแบบเพื่อเลือกค่า โพลและซีโรให้ได้ ผลตอบสนองตามต้องการ สามารถดำเนินงานออกแบบตามขั้นตอนต่อไปนี้ (สราวุธ สุจิตจร, 2546)

ขั้นตอนที่ 1 ตรวจสอบค่า PM ของระบบก่อนมีการชดเชย ถ้าพบว่า PM ของระบบก่อนมีการชดเชย หรือ PM_{uncomp} น้อยกว่า PM ตามข้อกำหนดสมรรถนะ หรือ PM_{comp} ก็ให้ดำเนินงานในขั้นตอนที่สอง ซึ่งค่า PM ของระบบเท่ากับ -1.62° ซึ่งไม่เป็นไปตามข้อกำหนดสมรรถนะคือ 30°

ขั้นตอนที่ 2 กำหนดค่า ϕ_{max} จาก

$$\phi_{max} = PM_{comp} - PM_{uncomp} + 5^\circ \cong 60^\circ$$

ขั้นตอนที่ 3 กำหนดค่า α จาก

$$\alpha = \frac{1 + \sin \phi_{max}}{1 - \sin \phi_{max}} = 13.92$$

ขั้นตอนที่ 4 คำนวณค่า $-10\log \alpha = -11.4\text{ dB}$ และตรวจสอบแผนภาพโพลของระบบ ก่อนการชดเชย ว่า กราฟของขนาดปรากฏปริมาณ -11.4 dB นี้ที่ความถี่ $\omega_{m1} = 4.32 \times 10^4$ เรเดียน/วินาที

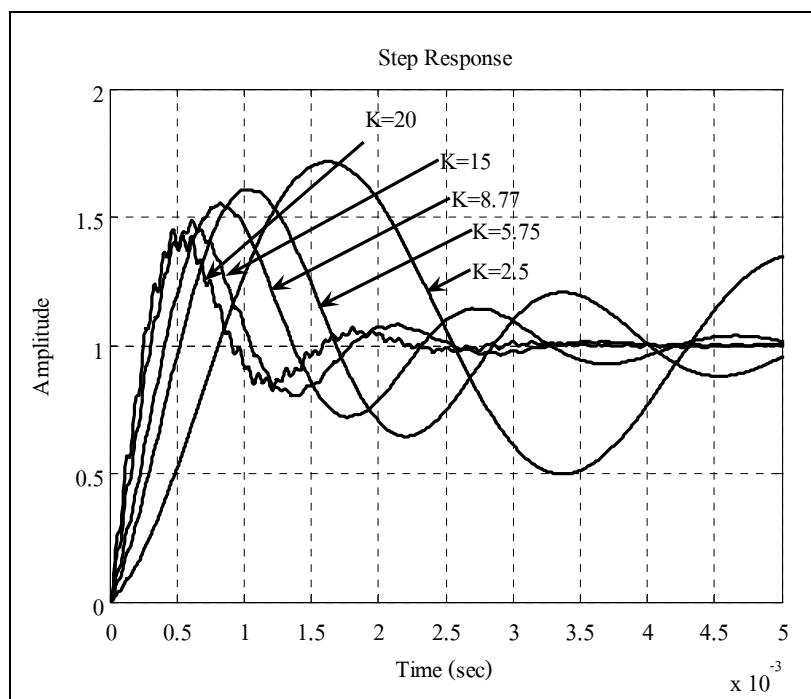
ขั้นตอนที่ 5 คำนวณโพลของตัวชดเชย $p = \omega_m \sqrt{\alpha}$ และ ซีโร $z = \frac{p}{\alpha}$
 ดังนั้น $p = \omega_m \sqrt{\alpha} = 161177.11$ และ $z = \frac{p}{\alpha} = 11578.81$
 จะได้ตัวชดเชยมีฟังก์ชันถ่ายโอนดังสมการที่ (4-2)

$$G_c(s) = K \left(\frac{s+11578}{s+161177} \right) \quad (4-2)$$

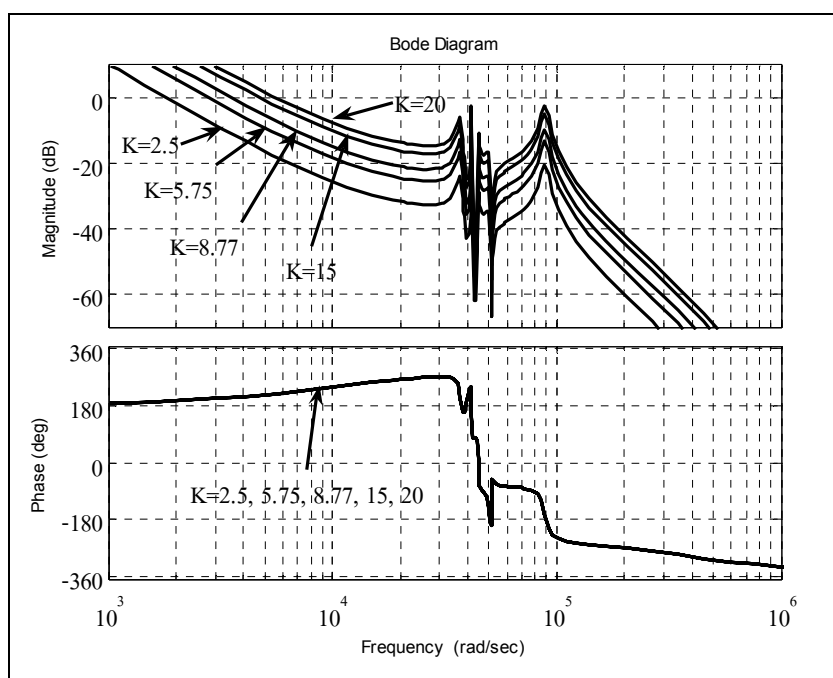
การดำเนินงานตามขั้นตอนข้างต้น ได้ปฏิบัติสามารถเพราะการสร้างเสถียรภาพทำได้ยาก จึงต้องใช้ตัวชดเชยอันดับสาม การออกแบบตัวชดเชยอีกสองชุดที่เสริมเข้าไบนั้น มีการดำเนินงานคล้ายกับการดำเนินงานข้างต้นโดยเลือกค่า PM_{comp} เท่ากับ 15° และ 50° ตามลำดับเมื่อออกแบบเสร็จแล้ว อาจสรุปได้ว่าตัวชดเชยแบบเฟสล้ำหน้าอันดับสามมีฟังก์ชันถ่ายโอนแสดงได้ดังสมการที่ (4-3)

$$G_{c1}(s) = K \left(\frac{s+11578}{s+161177} \right) \left(\frac{s+60049.648}{s+453374.8449} \right) \left(\frac{s+13429}{s+22803} \right) \quad (4-3)$$

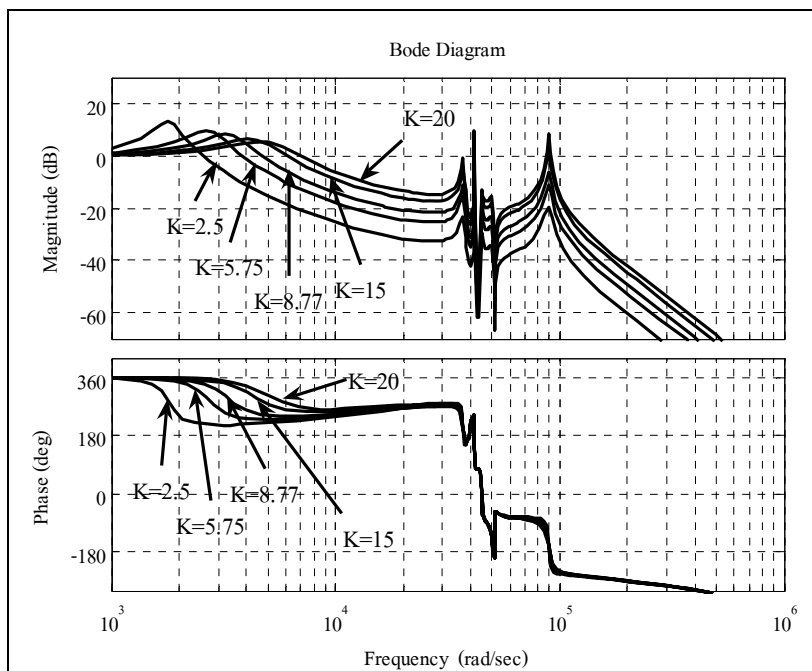
ขั้นตอนที่ 6 เมื่อออกแบบเสร็จสิ้น ให้ปรับอัตราขยายคิซีให้ถูกต้อง จากการลองปรับจูนค่า K เป็น 2.5, 5.75, 8.77, 15 และ 20 จะได้ผลตอบสนองทางเวลาดังแสดงในรูปที่ 4.2 แผนภาพโพล ของระบบวงเปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.3 และแผนภาพการตอบสนองทางความถี่ของระบบวงปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.4 และค่าการตอบสนองของระบบแสดงได้ดังตารางที่ 4.1 ตามลำดับ



รูปที่ 4.2 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วยเฟสล้ำหน้าเมื่อปรับจูนค่า K



รูปที่ 4.3 แผนภาพโบดของระบบวงเปิดกรณีมีการชดเชยด้วยเฟสล้ำหน้าเมื่อปรับจูนค่า K



รูปที่ 4.4 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วยเฟสล้ำหน้า
เมื่อปรับจูนค่า K

ตารางที่ 4.1 ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับจูนค่า K กรณีมีการชดเชยด้วยเฟสล้ำหน้า

K	การพุ่งเกิน (%)	เวลาในการเข้าสู่สถานะอยู่ตัว(ms)	ส่วนเฟื่อเฟส (°)	ส่วนเฟื่ออัตราขยาย (dB)	แบนด์วิดท์ เรเดียนต่อวินาที
2.5	71.6	19.3	12.9	20.6	2,824
5.75	60.9	8.26	19.5	13.4	4,296
8.77	55.3	4.97	24.1	9.73	5,360
15	49	3.1	31.7	5.07	7,019
20	45	2.76	36.6	2.57	8,166

เมื่อทำการปรับจูนค่า K ดังแสดงในตารางที่ 4.1 พบว่าเมื่อค่า K มีค่าเพิ่มขึ้นจะทำให้ค่า การพุ่งเกิน เวลาในการเข้าสู่สถานะอยู่ตัว ส่วนเฟื่อเฟส และค่าแบนด์วิดท์ มีสมรรถนะดีขึ้น คือ การพุ่งเกินและเวลาในการเข้าสู่สถานะอยู่ตัวมีค่าลดลง ส่วนเฟื่อเฟสและแบนด์วิดท์มีค่าเพิ่มขึ้น แต่ส่วนเฟื่ออัตราขยายนั้นกลับลดลง เมื่อพิจารณาผลตอบสนองทางเวลาดังแสดงในรูปที่ 4.2 จะสังเกตเห็นได้ว่า ถ้าค่า K มีค่าเพิ่มขึ้นจะทำให้การตอบสนองปรากฏการสั่นไกวมากขึ้น ซึ่งเมื่อพิจารณาสมรรถนะตามข้อกำหนดแล้วผล

การตอบสนองกรณีมีการชดเชยด้วยเฟสล่าช้ายังไม่เป็นไปตามข้อกำหนด แสดงว่าตัวชดเชยดังกล่าวที่ได้ออกแบบยังไม่เหมาะสมต่อระบบ

4.3 การใช้ตัวชดเชยแบบเฟสล่าช้าล้าหลังเชิงซ้อน

ดำเนินการออกแบบตัวชดเชยด้วยวิธีเฟสล่าช้าและล้าหลังเชิงซ้อนเป็นวิธีของ Massner และคณะ (Messner, Bedillion, Xia, and Karns, 2007) ซึ่งตัวชดเชยมีฟังก์ชันถ่ายโอนดังสมการที่ (4-4) และมีขั้นตอนแสดงวิธีการออกแบบค่า ω_p และ ω_z ดังนี้

$$G_{\text{complex}}(s) = \frac{\omega_p}{\omega_z} \left(\frac{s^2 + 2\zeta\omega_z s + \omega_z^2}{s^2 + 2\zeta\omega_p s + \omega_p^2} \right) \quad (4-4)$$

ขั้นตอนที่ 1 กำหนดค่า $\phi_{\max 2}$ ซึ่งได้เลือกใช้ค่า 45° เพื่อการออกแบบ (arbitrary)

ขั้นตอนที่ 2 กำหนดค่า $\omega_{m2} = 4.2 \times 10^4$ เรเดียน/วินาที และ ค่า $\zeta = 0.9$ (arbitrary)

ขั้นตอนที่ 3 คำนวณค่า ω_p และ ω_z จาก

$$\omega_p = \omega_{m2} \left(\zeta \tan \phi_{\max 2} + \sqrt{\zeta^2 \tan^2 \phi_{\max 2} + 1} \right) = 9.43 \times 10^4 \text{ เรเดียน/วินาที}$$

$$\omega_z = \omega_{m2} \left(-\zeta \tan \phi_{\max 2} + \sqrt{\zeta^2 \tan^2 \phi_{\max 2} + 1} \right) = 1.87 \times 10^4 \text{ เรเดียน/วินาที}$$

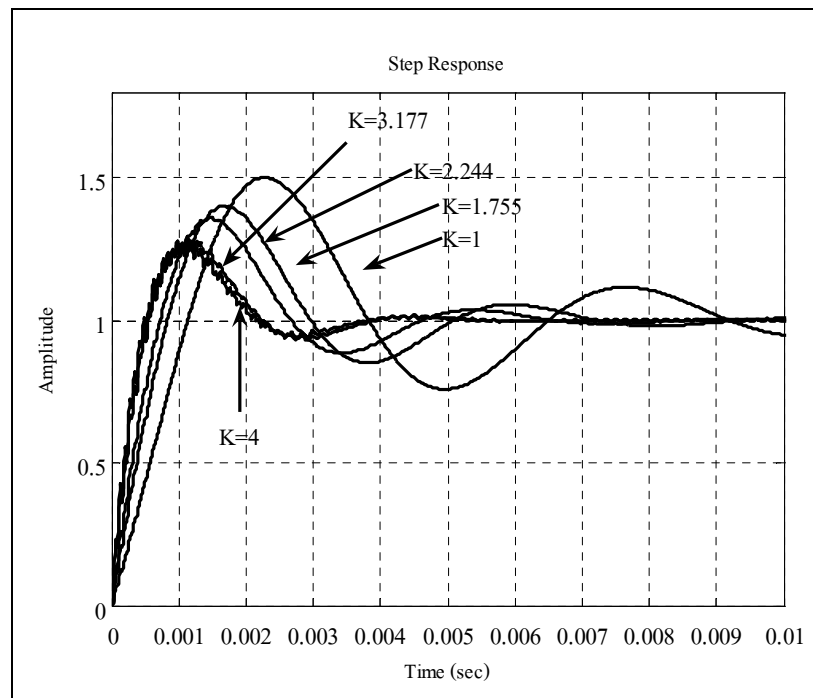
ขั้นตอนที่ 4 แทนค่า ω_p และ ω_z ลงในสมการที่ (4-4) เมื่อแทนค่า ω_p และ ω_z จะได้ตัวชดเชยมีฟังก์ชันถ่ายโอนดังสมการที่ (4-5)

$$G_{\text{complex}}(s) = K \left(\frac{s^2 + 3.367 \times 10^4 s + 3.499 \times 10^8}{s^2 + 1.697 \times 10^5 s + 8.893 \times 10^9} \right) \quad (4-5)$$

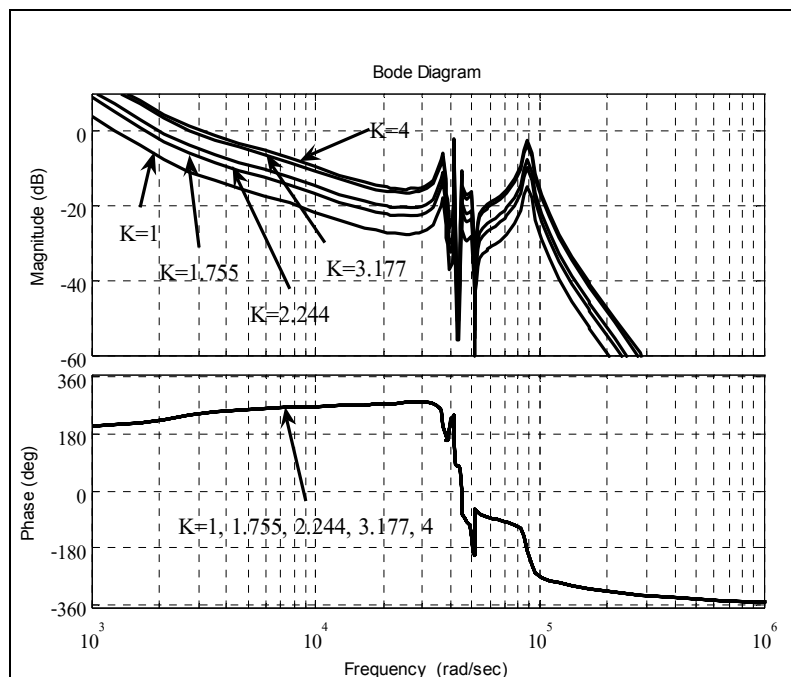
เนื่องจากตัวชดเชยชนิดนี้เป็นอันดับสองเชิงซ้อน การสร้างเสถียรภาพให้หัวอ่านจำเป็นต้องใช้ตัวชดเชยสองชุดร่วมทำงาน จึงต้องดำเนินการออกแบบสองรอบ ที่แสดงข้างต้นเป็นการออกแบบในรอบแรก ซึ่งใช้ค่า $\phi_{\max 2}$ เป็น 45° (arbitrary), $\omega_{m2} = 4.2 \times 10^4$ เรเดียน/วินาที และ $\zeta = 0.9$ ในการออกแบบรอบที่สอง ใช้ค่า $\phi_{\max 2}$ เป็น 30° , $\omega_{m2} = 4.53 \times 10^3$ เรเดียน/วินาที และ $\zeta = 0.8$ ตามลำดับ เมื่อดำเนินการเสร็จสมบูรณ์ได้ตัวชดเชยแบบเฟสล่าช้าและล้าหลังเชิงซ้อนอันดับสี่มีฟังก์ชันถ่ายโอนแสดงดังสมการที่ (4-6)

$$G_{C2}(s) = K \left(\frac{s^2 + 3.367 \times 10^4 s + 3.499 \times 10^8}{s^2 + 1.697 \times 10^5 s + 8.893 \times 10^9} \right) \left(\frac{s^2 + 4636s + 8.396 \times 10^6}{s^2 + 1.33 \times 10^4 s + 5.016 \times 10^7} \right) \quad (4-6)$$

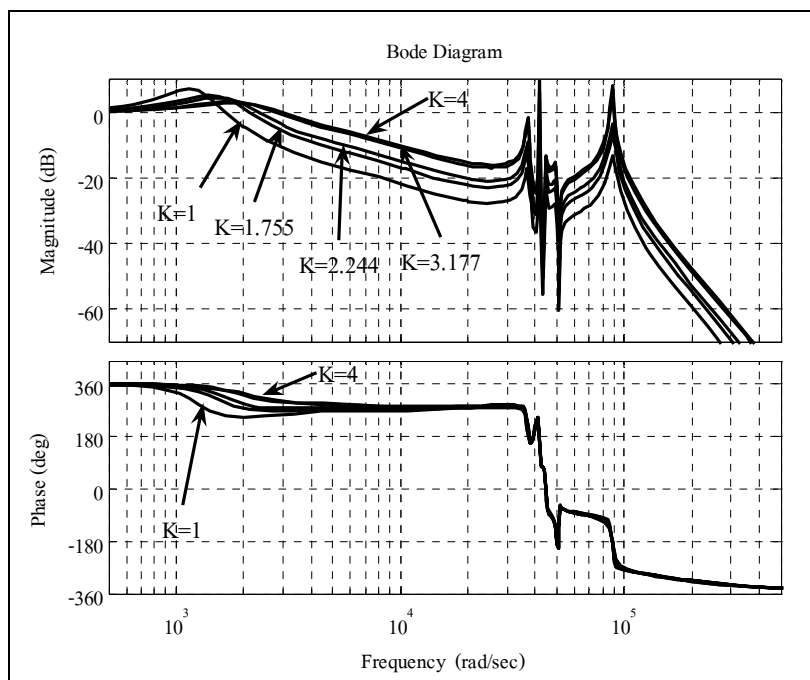
ขั้นตอนที่ 5 ให้ปรับอัตราขยายคิซีให้ถูกต้อง จากการลองปรับจูนค่า K เป็น 1 1.755 2.244 3.177 และ 4 จะได้ผลตอบสนองทางเวลาดังแสดงในรูปที่ 4.5 แผนภาพโบคของระบบวงเปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.6 และแผนภาพการตอบสนองทางความถี่ของระบบวงปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.7 และค่าการตอบสนองของระบบแสดงได้ดังตารางที่ 4.2 ตามลำดับ



รูปที่ 4.5 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วยเฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K



รูปที่ 4.6 แผนภาพโบคของระบบวงเปิดกรณีมีการชดเชยด้วย
เฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K



รูปที่ 4.7 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วย
เฟสล้ำหน้าและล้ำหลังเชิงซ้อนเมื่อปรับจูนค่า K

ตารางที่ 4.2 ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับจูนค่า K กรณีมีการชดเชยด้วยเฟสล้ำหน้าและล่าหลังเชิงซ้อน

K	การพุ่งเกิน (%)	เวลาในการเข้าสู่สถานะอยู่ตัว(ms)	ส่วนเฟส (°)	ส่วนเพื่ออัตราขยาย (dB)	แบนด์วิดท์ เรเดียนต่อวินาที
1	50	13.7	27.2	14.5	1,879
1.755	40.2	8.23	37.6	9.66	2,489
2.244	36.4	6.05	43.5	7.53	2,850
3.177	30.4	4.69	53.8	4.51	3,474
4	28.4	3.45	61.9	2.51	4,056

เมื่อทำการปรับจูนค่า K ดังแสดงในตารางที่ 4.2 พบว่าเมื่อค่า K มีค่าเพิ่มขึ้นจะทำให้ค่า การพุ่งเกินและเวลาในการเข้าสู่สถานะอยู่ตัวมีค่าลดลง ส่วนเฟสและแบนด์วิดท์มีค่าเพิ่มขึ้น และส่วนเพื่ออัตราขยายนั้นมีค่าลดลง และเมื่อพิจารณาผลตอบสนองทางเวลาดังแสดงในรูปที่ 4.2 จะสังเกตเห็นได้ว่า ถ้าค่า K มีค่าเพิ่มขึ้นจะทำให้การตอบสนองปรากฏการสั่นไกวมากขึ้น ถ้าเราพิจารณาแต่เพียงข้อกำหนดสมรรถนะในโดเมนความถี่ จะพบว่าเมื่อค่า K เท่ากับ 1.755 และ 2.244 เท่านั้น ที่ทำให้ระบบเมื่อผ่านการชดเชยแล้ว มีส่วนเฟสเสถียรภาพตามประสงค์ทุกประการ แต่เมื่อพิจารณาสมรรถนะของระบบในโดเมนเวลาแล้วพบว่าไม่มีค่า K ใดเลยที่ทำให้สมรรถนะของระบบในโดเมนเวลาเป็นไปตามข้อกำหนด

4.4 การใช้ตัวชดเชยแบบเฟสล้ำหน้าล่าหลังเชิงซ้อนตามวิธีของ Messner และคณะเมื่อใช้ค่า $\zeta > 1$

การออกแบบตัวชดเชยในหัวข้อที่ 4.4 นี้ เป็นการออกแบบวิธีเดียวกันกับหัวข้อที่ 4.3 มีข้อแตกต่างอยู่ที่ การกำหนดให้ค่า $\zeta > 1$ ดำเนินการออกแบบตัวชดเชยตามได้ดังขั้นตอนต่อไปนี้

ขั้นตอนที่ 1 กำหนดค่า $\phi_{\max 3}$ ซึ่งได้เลือกใช้ค่า 30° เพื่อการออกแบบ

ขั้นตอนที่ 2 กำหนดค่า $\omega_{m3} = 4.18 \times 10^4$ เรเดียน/วินาที และ ค่า $\zeta = 5$

ขั้นตอนที่ 3 กำหนดค่า ω_p และ ω_z จาก

$$\omega_p = \omega_{m3} \left(\zeta \tan \phi_{\max 3} + \sqrt{\zeta^2 \tan^2 \phi_{\max 3} + 1} \right) = 2.48 \times 10^5 \text{ เรเดียน/วินาที}$$

$$\omega_z = \omega_{m3} \left(-\zeta \tan \phi_{\max 3} + \sqrt{\zeta^2 \tan^2 \phi_{\max 3} + 1} \right) = 7.03 \times 10^3 \text{ เรเดียน/วินาที}$$

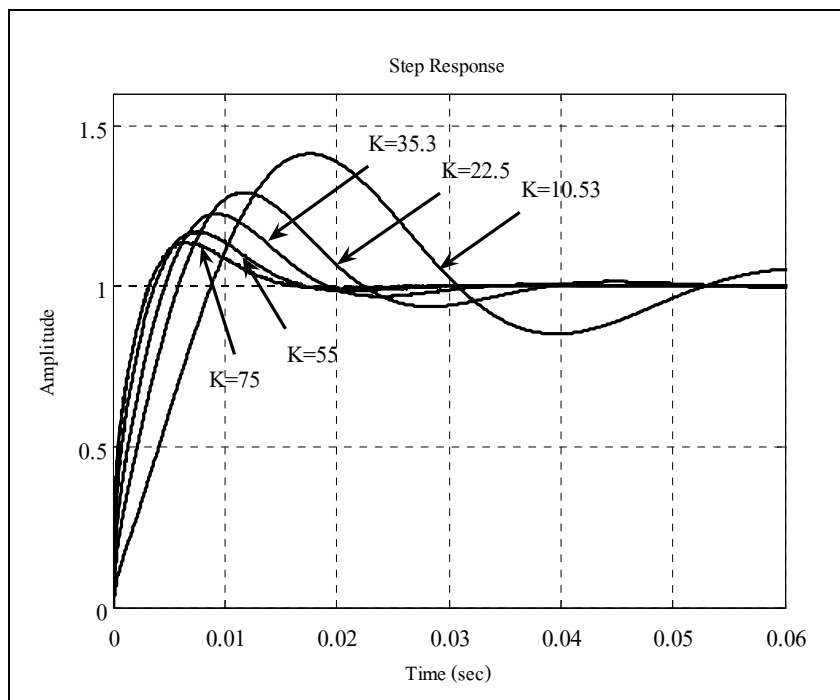
ขั้นตอนที่ 4 แทนค่า ω_p และ ω_z ลงในสมการที่ (4-4) เมื่อแทนค่า ω_p และ ω_z จะได้ตัวชดเชยมีฟังก์ชันถ่ายโอนดังสมการที่ (4-7)

$$G_{\text{complex}}(s) = K \left(\frac{s^2 + 7.035 \times 10^4 s + 4.949 \times 10^7}{s^2 + 2.484 \times 10^5 s + 6.169 \times 10^{10}} \right) \quad (4-7)$$

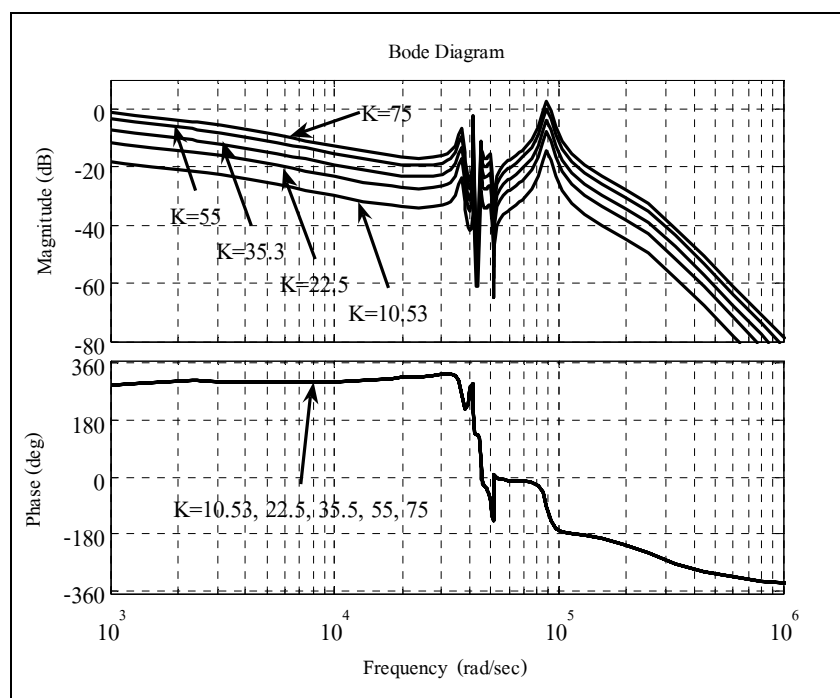
การดำเนินงานออกแบบตัวชดเชยมีความคล้ายกับหัวข้อ 4.3 ที่ผ่านมาซึ่งต้องออกแบบสองรอบ โดยใช้ $\phi_{\max 3}$ เท่ากับ 30° , $\omega_{m3} = 4.18 \times 10^4$ เรเดียน/วินาที และ $\zeta = 5$ ในการออกแบบรอบที่สอง ได้เลือกค่า $\phi_{\max 3}$ เท่ากับ 15° , $\omega_{m3} = 1 \times 10^4$ เรเดียน/วินาที และ $\zeta = 5$ ตามลำดับ เมื่อดำเนินการเสร็จสิ้นสมบูรณ์จะได้ ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังอันดับสี่แบบหน่วงเกิน ($\zeta > 1$) มีฟังก์ชันถ่ายโอนดังแสดงในสมการที่ (4-8)

$$G_{C3}(s) = K \left(\frac{s^2 + 3.321 \times 10^4 s + 1.103 \times 10^7}{s^2 + 3.012 \times 10^5 s + 9.069 \times 10^8} \right) \left(\frac{s^2 + 7.035 \times 10^4 s + 4.949 \times 10^7}{s^2 + 2.484 \times 10^5 s + 6.169 \times 10^{10}} \right) \quad (4-8)$$

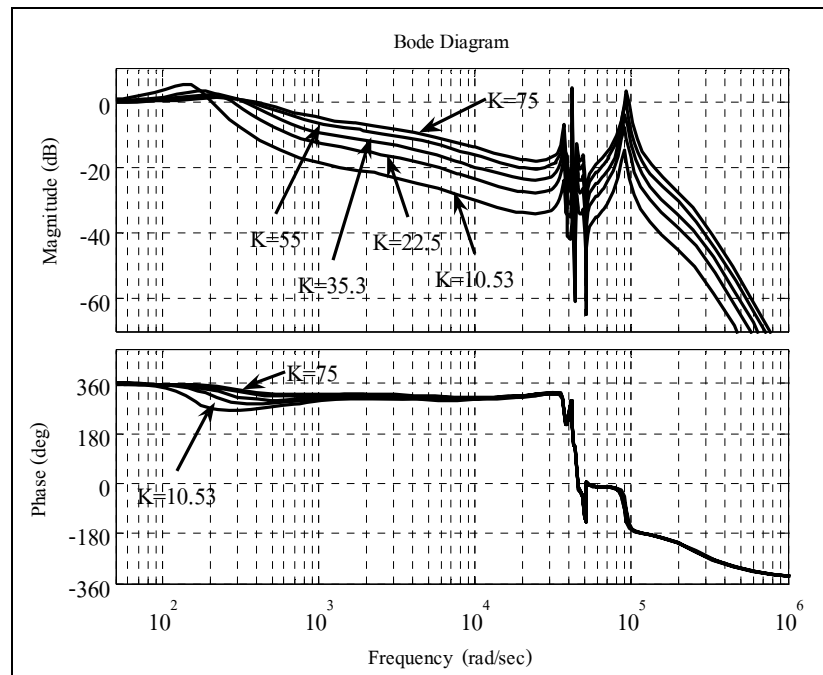
ขั้นตอนที่ 5 ปรับอัตราขยายคิซีให้ถูกต้อง ลองปรับจูนค่า K เป็น 10.53 22.5 35.3 55 และ 75 จะได้ผลตอบสนองทางเวลาดังแสดงในรูปที่ 4.8 แผนภาพโพล ของระบบวงเปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.9 และแผนภาพการตอบสนองทางความถี่ของระบบวงปิดที่มีการชดเชยแล้วแสดงได้ในรูปที่ 4.10 และค่าการตอบสนองของระบบแสดงได้ดังตารางที่ 4.3 ตามลำดับ



รูปที่ 4.8 ผลการตอบสนองทางเวลาของระบบของระบบกรณีมีการชดเชยแบบเฟสล้ำหน้าและล้ำหลัง
อันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K



รูปที่ 4.9 แผนภาพโบดของระบบวงเปิดกรณีมีการชดเชยแบบเฟสล้ำหน้าและล้ำหลัง
อันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K



รูปที่ 4.10 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยแบบเฟสล้ำหน้าและล้ำหลัง
อันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K

ตารางที่ 4.3 ข้อมูลสมรรถนะในโดเมนเวลาและเสถียรภาพสัมพัทธ์เมื่อปรับจูนค่า K กรณีมีการชดเชยด้วย
เฟสล้ำหน้าและล้ำหลังอันดับสี่แบบหน่วงเกิน

K	การพุ่งเกิน (%)	เวลาในการเข้าสู่ สถานะอยู่ตัว(ms)	ส่วนเฟื่อเฟส (°)	ส่วนเฟื่อ อัตราขยาย (dB)	แบนด์วิดท์ เรเดียนต่อวินาที
10.53	41.2	70.9	36.7	22.1	238
22.5	29.2	35.9	54.2	15.5	348
35.3	22.6	28.2	68.6	11.6	439
55	17	15.5	87.1	7.78	564
75	13.6	14.1	45.1	5.09	687

พิจารณาผลตอบสนองทางเวลาและทางความถี่ของระบบการเคลื่อนที่ของหัวอ่านที่ได้รับการ
ชดเชยด้วยตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังอันดับสี่แบบหน่วงเกินเมื่อปรับจูนค่า K ให้มีค่าเพิ่มขึ้นดัง
แสดงในตารางที่ 4.3 จะเห็นว่าถ้าพิจารณาสมรรถนะของระบบในโดเมนความถี่แล้วค่า K เท่ากับ 75
เท่านั้นที่มีส่วนเฟื่อเสถียรภาพไม่เป็นไปตามข้อกำหนด คือมีค่าส่วนอัตราขยายเท่ากับ 5.09 dB ส่วนค่า K
เท่ากับ 10.53 22.5 35.5 55 และ 75 นั้นมีส่วนเฟื่อเสถียรภาพเป็นไปตามข้อกำหนด และเมื่อพิจารณา

สมรรถนะของระบบในโดเมนเวลาพบว่าไม่มีค่า K โดยเลยที่ทำให้สมรรถนะของระบบในโดเมนเวลาเป็นไปตามข้อกำหนด

4.5 สรุป

เนื้อหาในบทนี้ได้กล่าวถึงตัวชดเชยที่มีรูปแบบ 3 แบบด้วยกัน คือ การใช้ตัวชดเชยแบบเฟสล้ำหน้า (สราวุฒิ สุจิตจร, 2546) ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อน (Messner, Bedillion, Xia, and Karns, 2007) และตัวชดเชยแบบเฟสล้ำหน้าล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$ จากผลการจำลองสถานการณ์เพื่อพิจารณาพลวัตของระบบที่ผ่านการชดเชยด้วยตัวชดเชยทั้ง 3 แบบ โดยดำเนินการเพื่อพิจารณาผลตอบสนองทางเวลาและทางความถี่ เมื่อพิจารณาการตอบสนองของระบบที่มีการชดเชยทั้ง 3 แบบดังกล่าวข้างต้น พบว่าถ้าเราพิจารณาแต่เพียงข้อกำหนดสมรรถนะในโดเมนความถี่จะเห็นว่ามีความชดเชย 2 แบบ ที่มีให้ส่วนเพื่อเสถียรภาพตามประสงค์ทุกประการ คือ การใช้ตัวชดเชยแบบเฟสล้ำหน้าล้ำหลังเชิงซ้อน และชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$ แต่เมื่อพิจารณาสมรรถนะของระบบในโดเมนเวลานั้นการชดเชยทั้ง 3 วิธีที่ได้กล่าวมานั้นสมรรถนะของระบบวงปิดในโดเมนเวลายังมีคุณภาพที่ไม่ดีเท่าที่ควร เนื่องจากการพุ่งเกินยังมีอยู่สูงมาก ดังนั้นเพื่อปรับปรุงสมรรถนะของหัวอ่านฮาร์ดดิสก์ให้ดียิ่งขึ้น จึงได้ประยุกต์เทคนิคการค้นหาทางปัญญาประดิษฐ์ซึ่งเรียกว่า การค้นหาแบบตามูเชิงปรับตัว หรือ ATS มาทำการออพติไมซ์ค่าพารามิเตอร์ของตัวชดเชยซึ่งจะอธิบายในบทที่ 5 ต่อไป

บทที่ 5

การหาค่าเหมาะที่สุดด้วยการค้นหาแบบตามูเชิงปรับตัว สำหรับตัวชดเชยรีโซแนนซ์

5.1 บทนำ

จากเนื้อหาในบทที่ 2 และ 4 ที่ผ่านมา เมื่อพิจารณาผลตอบแทนทางพลวัตของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ในย่านความถี่สูง พบว่ามีสภาวะรีโซแนนซ์เกิดขึ้นกับระบบและระบบขาดเสถียรภาพ ดังนั้น จึงจำเป็นต้องออกแบบตัวชดเชยเพื่อลดทอนผลกระทบดังกล่าวให้กับระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ ดังที่ได้บรรยายไว้ในบทที่ 4 ซึ่งได้แสดงผลไว้ว่า สมรรถนะของระบบวงปิดในโดเมนเวลายังมีคุณภาพไม่ดีเท่าที่ควร เนื่องจากการพุ่งเกินยังมีอยู่สูงมาก ดังนั้นเพื่อปรับปรุงสมรรถนะของหัวอ่านฮาร์ดดิสก์ไครฟ์ให้ดียิ่งขึ้น จึงได้ประยุกต์เทคนิคการค้นหาทางปัญญาประดิษฐ์ซึ่งเรียกว่า การค้นหาแบบตามูเชิงปรับตัว หรือ ATS (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004) มาทำการออฟติไมซ์ค่าพารามิเตอร์ของตัวชดเชย การค้นหาแบบตามูเชิงปรับตัวเป็นเทคนิคการค้นหาทางปัญญาประดิษฐ์ที่ทรงประสิทธิภาพเทคนิคหนึ่ง ซึ่งได้รับการพัฒนามาจากการค้นหาแบบตามู (Glover, 1986)

เนื้อหาในบทที่ 5 นี้ กล่าวถึงการค้นหาแบบตามูเชิงปรับตัว การหาค่าเหมาะที่สุดสำหรับตัวชดเชยในย่านความถี่สูง เพื่อสร้างเสถียรภาพให้กับระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ที่ปรากฏสภาวะรีโซแนนซ์ และดำเนินการจำลองสถานการณ์ ทั้งนี้พิจารณาทั้งผลตอบแทนทางเวลาและทางความถี่ของระบบ เพื่อพิจารณาประสิทธิผลของตัวชดเชยที่ได้ออกแบบ โดยมีข้อกำหนดทางเทคนิคในการออกแบบเป็น การพุ่งเกินไม่เกิน 10% ค่าผิดพลาดในสถานะอยู่ตัวอยู่ในช่วง $\pm 1\%$ ส่วนเฟสผิดพลาดมากกว่า 30° และส่วนเฟสอัตราขยายมากกว่า 6dB

5.2 การค้นหาแบบตามูเชิงปรับตัว

ในปัญหาที่มีความซับซ้อนสูง และปรากฏค่าตอบวงแคบเฉพาะถิ่นจำนวนมาก การค้นหาแบบตามูอาจประสบปัญหาเรื่องของการล่อลวงโดยคำตอบวงแคบเฉพาะถิ่น ทำให้ไม่สามารถเข้าสู่หาคำตอบวงกว้าง (global solution) ได้อย่างแท้จริง (กองพัน อารีรักษ์, 2544) การค้นหาแบบตามูเชิงปรับตัว (adaptive Tabusearch, ATS) (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004) (เดชา พวงดาวเรือง, 2547) ได้รับการปรับปรุงมาจากวิธีการค้นหาแบบตามู โดยเพิ่มกลไกย้อนรอยการค้นหา (back-tracking mechanism) เพื่อแก้ปัญหาการล่อลวงโดยคำตอบวงแคบเฉพาะถิ่น และกลไกปรับรัศมีการค้นหา (adaptive radius mechanism) เพื่อเพิ่มความเร็วให้กับกระบวนการค้นหาคำตอบ นอกจากนั้นแล้ว การเคลื่อนที่ของ

กระบวนการค้นหา หรือการผลิตค่าใกล้เคียงยังได้ถูกปรับให้เป็นแบบสุ่ม (random) เพื่อช่วยเพิ่มโอกาสในการค้นพบคำตอบวงกว้างอีกด้วย อัลกอริทึมของการค้นหาแบบตามูขีงปรับตัวมีรายละเอียดดังต่อไปนี้

- ขั้นตอนที่ 1 กำหนดขอบเขตปริภูมิการค้นหา (search space) รัศมีการค้นหา (R) ตัวนับ (counter) รายการตามู (tabu list, TL) และเกณฑ์ยุติการค้นหา (termination criteria, TC)
- ขั้นตอนที่ 2 สุ่มเลือกคำตอบเริ่มต้น S_0 ภายในปริภูมิการค้นหา โดยกำหนดให้ S_0 มีค่าเป็น *best_neighbor*
- ขั้นตอนที่ 3 ผลิตค่าใกล้เคียง (neighborhood) แบบสุ่มรอบคำตอบเริ่มต้น S_0 ภายในปริภูมิการค้นหา รัศมี R จำนวน N ตัว แล้วเก็บไว้ในเซต $S_1(r)$
- ขั้นตอนที่ 4 ประเมินค่าใกล้เคียงด้วยฟังก์ชันวัตถุประสงค์ (cost function หรือ objective function) โดยกำหนดให้ S_1 คือค่าใกล้เคียงที่ทำให้ค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด และให้ S_1 มีค่าเป็น *best_neighbor1*
- ขั้นตอนที่ 5 ถ้า $S_1 < S_0$ แล้วทำการเก็บ S_0 ไว้ใน TL จากนั้นปรับค่า $S_0 = S_1$ ไม่เช่นนั้น แล้วทำการเก็บ S_1 ไว้ใน TL แทน
- ขั้นตอนที่ 6 เรียกใช้กลไกย้อนรอยการค้นหา เมื่อกระบวนการค้นหาเกิดการล๊อคโดยคำตอบวงแคบเฉพาะถิ่น
- ขั้นตอนที่ 7 ตรวจสอบ TC ถ้า TC สอดคล้อง ให้ยุติกระบวนการค้นหา ค่า S_0 คือคำตอบที่ดีที่สุด ไม่เช่นนั้นให้ทำขั้นตอนต่อไป
- ขั้นตอนที่ 8 เรียกใช้กลไกปรับรัศมีการค้นหา เมื่อกระบวนการค้นหาเข้าใกล้คำตอบ
- ขั้นตอนที่ 9 ปรับเพิ่มค่าตัวนับ และวนกลับไปยังขั้นตอนที่ 3 เพื่อดำเนินกระบวนการค้นหาในรอบต่อไป

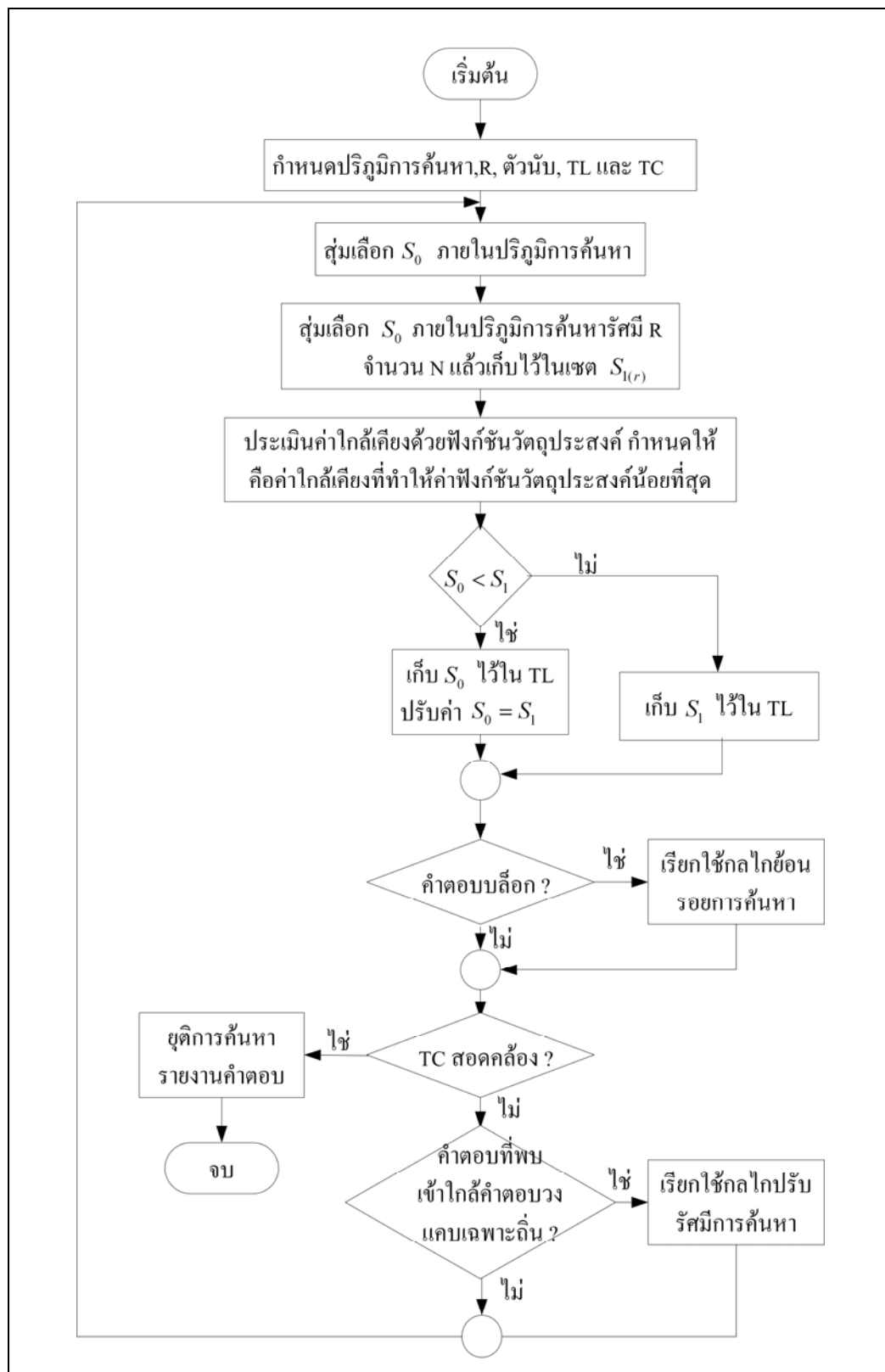
กลไกย้อนรอยการค้นหา เป็นกระบวนการเรียกใช้คำตอบเก่าที่บันทึกไว้ใน TL เมื่อพบว่าคำตอบที่ได้จากการค้นหามีค่าซ้ำกันตามจำนวนครั้งที่กำหนด ซึ่งทำให้เกิดความไม่แน่ใจว่าเป็นคำตอบวงกว้างหรือเป็นการล๊อคโดยคำตอบวงแคบเฉพาะถิ่น เมื่อกระบวนการค้นหาพบคำตอบวงแคบเฉพาะถิ่น และเกิด

การเลือกโดยคำตอบวงแคบเฉพาะถิ่นขึ้น กลไกการย้อนรอยค้นหาจะอนุญาตให้ใช้คำตอบที่ได้มีการค้นหาไว้แล้วซึ่งบรรจุอยู่ใน TL เพื่อใช้เป็นคำตอบเริ่มต้นสำหรับการค้นหาในรอบต่อไป ซึ่งจะทำให้เกิดปฏิกิริมีย่อยของการค้นหาใหม่ เป็นการเปิดโอกาสให้กระบวนการค้นหาเปลี่ยนทิศทาง และสามารถหลุดออกจาก การเลือกโดยคำตอบวงแคบเฉพาะถิ่นได้ จากนั้นกระบวนการค้นหาจะดำเนินต่อไปจนพบกับคำตอบวงแคบเฉพาะถิ่นอื่น ๆ และในที่สุดก็จะพบกับคำตอบวงกว้าง

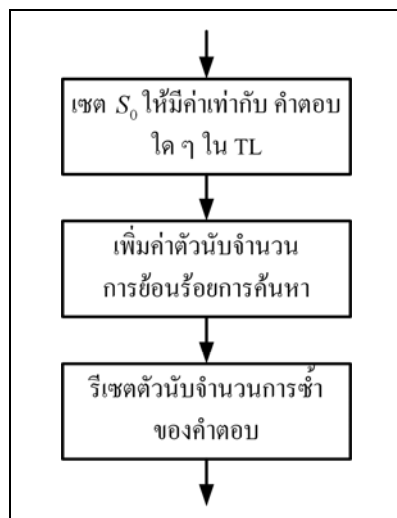
สำหรับกลไกปรับรัศมีการค้นหา เป็นกระบวนการเร่งความเร็วให้กับกระบวนการค้นหา กลไกดังกล่าวจะดำเนินการปรับลดรัศมีการค้นหาลงเมื่อกระบวนการค้นหาพบคำตอบที่ใกล้คำตอบวงแคบเฉพาะถิ่นหรือคำตอบวงกว้าง เมื่อรัศมีการค้นหามีค่าลดลงขณะที่จำนวนค่าใกล้เคียงเท่าเดิมจะทำให้กระบวนการค้นหาสามารถพิจารณาค่าใกล้เคียงได้อย่างละเอียดมากยิ่งขึ้น กล่าวอีกนัยหนึ่งก็คือ เมื่อกระบวนการค้นหาเข้าสู่ใกล้คำตอบ รัศมีการค้นหาที่ยังเล็กลง และค่าใกล้เคียงที่ถูกผลิตขึ้นมาแบบสุ่มก็ จะยังมีความละเอียดมากขึ้น ทำให้กระบวนการค้นหาพบคำตอบได้อย่างรวดเร็ว เงื่อนไขการปรับรัศมีการ ค้นหาอาจกำหนดโดยใช้ค่าฟังก์ชันวัตถุประสงค์ ดังแสดงในความสัมพันธ์ (5-1) เมื่อ ε คือ ค่าฟังก์ชัน วัตถุประสงค์ที่พบ δ_i คือค่าฟังก์ชันวัตถุประสงค์ที่กำหนด DF_i คือ ตัวประกอบการลดรัศมีการค้นหา (decreasing factor) ซึ่ง $DF_i > 1$ และ $i=1, \dots, n$ คือ จำนวนเงื่อนไข ค่า δ DF และ i ขึ้นอยู่กับปัญหาที่สนใจ ในแต่ละกรณีปัญหานั้นไม่จำเป็นจะต้องมีค่าเท่ากัน

$$\text{if } [\varepsilon \leq \delta_i] \text{ then } \left[R = \frac{R}{DF_i} \right] \quad (5-1)$$

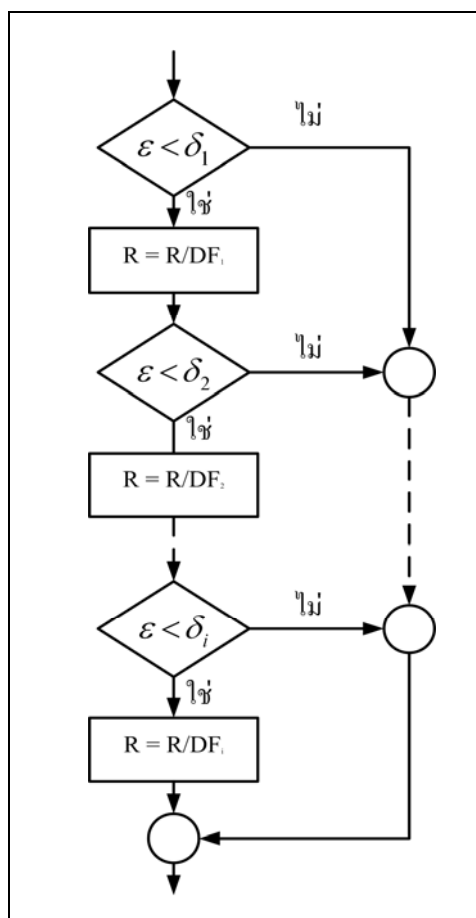
แผนภูมิการดำเนินงานของการค้นหาแบบดาบเชิงปรับตัว แสดงไว้ในรูปที่ 5.1 โดยที่รายละเอียดของกลไกย้อนรอยการค้นหาคำตอบ แสดงด้วย แผนภูมิในรูปที่ 5.2 และรายละเอียดของกลไกปรับรัศมีการ ค้นหาแสดง ด้วยแผนภูมิในรูปที่ 5.3 ตามลำดับ



รูปที่ 5.1 แผนภูมิการดำเนินงานของของการค้นหาแบบตามูเชิงปรับตัว (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004)



รูปที่ 5.2 แผนภูมิการทำงานของกลไกย้อนรอยการค้นหา (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004)



รูปที่ 5.3 แผนภูมิการทำงานของกลไกปรับรมีการค้นหา (Puangdownreong, Kulworawanichpong, and Sujitjorn, 2004)

5.3 การหาค่าเหมาะที่สุดสำหรับตัวชดเชยรีโซแนนซ์

การปรับปรุงสมรรถนะของหัวอ่านฮาร์ดดิสก์ไดรฟ์ที่ผ่านการชดเชยมาแล้วในระดับหนึ่งให้ดียิ่งขึ้น ในหัวข้อนี้เป็นการประยุกต์ ATS เพื่อหาพารามิเตอร์ของตัวชดเชยที่เหมาะสมที่สุด ซึ่งพิจารณาตัวชดเชยมีอันดับสี่แบบหนึ่งวงเกิน แสดงด้วยฟังก์ชันถ่ายโอนดังสมการที่ (5-2)

$$G_{c4}(s) = K \frac{(s+z_1)(s+z_2)(s+z_3)(s+z_4)}{(s+p_1)(s+p_2)(s+p_3)(s+p_4)} \quad (5-2)$$

พารามิเตอร์ $z_1, z_2, z_3, z_4, p_1, p_2, p_3, p_4$ และ K เป็นค่าจริงบวก (positive real) ที่เหมาะสมที่สุด มีพารามิเตอร์การค้นหาแบบตามูเชิงปรับตัว ดังต่อไปนี้

- ขอบเขตปริภูมิการค้นหาของ $z_1, z_2, z_3, z_4, p_1, p_2, p_3, p_4$ และ K เท่ากับ $[0 \ 100000]$
- รัศมีการค้นหา (R) เมื่อเริ่มต้นมีค่าเท่ากับ 10
- จำนวนสมาชิกค่าใกล้เคียง (n) มีค่าเท่ากับ 50

การค้นหาจะยุติลงเมื่อ TC (termination criteria) สอดคล้องกับเงื่อนไขดังนี้

- จำนวนรอบของการค้นหา (maximum search round) มีค่าเท่ากับ 100,000 รอบ หรือ
- ค่า cost มีค่าน้อยกว่า 10 โดย ค่า cost คือค่าความคลาดเคลื่อนของผลตอบสนองทางโดเมน

เวลาของของระบบกรณีมีการชดเชยด้วยตัวชดเชยจากการค้นหาแบบตามูเชิงปรับตัว แสดงรายละเอียดได้ดังรูปที่ 5.4

```
fstep=100000;
t=[0:1/fstep:1];
[fsim t]=step(sys,t);
error=sum(abs(1-fsim))
```

รูปที่ 5.4 การคำนวณค่า cost

รายการตามู (tabu list, TL)

- เงื่อนไขการย้อนรอย (back tracking, BT) เท่ากับ 5
- เงื่อนไขการเรียกกลไกปรับรัศมีการค้นหา มีดังนี้

ถ้าค่า cost น้อยกว่า 20 ให้ปรับรัศมีการค้นหาเป็น 8 (กำหนดเป็นค่าสัมบูรณ์)

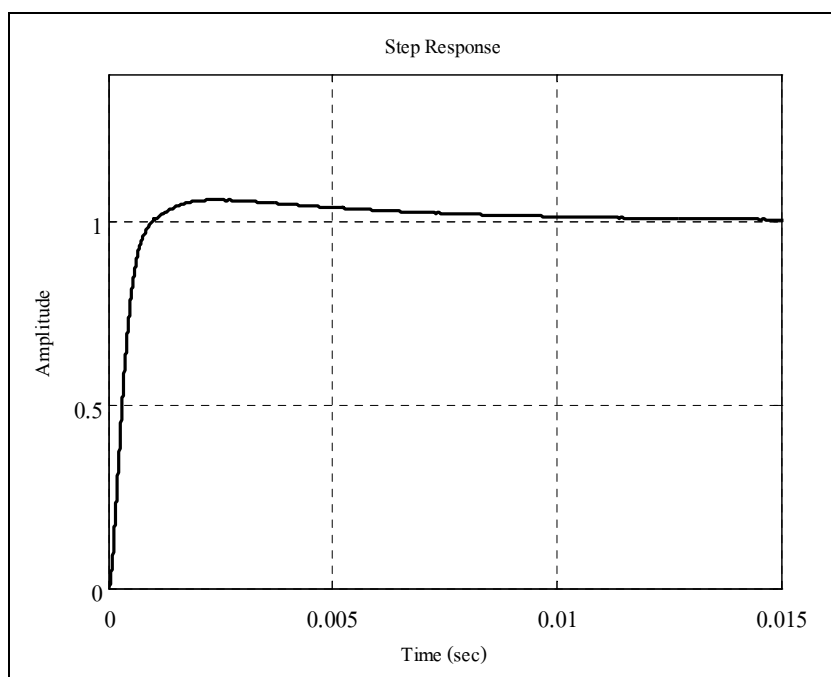
ถ้าค่า cost น้อยกว่า 15 ให้ปรับรัศมีการค้นหาเป็น 6 (กำหนดเป็นค่าสัมบูรณ์)

ถ้าค่า cost น้อยกว่า 12 ให้ปรับรัศมีการค้นหาเป็น 4 (กำหนดเป็นค่าสัมบูรณ์)

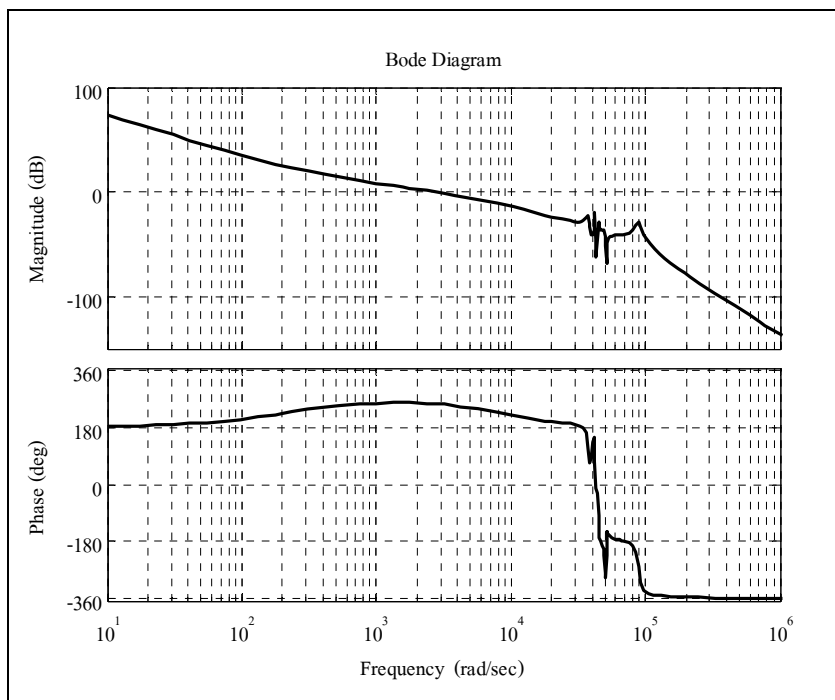
จากผลการดำเนินงานเพื่อหาค่าเหมาะสมที่สุดของตัวชดเชยให้กับระบบการเคลื่อนที่ของหัวอ่านในย่านความถี่สูง ด้วยวิธีการค้นหาแบบตามูเชิงปรับตัว ได้ตัวชดเชยดังที่แสดงในสมการที่ (5-3) ซึ่งการค้นหายุติลงด้วย จำนวนรอบของการค้นหา ค่า cost เท่ากับ 69.84

$$G_{c4}(s) = 0.1 \frac{(s + 203.2)(s + 2727)(s + 9640)(s + 4322)}{(s + 4292)(s + 4348)(s + 8920)(s + 6576)} \quad (5-3)$$

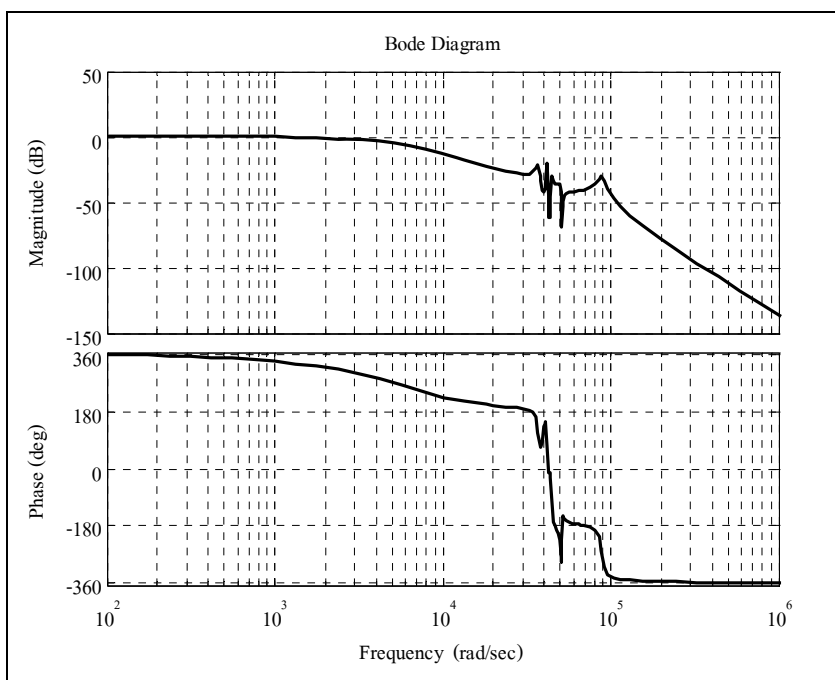
จำลองสถานการณ์ของระบบที่มีการชดเชยด้วยตัวชดเชยในสมการที่ (5-3) เพื่อตรวจสอบสมรรถนะของระบบต่ออินพุตขั้นบันไดหนึ่งหน่วย มีผลตอบสนองทางเวลาดังแสดงในรูปที่ 5.5 แผนภาพโบดของระบบวงเปิดที่มีการชดเชยแล้วแสดงไว้ในรูปที่ 5.6 และแผนภาพการตอบสนองทางความถี่ของระบบวงปิดที่มีการชดเชยแล้วแสดงไว้ในรูปที่ 5.7



รูปที่ 5.5 ผลการตอบสนองทางเวลาของระบบกรณีมีการชดเชยด้วยตัวชดเชยจากการค้นหาแบบตามูเชิงปรับตัว



รูปที่ 5.6 แผนภาพโบคของระบบวงเปิดกรณีมีการชดเชยด้วยตัวชดเชยจากการค้นหาแบบตามเชิงปรับตัว



รูปที่ 5.7 การตอบสนองทางความถี่ของระบบวงปิดกรณีมีการชดเชยด้วยตัวชดเชยจากการค้นหาแบบตามเชิงปรับตัว

ผลการจำลองสถานการณ์ระบบเมื่อใช้ตัวชดเชย G_{c4} ดังสมการที่ (5-3) แสดงไว้ในรูปที่ 5.5 จะเห็นจากผลตอบสนองต่ออินพุตขั้นบันไดหนึ่งหน่วย ว่าระบบวงปิดมีผลตอบสนองที่รวดเร็ว มีการพุ่งเกินเพียง 5.6% เข้าสู่สถานะอยู่ตัวภายใน 7.71 มิลลิวินาที มีค่าผิดพลาดเป็นศูนย์เมื่อเวลาผ่านไป 15 มิลลิวินาที จากรูปที่ 5.6 จะเห็นว่าระบบห้วอ่านฮาร์ดดิสก์ไครฟ์ที่ชดเชยแล้วมี $GM = 27.1\text{ dB}$ และ $PM = 74.7^\circ$ นอกจากนั้นแล้ว รีโซแนนซ์และแอนตี้รีโซแนนซ์ยังปรากฏน้อยลงในระบบที่ใช้ตัวชดเชยดังกล่าว ซึ่งแสดงได้ด้วยแผนภาพการตอบสนองทางความถี่ของระบบวงปิดดังรูปที่ 5.7 และยังเห็นได้ว่าระบบวงปิดมีแบนด์วิดท์ เท่ากับ 3,780 เรเดียนต่อวินาที

5.4 สรุป

เนื้อหาในบทนี้ได้กล่าวถึงการค้นหาแบบตามูเชิงปรับตัว การหาค่าเหมาะที่สุดสำหรับตัวชดเชยย่านความถี่สูง เพื่อสร้างเสถียรภาพให้กับระบบการเคลื่อนที่ของห้วอ่านฮาร์ดดิสก์ไครฟ์ที่ปรากฏสภาวะรีโซแนนซ์และการขาดเสถียรภาพ จากผลการจำลองสถานการณ์เพื่อพิจารณาพลวัตของระบบที่ผ่านการชดเชยด้วยตัวชดเชยที่ได้จากผลการค้นหาค่าพารามิเตอร์ด้วยการค้นหาแบบตามูเชิงปรับตัว เมื่อพิจารณาการตอบสนองของระบบที่มีการชดเชยแล้วพบว่าผลการผลตอบสนองทางเวลาและทางความถี่เป็นไปตามข้อกำหนดสมรรถนะทุกประการ ซึ่งจะเห็นได้ว่าค้นหาแบบตามูเชิงปรับตัวช่วยในการค้นหาค่าเหมาะสมที่สุดของตัวชดเชยได้เป็นอย่างดี

บทที่ 6

6.1 สรุป

งานวิจัยวิทยานิพนธ์นี้ดำเนินการศึกษาวิจัยด้านการออกแบบระบบควบคุม เพื่อสร้างเสถียรภาพ และปรับปรุงสมรรถนะให้กับหัวอ่านฮาร์ดดิสก์ไดรฟ์ เนื่องจากการควบคุมหัวอ่านฮาร์ดดิสก์โดยปกติ ดำเนินงานทั้งในย่านความถี่ต่ำและความถี่สูง การควบคุมในย่านความถี่ต่ำมีประเด็นสำคัญเป็นการ แก้ปัญหาสภาวะไม่เป็นเชิงเส้นของแรงเสียดทาน ซึ่งเกิดขึ้นที่จุดหมุนของแขนหัวอ่านที่มีการเสียดสีกัน ของโลหะ และยังเกิดจากสายเคเบิลข้อมูลที่เชื่อมต่อระหว่างหัวอ่านกับแขนโลหะอีกด้วย ส่วนในย่าน ความถี่สูง การควบคุมให้หัวอ่านติดตามแทร็กข้อมูล ประสบปัญหาสำคัญจากสภาวะรีโซแนนซ์ และการ ขาดเสถียรภาพ การควบคุมในย่านความถี่ต่ำได้ใช้ตัวควบคุมแบบพีซีซี ซึ่งได้ออกแบบกฎการควบคุม ออกเป็นสองชุด กฎการควบคุมชุดแรกเพื่อช่วยปรับปรุงในการตอบสนองในสถานะชั่วครู่ให้ดีขึ้น และกฎ ชุดที่สองซึ่งเป็นกฎแบบพีซีซีเพื่อรักษาเสถียรภาพและกำจัดความผิดพลาดในสถานะอยู่ตัว ในการกำหนด กฎพีซีซีแต่ละกฎประกอบด้วยตัวแปรอินพุตสองตัว เรียกกฎพีซีซีในลักษณะนี้ว่า กฎพีซีซีแบบกฎสอง อินพุต การออกแบบกฎการควบคุมชุดแรกและกฎพีซีซีแต่ละกฎ อาศัยการวิเคราะห์พฤติกรรมของระบบ จากการจำลองสถานการณ์ ดังปรากฏในบทที่ 2 ซึ่งผลการจำลองสถานการณ์แสดงให้เห็นว่า โครงสร้าง ของตัวควบคุมที่ได้ออกแบบนั้น สามารถสร้างเสถียรภาพและปรับปรุงสมรรถนะของหัวอ่านฮาร์ดดิสก์ ไดรฟ์ได้ภายในระยะเวลาที่สั้นกว่าตัวควบคุมแบบพีซีซีอย่างเดียว สำหรับการควบคุมในย่านความถี่สูงได้ ออกแบบตัวชดเชยด้วยวิธีการต่างๆ ได้แก่ ตัวชดเชยแบบเฟสล้ำหน้า ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลัง เชิงซ้อน ตัวชดเชยแบบเฟสล้ำหน้าและล้ำหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$ และการอพ ดิไมซ์ค่าพารามิเตอร์ของตัวชดเชยด้วยค้นหาแบบตามูเชิงปรับตัว หลังจากที่ได้ออกแบบตัวชดเชยด้วย วิธีการต่าง ๆ ดังที่ได้กล่าวมาแล้วนั้น ได้ดำเนินการจำลองสถานการณ์เพื่อวิเคราะห์ผลตอบสนองทางเวลา และทางความถี่ของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไดรฟ์ พบว่าตัวชดเชยที่ได้จากการอพดิไมซ์ ค่าพารามิเตอร์ของตัวชดเชยด้วยค้นหาแบบตามูเชิงปรับตัวนั้น ให้ผลการตอบสนองทั้งทางโดเมนเวลา และความถี่เป็นไปตามข้อกำหนดสมรรถนะเพียงตัวเดียว

ในบทที่ 1 ของวิทยานิพนธ์ได้กล่าวถึงความสำคัญของปัญหา คือ การเพิ่มอัตราการอ่านข้อมูลของ ฮาร์ดดิสก์ไดรฟ์ให้เร็ว โดยการลดค่าความผิดพลาดของการควบคุมตำแหน่งหัวอ่านฮาร์ดดิสก์ไดรฟ์เมื่อ เคลื่อนตัวไปตามแทร็ก (track) ต่าง ๆ เพื่ออ่านข้อมูลโดยความผิดพลาดดังกล่าวมีสาเหตุมาจากสภาพไม่ เป็นเชิงเส้นและความเสียดทานของตัวกระตุ้นชนิด voice-coil-motor (VCM) ซึ่งทำหน้าที่เป็นตัวยับเคลื่อนหัวอ่านฮาร์ดดิสก์ไดรฟ์ ส่วนในบทที่ 2 ของวิทยานิพนธ์ได้กล่าวถึง แบบจำลองของหัวอ่านฮาร์ดดิสก์ไดรฟ์ ซึ่งแบ่งออกเป็นสองย่านการทำงาน คือ ย่านความถี่ต่ำ และย่านความถี่สูง วิทยานิพนธ์บทที่ 3 ได้กล่าวถึง วิธีการควบคุมแบบดั้งเดิม ยังมีข้อจำกัดในการควบคุมระบบที่มีความซับซ้อน ความไม่

แน่นอน ความไม่เป็นเชิงเส้น และไม่เสถียร ดังนั้นจึงอาศัยวิธีการควบคุมแบบชาญฉลาดเข้าช่วยเพื่อแก้ปัญหาต่างๆ เหล่านี้ โดยได้กล่าวถึงวิธีการควบคุมฟัซซี ในการสร้างเสถียรภาพให้กับระบบการเคลื่อนที่ของหัวอ่านในย่านความถี่ต่ำซึ่งประสบปัญหาจากความไม่เป็นเชิงเส้น และได้อธิบายถึงการดำเนินงานออกแบบตัวควบคุมฟัซซี เพื่อให้สามารถสร้างเสถียรภาพให้ระบบการเคลื่อนที่ของหัวอ่านได้ภายในระยะเวลาอันสั้น ส่วนในบทที่ 4 ของวิทยานิพนธ์ ได้กล่าวถึง วิธีการออกแบบตัวชดเชยในย่านความถี่สูงซึ่งประสบปัญหาจากสภาวะรีโซแนนซ์ และการขาดเสถียรภาพ 3 วิธีการ ได้แก่ ตัวชดเชยแบบเฟสล้ำหน้า ตัวชดเชยแบบเฟสล้ำหน้าและล่าหลังเชิงซ้อน ตัวชดเชยแบบเฟสล้ำหน้าและล่าหลังเชิงซ้อนตามวิธีของ Messner เมื่อใช้ค่า $\zeta > 1$ ซึ่งจากการจำลองสถานการณ์แสดงให้เห็นว่าตัวชดเชยทั้ง 3 ให้ผลการตอบสนองไม่เป็นไปตามข้อกำหนดสมรรถนะ ดังนั้นจึงได้ใช้เทคนิคการค้นหาทางปัญญาประดิษฐ์ โดยได้นำการค้นหาแบบตามเชิงปรับตัวมาค้นหาค่าพารามิเตอร์ของตัวชดเชย ซึ่งได้อธิบายไว้ในบทที่ 5 ของวิทยานิพนธ์ ซึ่งตัวชดเชยที่ได้จากการค้นหาแบบตามเชิงปรับตัวนั้นสามารถชดเชยให้กับระบบทำให้ระบบการเคลื่อนที่ของหัวอ่านเป็นไปตามข้อกำหนดสมรรถนะทั้งในโดเมนเวลาและความถี่

6.2 ข้อเสนอแนะ

1) จากการออกแบบตัวชดเชยด้วยวิธีการเฟสล้ำหน้าและล่าหลังเชิงซ้อน ต้องมีการกำหนดค่า ω_m และ ζ เพื่อใช้ในการออกแบบ ซึ่งการกำหนดค่าดังกล่าวจะไม่มีหลักเกณฑ์ตายตัว ดังนั้นในการออกแบบตัวชดเชยระบบ อาจนำวิธีการค้นหารูปแบบต่าง ๆ เข้าช่วยในการค้นหาค่าดังกล่าวที่เหมาะสมที่สุดสำหรับระบบที่สนใจ

2) แบบจำลองคณิตศาสตร์ที่ใช้แทนพลวัตของระบบการเคลื่อนที่ของหัวอ่านในย่านความถี่สูง มีอันดับ 12 ซึ่งอันดับค่อนข้างสูง ดังนั้นการออกแบบตัวชดเชยจึงมีความยุ่งยากซับซ้อน และอาจต้องใช้ตัวชดเชยอันดับสูงเพื่อกำจัดรีโซแนนซ์ ดังนั้นแนวทางหนึ่งอาจใช้ใช้วิธีการทางปัญญาประดิษฐ์ในการค้นหาอันดับของตัวชดเชย เพื่อให้ผลตอบสนองที่ดียิ่งขึ้น

รายการอ้างอิง

- กองพัน อารีรักษ์. (2544). การระบุเอกลักษณ์ไม่เป็นเชิงเส้นและการตรวจสอบเสถียรภาพของระบบสอง
มวลความเฉื่อยที่ปรากฏการก้ำก๋อเชิงกล. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต,
มหาวิทยาลัยเทคโนโลยีสุรนารี, นครราชสีมา.
- เดชา พวงดาวเรือง. (2547). การระบุเอกลักษณ์ของระบบพลวัตด้วยการประมวลผลภาพ. วิทยานิพนธ์
ปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต. มหาวิทยาลัยเทคโนโลยีสุรนารี. นครราชสีมา.
- สราวุฒิ สุจิตจร. (2538). พืชเชื่อมจิกกับระบบควบคุม. วารสารเทคโนโลยีสุรนารี 2(3): 171-196.
- สราวุฒิ สุจิตจร. (2546) การควบคุมอัตโนมัติ. เพียร์สัน เอ็ดดูเคชั่น อินโดไชน่า จำกัด.
- Chen, B.M., Lee, T.H., Peng, K., and Venkataramanan, V. (2006) **Hard Disk Drive Servo Systems**.
Springer-Verlag, London Limited.
- Mamdani, E. H., and Assilian, S. (1975). **An experiment in linguistic synthesis with a fuzzy logic
controller**. Int. J. Man-Mach. Stud.: 7: 1-13.
- Glover, F. (1986). **Future paths for integer programming and links to artificial intelligence**.
Computers and Operations Research. 13: 533-549.
- Hagras, H., and Sobh, T. (2002). **Intelligent learning and control of autonomous robotic agents
operating in unstructured environments**. Information Sciences. 145(1-2): 1-12.
- Iokibe, T., Tobi, T., and Araki, S. (1999). **Intelligent control of air conditioning systems**. In L. C. Jain
and C. W. de Silva (eds). Intelligent adaptive control industrial applications (pp. 219-246). Boca
Raton: CRC Press LLC.
- Peng, K., Chen, B. M., Cheng G., and Lee, T. H. (2005). **Modeling and Compensation of
Nonlinearities and Friction in a Micro Hard Disk Drive Servo System With Nonlinear
Feedback Control**. IEEE Transactions on Control Systems Technology.: 13(5): 708-721.
- King, R. E. (1999). **Computational intelligence in control engineering**. New York: Marcel Dekker.
- Zadeh, L. A. (1968). **Fuzzy Algorithms. Information and Control**.: 12: 94-102.
- Zadeh, L. A. (1973). **Outline of a New Approach to the Analysis of Complex System and Decision
Processes**. IEEE Transaction on System, Man, and Cybernetics.: 3(1): 28-44.
- Zadeh, L. A. (1976). **A Fuzzy-Algorithm Approach to the Definition of Complex or imprecise
Concepts**. Int. J. Man-Machine Studies.: 8: 249-291.

- Meireles, M. R. G., Almeida, P. E. M., and Simoes, M. G. (2003). **A comprehensive review for industrial applicability of artificial neural network**. IEEE Trans. Industrial Electronics. 50(3): 585-601.
- Puangdownreong, D., Kulworawanichpong, T., and Sujitjorn, S. (2004). **Finite convergence and performance evaluation of adaptive tabu search**. Lecture Notes in Computer Science. Springer-Verlag Heidelberg. (online)
- Sujitjorn, S., Mellitt, B., and Rabukwella, N. B. (1987). **Energy minimisation in transit systems using a train based expert system for dynamic coast control**. Computer in railway operations. Springer-Verlag: 303-316.
- Mohagheghi, S., Venayagamoorthy, G. K., and Harley, R. G. (2005). **An Adaptive Mamdani Fuzzy Logic Based Controller for a Static Compensator in a Multimachine Power System**. ISAP.: 228-233.
- Takagi T., and Sugeno, M. (1985). **Fuzzy identification of system and its application to modeling and control**. IEEE Trans. on SMC.: 15: 116-132.
- Messner, W. C., Bedillion, M. D., Xia, L., and Karns, D. C. (2007). **Lead and Lag Compensators with Complex Poles and Zeros**. IEEE Control System Magazine.: 44-54.

ภาคผนวก ก

รายละเอียดโปรแกรมจำลองการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์

โปรแกรมจำลองการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์

โปรแกรมสำหรับแบบจำลองของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์โดยใช้โปรแกรม MATLAB™ แบ่งออกเป็น 2 ส่วนการทำงาน คือ ในย่านความถี่ต่ำและ ย่านความถี่สูง

โปรแกรมสำหรับแบบจำลองของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ย่านความถี่ต่ำ
โดย วิไลพร เงินบาท (สิงหาคม 2550) สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

Program: VCM_low_frequency.m

```
clear all;  
clc;
```

```
T=0.0001;  
x1=0;  
x2=0;  
Tf=0;  
x1n=[];  
x2n=[];  
i=[];  
un=[];  
u0=0;  
x10=0;  
r=1;  
a=0.015;  
error_1=0;  
e=1;  
errorn=[];
```

```
for n=1:0.02/T;
```

```
    R=n*T;  
    Rh=n*T;  
    u=a;  
    e=e+1;
```

```
    f11=x2;  
    f21=2.35*10^8*u-6.78445*10^6*atan(0.5886*x1)+Tf;  
    x1h=x1+0.5*T*f11;  
    x2h=x1+0.5*T*f21;  
    Rh=(n+0.5)*T;
```

```

uh=a;
f12=x2h;
f22=2.35*10^8*uh-6.78445*10^6*atan(0.5886*x1h)+Tf;
x1h=x1+0.5*T*f12;
x2h=x1+0.5*T*f22;
Rh=(n+0.5)*T;
uh=a;
f13=x2h;
f23=2.35*10^8*uh-6.78445*10^6*atan(0.5886*x1h)+Tf;
f23=-2*x1h-2*x2h+2*uh;
x1h=x1+0.5*T*f13;
x2h=x1+0.5*T*f23;
Rh=(n+1)*T;
uh=a;

f14=x2h;
f24=2.35*10^8*uh-6.78445*10^6*atan(0.5886*x1h)+Tf;
x1=x1+(1/6)*T*(f11+2*f12+2*f13+f14);
x1n(n,1)=x1;
i(n,1)=R;
x2=x2+(1/6)*T*(f21+2*f22+2*f23+f24);
x2n(n,1)=x2;
un(n,1)=u;

Te=2.35*10^8*(-0.02887*atan(0.5886*x1)+u);
Ts=1.293*10^6*abs(u0*x10)+1.65*10^4;
Te1=abs(Te);

if Te>0;
    Te2=1;
elseif Te<0;
    Te2=-1;
else
    Te2=0;
end

if x2>0;
    x22=1;
elseif x2<0;
    x22=-1;
else
    x22=0;
end

if x2==0
    if Te1<=Ts
        Tf=-Te;
    elseif Te1>Ts
        Tf=-Ts*Te2;
    end
    else
        Tf=-(abs(1.175*10^6*u*x1+0.01*(x2^2))+15000)*x22-282.6*x2;
    end
end

figure(1)

```

```

plot(i,x1n)
grid on
hold on
title('Comparison of time-domain responses of the VCM actuator')
xlabel('Time (seconds)')
ylabel('Displacement(micrometers)')
figure(2)
plot(i,un)
title('input of the VCM actuator')
xlabel('Time (seconds)')
ylabel('Input signal to VCM (volts)')
grid on
hold on

```

โปรแกรมสำหรับแบบจำลองของระบบการเคลื่อนที่ของหัวอ่านฮาร์ดดิสก์ไครฟ์ย่านความถี่สูง

Program: VCM_high_frequency.m

```

clc;
clear all;

z1=[0.8709 1726 1.369e9];
p1=[1 1480 1.369e9];
G1=tf(z1,p1);

z2=[0.9332 -805.8 1.739e9];
p2=[1 125.1 1.739e9];
G2=tf(z2,p2);
z3=[1.072 925.1 1.997e9];
p3=[1 536.2 1.997e9];
G3=tf(z3,p3);

z4=[0.9594 98.22 2.514e9];
p4=[1 1805 2.514e9];
G4=tf(z4,p4);

z5=[0 0 7.877e9];
p5=[1 6212 7.877e9];
G5=tf(z5,p5);

z6=[0 0 2.35e8];
p6=[1 0 0];
G6=tf(z6,p6);

G=G1*G2*G3*G4*G5*G6;
Gp=feedback(G,1)

```

```

figure(1)
step(Gp)
grid on
figure(2)
bode(G)
grid on

```

ภาคผนวก ข

รายละเอียดโปรแกรมตัวควบคุมพืชจีแบบกฏสองอินพุต

โปรแกรมควบคุมฟัซซีแบบกฎสองอินพุต

โปรแกรมสำหรับควบคุมฟัซซีแบบกฎสองอินพุต ในงานวิจัยวิทยานิพนธ์นี้ ได้ใช้โปรแกรมที่พัฒนาขึ้นโดยอาศัยโปรแกรม MATLAB™ ดังมีรายละเอียดต่อไปนี้

โปรแกรมสำหรับควบคุมฟัซซีแบบกฎสองอินพุต

โดย วิไลพร เงินบาท (มิถุนายน 2551) สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรม fuzzy.m

```
function [control]=fuzzy(error,change_error,ee,gg,hh,ij)
syms NL NS ZE PS PL ZD XLD LD MD SD NC SI MI LI XLI
dx=0.00001;
xmin_error=ee;
xmax_error=gg;
xmin_change_error=hh;
xmax_change_error=ij;
```

```
Rule1=1;
Rule2=0.75;
Rule3=0.5;
Rule4=0.25;
Rule5=0;
Rule6=0.75;
Rule7=0.5;
Rule8=0.25;
Rule9=0;
Rule10=-0.25;
Rule11=0.5;
Rule12=0.25;
Rule13=0;
Rule14=-0.25;
Rule15=-0.5;
Rule16=0.25;
Rule17=0;
Rule18=-0.25;
Rule19=-0.5;
Rule20=-0.75;
Rule21=0;
Rule22=-0.25;
Rule23=-0.5;
Rule24=-0.75;
Rule25=-1;
```

```

%%%%membership error%%%%
%%%%Line1
    x1_error=[-1 xmin_error];
    y1_error=[1 0];
    p1_error=polyfit(x1_error,y1_error,1);
    x1n_error=[-1:dx:xmin_error];
    yy1_error=polyval(p1_error,x1n_error);
%%%% Line2
    x2_error=[-1 xmin_error];
    y2_error=[0 1];
    p2_error=polyfit(x2_error,y2_error,1);
    x2n_error=[-1:dx:xmin_error];
    yy2_error=polyval(p2_error,x2n_error);
%%%% Line3
    x3_error=[xmin_error 0];
    y3_error=[1 0];
    p3_error=polyfit(x3_error,y3_error,1);
    x3n_error=[xmin_error:dx:0];
    yy3_error=polyval(p3_error,x3n_error);
%%%% Line4
    x4_error=[xmin_error 0];
    y4_error=[0 1];
    p4_error=polyfit(x4_error,y4_error,1);
    x4n_error=[xmin_error:dx:0];
    yy4_error=polyval(p4_error,x4n_error);
%%%% Line5
    x5_error=[0 xmax_error];
    y5_error=[1 0];
    p5_error=polyfit(x5_error,y5_error,1);
    x5n_error=[0:dx:xmax_error];
    yy5_error=polyval(p5_error,x5n_error);
%%%%Line6
    x6_error=[0 xmax_error];
    y6_error=[0 1];
    p6_error=polyfit(x6_error,y6_error,1);
    x6n_error=[0:dx:xmax_error];
    yy6_error=polyval(p6_error,x6n_error);
%%%%Line7
    x7_error=[xmax_error 1];
    y7_error=[1 0];
    p7_error=polyfit(x7_error,y7_error,1);
    x7n_error=[xmax_error:dx:1];
    yy7_error=polyval(p7_error,x7n_error);
%%%%Line8
    x8_error=[xmax_error 1];
    y8_error=[0 1];
    p8_error=polyfit(x8_error,y8_error,1);
    x8n_error=[xmax_error:dx:1];
    yy8_error=polyval(p8_error,x8n_error);

%%%%membership change in error%%%%
    x1_change_error=[-1 xmin_change_error];
    y1_change_error=[1 0];
    p1_change_error=polyfit(x1_change_error,y1_change_error,1);
    x1n_change_error=[-1:dx:xmin_change_error];
    yy1_change_error=polyval(p1_change_error,x1n_change_error);

%%%% Line2

```

```

x2_change_error=[-1 xmin_change_error];
y2_change_error=[0 1];
p2_change_error=polyfit(x2_change_error,y2_change_error,1);
x2n_change_error=[-1:dx:xmin_change_error];
yy2_change_error=polyval(p2_change_error,x2n_change_error);
%% Line3
x3_change_error=[xmin_change_error 0];
y3_change_error=[1 0];
p3_change_error=polyfit(x3_change_error,y3_change_error,1);
x3n_change_error=[xmin_change_error:dx:0];
yy3_change_error=polyval(p3_change_error,x3n_change_error);
%% Line4
x4_change_error=[xmin_change_error 0];
y4_change_error=[0 1];
p4_change_error=polyfit(x4_change_error,y4_change_error,1);
x4n_change_error=[xmin_change_error:dx:0];
yy4_change_error=polyval(p4_change_error,x4n_change_error);
%% Line5
x5_change_error=[0 xmax_change_error];
y5_change_error=[1 0];
p5_change_error=polyfit(x5_change_error,y5_change_error,1);
x5n_change_error=[0:dx:xmax_change_error];
yy5_change_error=polyval(p5_change_error,x5n_change_error);
%% Line6
x6_change_error=[0 xmax_change_error];
y6_change_error=[0 1];
p6_change_error=polyfit(x6_change_error,y6_change_error,1);
x6n_change_error=[0:dx:xmax_change_error];
yy6_change_error=polyval(p6_change_error,x6n_change_error);
%% Line7
x7_change_error=[xmax_change_error 1];
y7_change_error=[1 0];
p7_change_error=polyfit(x7_change_error,y7_change_error,1);
x7n_change_error=[xmax_change_error:dx:1];
yy7_change_error=polyval(p7_change_error,x7n_change_error);
%% Line8
x8_change_error=[xmax_change_error 1];
y8_change_error=[0 1];
p8_change_error=polyfit(x8_change_error,y8_change_error,1);
x8n_change_error=[xmax_change_error:dx:1];
yy8_change_error=polyval(p8_change_error,x8n_change_error);

%% check rule error
if error < xmin_error
    xx1=NL;
    if (error >= -1)& (error <= 0)
        xx2=NS;
    elseif (error >= xmin_error)& (error <= xmax_error)
        xx2=ZE;
    elseif (error > 0)& (error <= 1)
        xx2=PS;
    elseif error > xmax_error
        xx2=PL;
    else

```

```

        xx2=ZD;
    end
elseif (error >= -1)& (error <=0)
    xx1=NS;
    if error < xmin_error
        xx2=NL;
    elseif (error >= xmin_error)& (error <=xmax_error)
        xx2=ZE;
    elseif (error > 0)& (error <= 1)
        xx2=PS;
    elseif error > xmax_error
        xx2=PL;
    else
        xx2=ZD;
    end

elseif (error >= xmin_error)& (error <=xmax_error)
    xx1=ZE;
    if (error >= -1)& (error <=0)
        xx2=NS;
    elseif error < xmin_error
        xx2=ZE;
    elseif (error > 0)& (error <= 1)
        xx2=PS;
    elseif error > xmax_error
        xx2=PL;
    else
        xx2=ZD;
    end

elseif (error > 0)& (error <= 1)
    xx1=PS;
    if (error >= -1)& (error <=0)
        xx2=NS;
    elseif (error >= xmin_error)& (error <=xmax_error)
        xx2=ZE;
    elseif error < xmin_error
        xx2=PS;
    elseif error > xmax_error
        xx2=PL;
    else
        xx2=ZD;
    end

elseif error > xmax_error
    xx1=PL;
    if (error >=-1)& (error <=0)
        xx2=NS;
    elseif (error >= xmin_error)& (error <=xmax_error)
        xx2=ZE;
    elseif (error > 0)& (error <=1)
        xx2=PS;
    elseif error < xmin_error
        xx2=PL;
    else

```

```

        xx2=ZD;
    end
end

%%%check rule change in error%%%
if change_error < xmin_change_error
    xx3=NL;
    if (change_error >=-1)& (change_error <=0)
        xx4=NS;
    elseif (change_error >= xmin_change_error)& (change_error <= xmax_change_error)
        xx4=ZE;
    elseif (change_error > 0)& (change_error <= 1)
        xx4=PS;
    elseif change_error > xmax_change_error
        xx4=PL;
    else
        xx4=ZD;
    end

elseif (change_error >= -1)& (change_error <=0)
    xx3=NS;
    if change_error < xmin_change_error
        xx4=NL;
    elseif (change_error >= xmin_change_error)& (change_error <= xmax_change_error)
        xx4=ZE;
    elseif (change_error > 0)& (change_error <= 1)
        xx4=PS;
    elseif change_error > xmax_change_error
        xx4=PL;
    else
        xx4=ZD;
    end

elseif (change_error >= xmin_change_error)& (change_error <= xmax_change_error)
    xx3=ZE;
    if (change_error >= -1)& (change_error <= 0)
        xx4=NS;
    elseif change_error < xmin_change_error
        xx4=ZE;
    elseif (change_error > 0)& (change_error <= 1)
        xx4=PS;
    elseif change_error > xmax_change_error
        xx4=PL;
    else
        xx4=ZD;
    end

elseif (change_error > 0)& (change_error <= 1)
    xx3=PS;
    if (change_error >= -1)& (change_error <= 0)
        xx4=NS;
    elseif (change_error >= xmin_change_error)& (change_error <= xmax_change_error)
        xx4=ZE;
    elseif change_error < xmin_change_error
        xx4=PS;
    elseif change_error > xmax_change_error

```

```

        xx4=PL;
    else
        xx4=ZD;
    end

elseif change_error > xmax_change_error
    xx3=PL ;
    if (change_error >= -1)& (change_error <= 0)
        xx4=NS;
    elseif (change_error >= xmin_change_error)& (change_error <= xmax_change_error)
        xx4=ZE;
    elseif (change_error > 0)& (change_error <= 1)
        xx4=PS;
    elseif change_error < xmin_change_error
        xx4=PL;
    else
        xx4=ZD;
    end
end

%%%%%%%%%defuzzification %%%%%%%%%%
%%%%%%%%%xx1,xx3%%%%%%%%%

if xx1==NL
    if error<=-1
        yy11_error=1;
    elseif (error>-1) & (error<=xmin_error)
        yy11_error=polyval(p1_error,error);
    end
    if xx3==NL %rule 1% if error is NL and change in error is NL then output is XLI
        if change_error<=-1
            yy11_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p1_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule1*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule1*yy11_change_error;
            hh11 = yy11_change_error;
        end
    end

elseif xx3==NS %rule 2% if error is NL and change in error is NS then output is LI

        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy11_change_error=polyval(p3_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule2*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule2*yy11_change_error;
            hh11 = yy11_change_error;
        end
    end
end

```

```

end

elseif xx3==ZE %rule 3% if error is NL and change in error is ZE then output is MI
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy11_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p5_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule3*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule3*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PS %rule 4% if error is NL and change in error is PS then output is SI
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy11_change_error=polyval(p7_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule4*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule4*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PL %rule 5% if error is NL and change in error is PL then output is NC
    if (change_error>=xmax_change_error) & (change_error<1)
        yy11_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy11_change_error=1;
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule5*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule5*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZD
    yy11_change_error=0;
    yy11_error=0;
    yy11 = 0;
    hh11 = 0;
end

end

if xx1==NS
    if (error>=-1) & (error<=xmin_error)
        yy11_error=polyval(p2_error,error);
    elseif (error>xmin_error) & (error<=0)

```



```

        yy11_error=polyval(p3_error,error);
    end
    if xx3==NL %rule 6% if error is NS and change in error is NL then output is LI
        if change_error<=-1
            yy11_change_error=1;
        elseif (change_error>=-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p1_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule6*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule6*yy11_change_error;
            hh11 = yy11_change_error;
        end
    elseif xx3==NS %rule 7% if error is NS and change in error is NS then output is MI

        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy11_change_error=polyval(p3_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule7*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule7*yy11_change_error;
            hh11 = yy11_change_error;
        end
    elseif xx3==ZE %rule 8% if error is NS and change in error is ZE then output is SI
        if (change_error>=xmin_change_error) & (change_error<0)
            yy11_change_error=polyval(p4_change_error,change_error);
        elseif (change_error>=0) & (change_error<=xmax_change_error)
            yy11_change_error=polyval(p5_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule8*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule8*yy11_change_error;
            hh11 = yy11_change_error;
        end
    elseif xx3==PS %rule 9% if error is NS and change in error is PS then output is NC
        if (change_error>=0) & (change_error<=xmax_change_error)
            yy11_change_error=polyval(p6_change_error,change_error);
        elseif (change_error>xmax_change_error) & (change_error<=1)
            yy11_change_error=polyval(p7_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule9*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule9*yy11_change_error;
            hh11 = yy11_change_error;
        end
    end

```

```

        end

elseif xx3==PL %rule 10% if error is NS and change in error is PL then output is SD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy11_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy11_change_error=1;
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule10*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule10*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZD
    yy11_change_error=0;
    yy11_error=0;
    yy11 = 0;
    hh11 = 0;
end
end

if xx1==ZE
    if (error>=xmin_error) & (error<=0)
        yy11_error=polyval(p4_error,error);
    elseif (error>0) & (error<=xmax_error)
        yy11_error=polyval(p5_error,error);
    end

    if xx3==NL %rule 11% if error is ZE and change in error is NL then output is MI
        if change_error<=-1
            yy11_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p1_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule11*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule11*yy11_change_error;
            hh11 = yy11_change_error;
        end

    elseif xx3==NS %rule 12% if error is ZE and change in error is NS then output is SI
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy11_change_error=polyval(p3_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule12*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule12*yy11_change_error;

```

```

        hh11 = yy11_change_error;
    end

elseif xx3==ZE %rule 13% if error is ZE and change in error is ZE then output is NC
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy11_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p5_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule13*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule13*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PS %rule 14% if error is ZE and change in error is PS then output is SD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy11_change_error=polyval(p7_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule14*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule14*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PL %rule 15% if error is ZE and change in error is PL then output is MD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy11_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy11_change_error=1;
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule15*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule15*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZD
    yy11_change_error=0;
    yy11_error=0;
    yy11 = 0;
    hh11 = 0;
end

end

if xx1==PS
    if (error>=0) & (error<=xmax_error)
        yy11_error=polyval(p6_error,error);

```

```

elseif (error>xmax_error) & (error<=1)
    yy11_error=polyval(p7_error,error);
end

if xx3==NL %rule 16% if error is PS and change in error is NL then output is SI
    if change_error<=-1
        yy11_change_error=1;
    elseif (change_error>-1) & (change_error<=xmin_change_error)
        yy11_change_error=polyval(p1_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule16*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule16*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==NS %rule 17% if error is PS and change in error is NS then output is NC
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy11_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy11_change_error=polyval(p3_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule17*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule17*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZE %rule 18% if error is PS and change in error is ZE then output is SD
    if (change_error>=xmin_change_error) & (change_error<0)
        yy11_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p5_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule18*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule18*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PS %rule 19% if error is PS and change in error is PS then output is MD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy11_change_error=polyval(p7_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule19*yy11_error;
        hh11 = yy11_error;
    else

```

```

        yy11 = Rule19*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PL %rule 20% if error is PS and change in error is PL then output is LD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy11_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy11_change_error=1;
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule20*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule20*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZD
    yy11_change_error=0;
    yy11_error=0;
    yy11 = 0;
    hh11 = 0;
end

end

if xx1==PL
    if (error>=xmax_error) & (error<1)
        yy11_error=polyval(p8,error);
    elseif error>=1
        yy11_error=1;
    end

    if xx3==NL %rule 21% if error is PL and change in error is NL then output is NC
        if change_error<=-1
            yy11_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy11_change_error=polyval(p1_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule21*yy11_error;
            hh11 = yy11_error;
        else
            yy11 = Rule21*yy11_change_error;
            hh11 = yy11_change_error;
        end

    elseif xx3==NS %rule 22% if error is PL and change in error is NS then output is SD
        if (change_error>=-1)& (change_error<xmin_change_error)
            yy11_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>=xmin_change_error) & (change_error<=0)
            yy11_change_error=polyval(p3_change_error,change_error);
        end
        if yy11_error <= yy11_change_error
            yy11 = Rule22*yy11_error;
            hh11 = yy11_error;
        end
    end
end

```

```

else
    yy11 = Rule22*yy11_change_error;
    hh11 = yy11_change_error;
end

elseif xx3==ZE %rule 23% if error is PL and change in error is ZE then output is MD
    if (change_error>=xmin_change_error) & (change_error<0)
        yy11_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p5_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule23*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule23*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PS %rule 24% if error is PL and change in error is PS then output is LD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy11_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy11_change_error=polyval(p7_change_error,change_error);
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule24*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule24*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==PL %rule 25% if error is PL and change in error is PL then output is XLD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy11_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy11_change_error=1;
    end
    if yy11_error <= yy11_change_error
        yy11 = Rule25*yy11_error;
        hh11 = yy11_error;
    else
        yy11 = Rule25*yy11_change_error;
        hh11 = yy11_change_error;
    end

elseif xx3==ZD
    yy11_change_error=0;
    yy11_error=0;
    yy11 = 0;
    hh11 = 0;
end

end

if xx1==ZD

```

```

yy11_change_error=0;
yy11_error=0;
yy11 = 0;
hh11 = 0;

end

%%%%defuzzification %%%%%%%%%%
%%%%%%%%xx1,xx4%%%%%%%%%

if xx1==NL
    if error<=-1
        yy22_error=1;
    elseif (error>-1) & (error<=xmin_error)
        yy22_error=polyval(p1_error,error);
    end

    if xx4==NL %rule 1% if error is NL and change in error is NL then output is XLI
        if change_error<=-1
            yy22_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p1_change_error,change_error);
        end
        if yy22_error <= yy11_change_error
            yy22 = Rule1*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule1*yy22_change_error;
            hh22 = yy22_change_error;
        end

    elseif xx4==NS %rule 2% if error is NL and change in error is NS then output is LI
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy22_change_error=polyval(p3_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule2*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule2*yy22_change_error;
            hh22 = yy22_change_error;
        end

    elseif xx4==ZE %rule 3% if error is NL and change in error is ZE then output is MI
        if (change_error>=xmin_change_error) & (change_error<=0)
            yy22_change_error=polyval(p4_change_error,change_error);
        elseif (change_error>0) & (change_error<=xmax_change_error)
            yy22_change_error=polyval(p5_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule3*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule3*yy22_change_error;

```

```

        hh22 = yy22_change_error;
    end

elseif xx4==PS %rule 4% if error is NL and change in error is PS then output is SI
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p7_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule4*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule4*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PL %rule 5% if error is NL and change in error is PL then output is NC
    if (change_error>=xmax_change_error) & (change_error<1)
        yy22_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy22_change_error=1;
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule5*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule5*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end

end

if xx1==NS
    if (error>=-1) & (error<=xmin_error)
        yy22_error=polyval(p2_error,error);
    elseif (error>xmin_error) & (error<=0)
        yy22_error=polyval(p3_error,error);
    end

    if xx4==NL %rule 6% if error is NS and change in error is NL then output is LI
        if change_error<=-1
            yy22_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p1_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule6*yy22_error;
            hh22 = yy22_error;
        else

```



```

        yy22 = Rule6*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==NS %rule 7% if error is NS and change in error is NS then output is MI
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy22_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p3_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule7*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule7*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZE %rule 8% if error is NS and change in error is ZE then output is SI
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p5_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule8*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule8*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PS %rule 9% if error is NS and change in error is PS then output is NC
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p7_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule9*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule9*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PL %rule 10% if error is NS and change in error is PL then output is SD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy22_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy22_change_error=1;
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule10*yy22_error;
        hh22 = yy22_error;
    else

```

```

        yy22 = Rule10*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end
end

if xx1==ZE
    if (error>=xmin_error) & (error<0)
        yy22_error=polyval(p4_error,error);
    elseif (error>=0) & (error<=xmax_error)
        yy22_error=polyval(p5_error,error);
    end

    if xx4==NL %rule 11% if error is ZE and change in error is NL then output is MI
        if change_error<=-1
            yy22_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p1_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule11*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule11*yy22_change_error;
            hh22 = yy22_change_error;
        end
    end

elseif xx4==NS %rule 12% if error is ZE and change in error is NS then output is SI
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy22_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p3_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule12*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule12*yy22_change_error;
        hh22 = yy22_change_error;
    end
end

elseif xx4==ZE %rule 13% if error is ZE and change in error is ZE then output is NC
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p5_change_error,change_error);
    end
    if yy22_error <= yy11_change_error
        yy22 = Rule13*yy11_error;
        hh22 = yy22_error;
    end
end

```

```

else
    yy22 = Rule13*yy11_change_error;
    hh22 = yy22_change_error;
end

elseif xx4==PS %rule 14% if error is ZE and change in error is PS then output is SD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p7_change_error,change_error);
    end
    if yy22_error <= yy11_change_error
        yy22 = Rule14*yy11_error;
        hh22 = yy22_error;
    else
        yy22 = Rule14*yy11_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PL %rule 15% if error is ZE and change in error is PL then output is MD
    if (change_error>=xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy22_change_error=1;
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule15*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule15*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end

end

if xx1==PS
    if (error>=0) & (error<=xmax_error)
        yy22_error=polyval(p6_error,error);
    elseif (error>xmax_error) & (error<=1)
        yy22_error=polyval(p7_error,error);
    end

    if xx4==NL %rule 16% if error is PS and change in error is NL then output is SI
        if change_error<=-1
            yy22_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p1_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule16*yy22_error;

```

```

        hh22 = yy22_error;
    else
        yy22 = Rule16*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==NS %rule 17% if error is PS and change in error is NS then output is NC
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy22_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p3_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule17*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule17*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZE %rule 18% if error is PS and change in error is ZE then output is SD
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy22_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p5_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule18*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule18*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PS %rule 19% if error is PS and change in error is PS then output is MD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p7_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule19*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule19*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PL %rule 20% if error is PS and change in error is PL then output is LD
    if (change_error>=xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy22_change_error=1;
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule20*yy22_error;

```

```

        hh22 = yy22_error;
    else
        yy22 = Rule20*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end
end

if xx1==PL
    if (error>=xmax_error) & (error<1)
        yy22_error=polyval(p8,error);
    elseif error>=1
        yy22_error=1;
    end

    if xx4==NL %%rule 21% if error is PL and change in error is NL then output is NC
        if change_error<=-1
            yy22_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy22_change_error=polyval(p1_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule21*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule21*yy22_change_error;
            hh22 = yy22_change_error;
        end

    elseif xx4==NS %rule 22% if error is PL and change in error is NS then output is SD
        if (change_error>=-1) & (change_error<xmin_change_error)
            yy22_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>=xmin_change_error) & (change_error<=0)
            yy22_change_error=polyval(p3_change_error,change_error);
        end
        if yy22_error <= yy22_change_error
            yy22 = Rule22*yy22_error;
            hh22 = yy22_error;
        else
            yy22 = Rule22*yy22_change_error;
            hh22 = yy22_change_error;
        end

    elseif xx4==ZE %rule 23% if error is PL and change in error is ZE then output is MD
        if (change_error>=xmin_change_error) & (change_error<=0)
            yy22_change_error=polyval(p4_change_error,change_error);
        elseif (change_error>0) & (change_error<=xmax_change_error)
            yy22_change_error=polyval(p5_change_error,change_error);
        end
        if yy22_error <= yy22_change_error

```

```

        yy22 = Rule23*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule23*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PS %rule 24% if error is PL and change in error is PS then output is LD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy22_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy22_change_error=polyval(p7_change_error,change_error);
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule24*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule24*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==PL %rule 25% if error is PL and change in error is PL then output is XLD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy22_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy22_change_error=1;
    end
    if yy22_error <= yy22_change_error
        yy22 = Rule25*yy22_error;
        hh22 = yy22_error;
    else
        yy22 = Rule25*yy22_change_error;
        hh22 = yy22_change_error;
    end

elseif xx4==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end

end

if xx1==ZD
    yy22_change_error=0;
    yy22_error=0;
    yy22 = 0;
    hh22 = 0;
end

%%%%defuzzification%%%%%%%%
%%%%%%%%xx2,xx3%%%%%%%%

if xx2==NL
    if error<=-1

```

```

        yy33_error=1;
elseif (error>-1) & (error<=xmin_error)
    yy33_error=polyval(p1_error,error);
end

if xx3==NL %rule 1% if error is NL and change in error is NL then output is XLI
    if change_error<=-1
        yy33_change_error=1;
    elseif (change_error>-1) & (change_error<=xmin_change_error)
        yy33_change_error=polyval(p1_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule1*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule1*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==NS %rule 2% if error is NL and change in error is NS then output is LI
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy33_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy33_change_error=polyval(p3_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule2*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule2*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZE %rule 3% if error is NL and change in error is ZE then output is MI
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy33_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p5_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule3*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule3*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==PS %rule 4% if error is NL and change in error is PS then output is SI
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy33_change_error=polyval(p7_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule4*yy33_error;
        hh33 = yy33_error;

```

```

else
    yy33 = Rule4*yy33_change_error;
    hh33 = yy33_change_error;
end

elseif xx3==PL %rule 5% if error is NL and change in error is PL then output is NC
    if (change_error>=xmax_change_error) & (change_error<1)
        yy33_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy33_change_error=1;
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule5*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule5*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;
end

end

if xx2==NS
    if (error>=-1) & (error<=xmin_error)
        yy33_error=polyval(p2_error,error);
    elseif (error>xmin_error) & (error<=0)
        yy33_error=polyval(p3_error,error);
    end

    if xx3==NL %rule 6% if error is NS and change in error is NL then output is LI
        if change_error<=-1
            yy33_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p1_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule6*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule6*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==NS %rule 7% if error is NS and change in error is NS then output is MI
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy33_change_error=polyval(p3_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule7*yy33_error;

```



```

        hh33 = yy33_error;
    else
        yy33 = Rule7*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZE %rule 8% if error is NS and change in error is ZE then output is SI
    if (change_error>xmin_change_error) & (change_error<0)
        yy33_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p5_change_error,change_error);
    if yy33_error <= yy33_change_error
        yy33 = Rule8*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule8*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==PS %rule 9% if error is NS and change in error is PS then output is NC
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy33_change_error=polyval(p7_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule9*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule9*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==PL %rule 10% if error is NS and change in error is PL then output is SD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy33_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy33_change_error=1;
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule10*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule10*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;
end
end
end

```

```

if xx2==ZE
    if (error>=xmin_error) & (error<=0)
        yy33_error=polyval(p4_error,error);
    elseif (error>0) & (error<=xmax_error)
        yy33_error=polyval(p5_error,error);
    end

    if xx3==NL %rule 11% if error is ZE and change in error is NL then output is MI
        if change_error<=-1
            yy33_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p1_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule11*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule11*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==NS %rule 12% if error is ZE and change in error is NS then output is SI
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy33_change_error=polyval(p3_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule12*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule12*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==ZE %rule 13% if error is ZE and change in error is ZE then output is NC
        if (change_error>=xmin_change_error) & (change_error<0)
            yy33_change_error=polyval(p4_change_error,change_error);
        elseif (change_error>=0) & (change_error<=xmax_change_error)
            yy33_change_error=polyval(p5_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule13*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule13*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==PS %rule 14% if error is ZE and change in error is PS then output is SD
        if (change_error>=0) & (change_error<=xmax_change_error)
            yy33_change_error=polyval(p6_change_error,change_error);
        elseif (change_error>xmax_change_error) & (change_error<=1)
            yy33_change_error=polyval(p7_change_error,change_error);
        end

```

```

        if yy33_error <= yy33_change_error
            yy33 = Rule14*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule14*yy33_change_error;
            hh33 = yy33_change_error;
        end

elseif xx3==PL %rule 15% if error is ZE and change in error is PL then output is MD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy33_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy33_change_error=1;
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule15*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule15*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;
end

end

if xx2==PS
    if (error>=0) & (error<=xmax_error)
        yy33_error=polyval(p6_error,error);
    elseif (error>xmax) & (error<=1)
        yy33_error=polyval(p7_error,error);
    end

    if xx3==NL %%rule 16% if error is PS and change in error is NL then output is SI
        if change_error<=-1
            yy33_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p1_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule16*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule16*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==NS %rule 17% if error is PS and change in error is NS then output is NC
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy33_change_error=polyval(p3_change_error,change_error);
        end
    end
end

```

```

end
if yy33_error <= yy33_change_error
    yy33 = Rule17*yy33_error;
    hh33 = yy33_error;
else
    yy33 = Rule17*yy33_change_error;
    hh33 = yy33_change_error;
end

elseif xx3==ZE %rule 18% if error is PS and change in error is ZE then output is SD
    if (change_error>=xmin_change_error) & (change_error<0)
        yy33_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>=0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p5_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule18*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule18*yy33_change_error;
        hh33 = yy33_change_error;
    end
end

elseif xx3==PS %rule 19% if error is PS and change in error is PS then output is MD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy33_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy33_change_error=polyval(p7_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule19*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule19*yy33_change_error;
        hh33 = yy33_change_error;
    end
end

elseif xx3==PL %rule 20% if error is PS and change in error is PL then output is LD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy33_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy33_change_error=1;
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule20*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule20*yy33_change_error;
        hh33 = yy33_change_error;
    end
end

elseif xx3==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;

end
end

```

```

end

if xx2==PL
    if (error>=xmax_error) & (error<1)
        yy33_error=polyval(p8_error,error);
    elseif error>=1
        yy33_error=1;
    end

    if xx3==NL %rule 21% if error is PL and change in error is NL then output is NC
        if change_error<=-1
            yy33_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy33_change_error=polyval(p1_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule21*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule21*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==NS %rule 22% if error is PL and change in error is NS then output is SD
        if (change_error>=-1) & (change_error < xmin_change_error)
            yy33_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>=xmin_change_error) & (change_error<=0)
            yy33_change_error=polyval(p3_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule22*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule22*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==ZE %rule 23% if error is PL and change in error is ZE then output is MD
        if (change_error>=xmin_change_error) & (change_error<=0)
            yy33_change_error=polyval(p4_change_error,change_error);
        elseif (change_error>=0) & (change_error<=xmax_change_error)
            yy33_change_error=polyval(p5_change_error,change_error);
        end
        if yy33_error <= yy33_change_error
            yy33 = Rule23*yy33_error;
            hh33 = yy33_error;
        else
            yy33 = Rule23*yy33_change_error;
            hh33 = yy33_change_error;
        end

    elseif xx3==PS %rule 24% if error is PL and change in error is PS then output is LD
        if (change_error>=0) & (change_error<=xmax_change_error)
            yy33_change_error=polyval(p6_change_error,change_error);
        elseif (change_error>xmax_change_error) & (change_error<=1)

```

```

        yy33_change_error=polyval(p7_change_error,change_error);
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule24*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule24*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==PL %rule 25% if error is PL and change in error is PL then output is XLD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy33_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy33_change_error=1;
    end
    if yy33_error <= yy33_change_error
        yy33 = Rule25*yy33_error;
        hh33 = yy33_error;
    else
        yy33 = Rule25*yy33_change_error;
        hh33 = yy33_change_error;
    end

elseif xx3==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;
end

end

if xx2==ZD
    yy33_change_error=0;
    yy33_error=0;
    yy33 = 0;
    hh33 = 0;
end

%%%%defuzzification%%%%%%%%
%%%%%%%%xx2,xx4%%%%%%%%

if xx2==NL
    if error<=-1
        yy44_error=1;
    elseif (error>-1) & (error<=xmin_change_error)
        yy44_error=polyval(p1_error,error);
    end

    if xx4==NL %rule 1% if error is NL and change in error is NL then output is XLI
        if change_error<=-1
            yy44_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy44_change_error=polyval(p1_change_error,change_error);
        end
    end
end

```

```

        if yy44_error <= yy44_change_error
            yy44 = Rule1*yy44_error;
            hh44 = yy44_error;
        else
            yy44 = Rule1*yy44_change_error;
            hh44 = yy44_change_error;
        end

elseif xx4==NS %rule 2% if error is NL and change in error is NS then output is LI
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy44_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p3_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule2*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule2*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZE %rule 3% if error is NL and change in error is ZE then output is MI
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p5_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule3*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule3*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PS % 4% if error is NL and change in error is PS then output is SI
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy44_change_error=polyval(p7_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule4*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule4*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PL %rule 5% if error is NL and change in error is PL then output is NC
    if (change_error>=xmax_change_error) & (change_error<1)
        yy44_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy44_change_error=1;

```

```

        end
        if yy44_error <= yy44_change_error
            yy44 = Rule5*yy44_error;
            hh44 = yy44_error;
        else
            yy44 = Rule5*yy44_change_error;
            hh44 = yy44_change_error;
        end

elseif xx4==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44 = 0;
end
end

if xx2==NS
    if (error>=-1) & (error<=xmin_error)
        yy44_error=polyval(p2_error,error);
    elseif (error>xmin_error) & (error<=0)
        yy44_error=polyval(p3_error,error);
    end

    if xx4==NL %rule 6% if error is NS and change in error is NL then output is LI
        if change_error<=-1
            yy44_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy44_change_error=polyval(p1_change_error,change_error);
        end
        if yy44_error <= yy44_change_error
            yy44 = Rule6*yy44_error;
            hh44 = yy44_error;
        else
            yy44 = Rule6*yy44_change_error;
            hh44 = yy44_change_error;
        end
    end

elseif xx4==NS %rule 7% if error is NS and change in error is NS then output is MI
    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy44_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p3_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule7*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule7*yy44_change_error;
        hh44 = yy44_change_error;
    end
end

elseif xx4==ZE %rule 8% if error is NS and change in error is ZE then output is SI
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)

```



```

        yy44_change_error=polyval(p5_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule8*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule8*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PS %rule 9% if error is NS and change in error is PS then output is NC
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy44_change_error=polyval(p7_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule9*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule9*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PL %rule 10% if error is NS and change in error is PL then output is SD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy44_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy44_change_error=1;
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule10*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule10*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44 = 0;
end
end

if xx2==ZE
    if (error>=xmin_error) & (error<=0)
        yy44_error=polyval(p4_error,error);
    elseif (error>0) & (error<=xmax_error)
        yy44_error=polyval(p5_error,error);
    end

    if xx4==NL %rule 11% if error is ZE and change in error is NL then output is MI
        if change_error<=-1
            yy44_change_error=1;

```

```

elseif (change_error>-1) & (change_error<=xmin_change_error)
    yy44_change_error=polyval(p1_change_error,change_error);
end
if yy44_error <= yy44_change_error
    yy44 = Rule11*yy44_error;
    hh44 = yy44_error;
else
    yy44 = Rule11*yy44_change_error;
    hh44 = yy44_change_error;
end

elseif xx4==NS %rule 12% if error is ZE and change in error is NS then output is SI

    if (change_error>=-1) & (change_error<=xmin_change_error)
        yy44_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p3_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule12*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule12*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZE %rule 13% if error is ZE and change in error is ZE then output is NC
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p5_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule13*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule13*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PS %rule 14% if error is ZE and change in error is PS then output is SD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy44_change_error=polyval(p7_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule14*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule14*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PL %rule 15% if error is ZE and change in error is PL then output is MD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy44_change_error=polyval(p8_change_error,change_error);

```

```

elseif change_error>=1
    yy44_change_error=1;
end
if yy44_error <= yy44_change_error
    yy44 = Rule15*yy44_error;
    hh44 = yy44_error;
else
    yy44 = Rule15*yy44_change_error;
    hh44 = yy44_change_error;
end

elseif xx4==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44=0;
end

end

if xx2==PS
    if (error>=0) & (error<=xmax_error)
        yy44_error=polyval(p6_error,error);
    elseif (error>xmax) & (error<=1)
        yy44_error=polyval(p7_error,error);
    end

    if xx4==NL %rule 16% if error is PS and change in error is NL then output is SI
        if change_error<=-1
            yy44_change_error=1;
        elseif (change_error>-1) & (change_error<=xmin_change_error)
            yy44_change_error=polyval(p1_change_error,change_error);
        end
        if yy44_error <= yy44_change_error
            yy44 = Rule16*yy44_error;
            hh44 = yy44_error;
        else
            yy44 = Rule16*yy44_change_error;
            hh44 = yy44_change_error;
        end
    end

    elseif xx4==NS %rule 17% if error is PS and change in error is NS then output is NC
        if (change_error>=-1) & (change_error<=xmin_change_error)
            yy44_change_error=polyval(p2_change_error,change_error);
        elseif (change_error>xmin_change_error) & (change_error<=0)
            yy44_change_error=polyval(p3_change_error,change_error);
        end
        if yy44_error <= yy44_change_error
            yy44 = Rule17*yy44_error;
            hh44 = yy44_error;
        else
            yy44 = Rule17*yy44_change_error;
            hh44 = yy44_change_error;
        end
    end

    elseif xx4==ZE %rule 18% if error is PS and change in error is ZE then output is SD
        if (change_error>=xmin_change_error) & (change_error<=0)

```

```

        yy44_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p5_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule18*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule18*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PS %rule 19% if error is PS and change in error is PS then output is MD
    if (change_error>=0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy44_change_error=polyval(p7_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule19*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule19*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PL %rule 20% if error is PS and change in error is PL then output is LD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy44_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy44_change_error=1;
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule20*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule20*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44=0;
end
end

if xx2==PL
    if (error>=xmax_error) & (error<1)
        yy44_error=polyval(p8_error,error);
    elseif error>=1
        yy44_error=1;
    end

    if xx4==NL %rule 21% if error is PL and change in error is NL then output is NC

```

```

if change_error<=-1
    yy44_change_error=1;
elseif (change_error>-1) & (change_error<=xmin_change_error)
    yy44_change_error=polyval(p1_change_error,change_error);
end
if yy44_error <= yy44_change_error
    yy44 = Rule21*yy44_error;
    hh44 = yy44_error;
else
    yy44 = Rule21*yy44_change_error;
    hh44 = yy44_change_error;
end

elseif xx4==NS %%rule 22% if error is PL and change in error is NS then output is SD

    if (change_error>=-1)&(change_error<xmax_change_error)
        yy44_change_error=polyval(p2_change_error,change_error);
    elseif (change_error>=xmax_change_error) & (change_error<=0)
        yy44_change_error=polyval(p3_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule22*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule22*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZE %rule 23% if error is PL and change in error is ZE then output is MD
    if (change_error>=xmin_change_error) & (change_error<=0)
        yy44_change_error=polyval(p4_change_error,change_error);
    elseif (change_error>0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p5_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule23*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule23*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==PS %rule 24% if error is PL and change in error is PS then output is LD

    if (change_error>=0) & (change_error<=xmax_change_error)
        yy44_change_error=polyval(p6_change_error,change_error);
    elseif (change_error>xmax_change_error) & (change_error<=1)
        yy44_change_error=polyval(p7_change_error,change_error);
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule24*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule24*yy44_change_error;
        hh44 = yy44_change_error;
    end
end

```

```

elseif xx4==PL %rule 25% if error is PL and change in error is PL then output is XLD
    if (change_error>=xmax_change_error) & (change_error<1)
        yy44_change_error=polyval(p8_change_error,change_error);
    elseif change_error>=1
        yy44_change_error=1;
    end
    if yy44_error <= yy44_change_error
        yy44 = Rule25*yy44_error;
        hh44 = yy44_error;
    else
        yy44 = Rule25*yy44_change_error;
        hh44 = yy44_change_error;
    end

elseif xx4==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44 = 0;
end
end

if xx2==ZD
    yy44_change_error=0;
    yy44_error=0;
    yy44 = 0;
    hh44 = 0;
end

control=(yy11+yy22+yy33+yy44)/(hh11+hh22+hh33+hh44);

```

```

*****

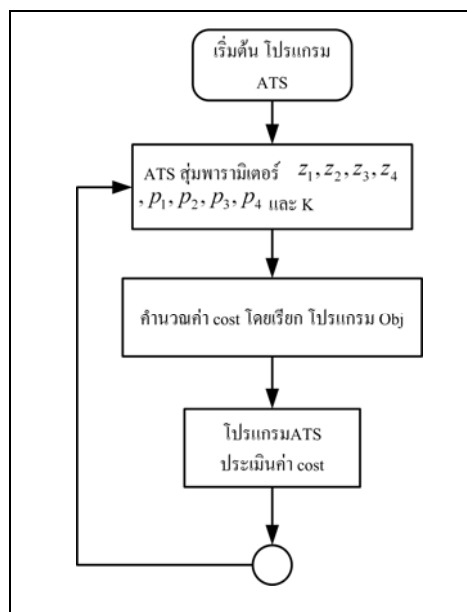
```

ภาคผนวก ค

รายละเอียดของโปรแกรม ATS เพื่อออฟติไมซ์ตัวชดเชย

รายละเอียดของโปรแกรม ATS

โปรแกรมการค้นหาแบบตามูเชิงปรับตัว หรือโปรแกรม ATS ในงานวิจัยวิทยานิพนธ์นี้ ได้ใช้โปรแกรมที่พัฒนาขึ้นโดยอาศัยโปรแกรม MATLABTM การกำหนดพารามิเตอร์ต่าง ๆ ของ ATS ได้กล่าวไว้แล้วในบทที่ 5 รายละเอียดของการเรียกฟังก์ชันวัตถุประสงค์แสดงด้วยแผนภูมิรูปที่ ค.1 โปรแกรมฟังก์ชันวัตถุประสงค์ หรือ Obj เป็นโปรแกรมที่ทำการจำลองสถานการณ์ของหัวอ่านฮาร์ดดิสก์ในย่านความถี่สูง โดยใช้แบบจำลองทางคณิตศาสตร์ดังสมการที่ (2-5) การคำนวณค่า cost มีรายละเอียดของโปรแกรมแสดงไว้ในภาคผนวก ค.



รูปที่ ค.1 การเรียกฟังก์ชันวัตถุประสงค์

โปรแกรม ATS

โดย วิไลพร เงินบาท (ธันวาคม 2550) สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรม ATS.m

```
clear all;
```

```
clc;
```

```
N=9;
```

```
xlimit=[100000 100000 100000 100000 100000 100000 100000 100000 100000;0 0 0 0 0 0 0 0]
```



```

for r=1:50
S(r,:)=((xlimit(1:)-xlimit(2,:)).*rand(1,N))+xlimit(2,:)
End
for k=1:size(S,1)
ysim=obj(S(k,1),S(k,2),S(k,3),S(k,4),S(k,5),S(k,6),S(k,7),S(k,8),S(k,9));
costvalue(k,1)=ysim;
[best_error,index]=min(costvalue);
S0=S(index,:);
max_count=100000;
best_neighbor=S0;
neighbor_list=zeros(5,N);
radius=10;
Number_neighb=50;
overall_best_error=best_error;
overall_neighbor=best_neighbor;
n=0;
t=0;
tt=0;
ttt=0;
count=0;
n_back_tracking=0;
tic;
t=t+1;
tt=tt+1;
ttt=ttt+1;
local(t,1)=count;
local(t,2:10)=best_neighbor;
local(t,4)=best_error;
tabu_list(tt,1)=count;
tabu_list(tt,2:10)=best_neighbor;
tabu_list(tt,4)=best_error;
best_error_list(ttt,1)=count;
best_error_list(ttt,2:10)=best_neighbor;
best_error_list(ttt,4)=best_error;
for count=1:max_count
    S1=random_neigh(Number_neighb,radius,xlimit,S0);
    [best_error1,best_neighbor1,best_error,best_neighbor]=objective(S1,best_error,S0);
    neighbor_list(k,1:size(S0,2))=[best_neighbor1];
    neighbor_list(k,size(S0,2)+1)=best_error1;
    if (best_error1-best_error)>1e-18
        n=n+1;
    else
        n=0;
    end
    tt=tt+1;
    tabu_list(tt,1)=count;
    tabu_list(tt,2:10)=best_neighbor1;
    tabu_list(tt,4)=best_error1;
    ttt= ttt+1;
    best_error_list(ttt,1)=count;
    best_error_list(ttt,2:10)=best_neighbor;
    best_error_list(ttt,4)=best_error;
    disp([count best_error overall_best_error])

if best_error<=20
radius=8;

```

```

end
if best_error<=15
radius=6;
end
if best_error<12
radius=4;
end

if (best_error<10)
t=t+1;
tt=tt+1;
disp([count best_error overall_best_error])
local(t,1)=count;
local(t,2:10)=best_neighbor;
local(t,4)=best_error;
break;
end

if n>=5
n_back_tracking=n_back_tracking+1;
TEMP=tabu_list(count-3:count+1,:);
[MAX,INDEX] = max(TEMP(:,4));
RANK(5,:)=TEMP(INDEX,:);
TEMP(INDEX,4)=0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(4,:)=TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(3,:)=TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(2,:)=TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
[MAX,INDEX] = max(TEMP(:,4));
RANK(1,:)=TEMP(INDEX,:);
TEMP(INDEX,4)= 0;
neighbor=RANK(5,2:10);
S0=neighbor;

if best_error<overall_best_error
overall_best_error=best_error;
overall_best_neighbor=best_neighbor;
t=t+1;
local(t,1)=count;
local(t,2:10)=best_neighbor;
local(t,4)=best_error
end
best_error=RANK(5,4);
n=0;
else
S0=best_neighbor;
best_error=best_error;
end
end
if overall_best_error<best_error
best_error=overall_best_error;

best_neighbor=overall_best_neighbor;

```

```

end
time=toc;
return

*****

Program Obj.m
function f=obj(Z1,Z2,Z3,Z4,P1,P2,P3,P4,K)
z1=[0.8709 1726 1.369e9];
p1=[1 1480 1.369e9];
G1=tf(z1,p1);

z2=[0.9332 -805.8 1.739e9];
p2=[1 125.1 1.739e9];
G2=tf(z2,p2);

z3=[1.072 925.1 1.997e9];
p3=[1 536.2 1.997e9];
G3=tf(z3,p3);

z4=[0.9594 98.22 2.514e9];
p4=[1 1805 2.514e9];
G4=tf(z4,p4);

z5=[0 0 7.877e9];
p5=[1 6212 7.877e9];
G5=tf(z5,p5);

z6=[0 0 2.35e8];
p6=[1 0 0];
G6=tf(z6,p6);

Gp=G1*G2*G3*G4*G5*G6;

%%%%%%%%%compensator%%%%%%%%%

Gc1=tf([1 Z1],[1 P1]);
Gc2=tf([1 Z2],[1 P2]);
Gc3=tf([1 Z3],[1 P3]);
Gc4=tf([1 Z4],[1 P4]);
Gc5=K*series(Gc1,Gc2);
Gc6=series(Gc3,Gc4);
Gc=series(Gc5,Gc6);

Gf=series(Gp,Gc);
sys=feedback(Gf,1,-1);

fstep=100000;
t=[0:1/fstep:1];
[fsim t]=step(sys,t);
error=sum(abs(1-fsim))
f=error
return

*****

```

```

function [best_error1,best_neighbor1,best_error,best_neighbor]=objective(S1,best_error,S0)
error=[];
for k=1:size(S1,1)
ysim=obj(S1(k,1),S1(k,2),S1(k,3),S1(k,4),S1(k,5),S1(k,6),S1(k,7),S1(k,8),S1(k,9));
error(k,1)=ysim;
end
[best_error1,index]=min(error);
best_neighbor1=S1(index,:);
if best_error1<best_error
best_error=best_error1;
best_neighbor=S1(index,:);
else
best_neighbor=S0;
end
return

```

Program rand1.m

```

function x=rand1(a,b)
x=a+rand*(b-a);
return

```

Program random_neigh.m

```

function S1=random_neigh(Number_neighb,radius,xlimit,S0)
for u=1:Number_neighb
for k=1:size(xlimit,2)
S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
while ( S1(u,k)>xlimit(1,k) | S1(u,k)<xlimit(2,k))
S1(u,k)=S0(1,k)+(radius*(xlimit(1,k)-xlimit(2,k))*rand1(-1,1));
end
end
end
return

```

ภาคผนวก ง

การเผยแพร่ผลงานขณะศึกษา

การเผยแพร่ผลงานขณะศึกษา

บทความวิจัยที่ได้รับการตีพิมพ์ในวารสารวิชาการนานาชาติ

Ngernbaht, W., Areerak, K-L., and Sujijorn, S. (2009) **Resonance and Friction Compensations in a Micro Hard Drive**. WSEAS Trans. on Systems. 8(2): 179-188.

บทความวิจัยที่ได้รับการตีพิมพ์ในการประชุมวิชาการนานาชาติ

Ngernbaht, W., Areerak, K-L., and Sujijorn, S. (2009) **Fuzzy Compensation for Nonlinear Friction in a Hard Drive**. WSEAS Proc. Int. Conf. Fuzzy Systems Processing. (FS'09), Prague Czech Republic, 23-25 March.: 35-40.

การเผยแพร่ผลงานในการประชุมวิชาการระดับชาติ

วิไลพร เงินบาท กองพล อารีรักษ์ และ สราวุฒิ สุจิตจร, (2551) **การปรับปรุงเสถียรภาพและสมรรถนะของหัวอ่านฮาร์ดดิสก์ในการติดตามแทร็กข้อมูล**. การประชุมวิชาการเพื่อการนำเสนอผลงานวิจัย ณ มหาวิทยาลัยบูรพา จังหวัดชลบุรี, 7 กรกฎาคม.

ประวัติผู้เขียน

นางสาววิไลพร เงินบาท เกิดเมื่อวันที่ 6 ธันวาคม 2525 ที่จังหวัดกาญจนบุรี เริ่มการศึกษาระดับประถมศึกษา ที่โรงเรียนเทศบาล 4 (บ้านชุกชุม) อำเภอเมือง จังหวัดกาญจนบุรี และระดับระดับมัธยมศึกษา ที่โรงเรียนกาญจนาภิเษก อำเภอเมือง จังหวัดกาญจนบุรี สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2546 หลังจากสำเร็จการศึกษาได้เข้าทำงานเป็นผู้ช่วยสอนและวิจัย สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์มหาวิทยาลัยเทคโนโลยีสุรนารี เป็นเวลา 2 ปี ทำให้เกิดแรงจูงใจที่จะศึกษาต่อในระดับปริญญาโท จึงได้เข้าศึกษาต่อในระดับปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยี สุรนารี ในปี พ.ศ. 2549 โดยได้รับทุนการศึกษาและวิจัยจาก “ทุนพัฒนาทรัพยากรบุคคลในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์” และมีผลงานทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในขณะศึกษา 3 บทความ ดังมีรายชื่อที่ปรากฏในภาคผนวก ง.