

การปรับแต่งค่าพารามิเตอร์ตัวควบคุมแบบฟัซซี่ฟิสแควรีไอดีด้วย
วิธีทางพันธุศาสตร์

TUNING FUZZY P²ID CONTROLLER USING GENETIC ALGORITHMS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

จพ.

ว 2387

2547

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

ศ.1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

b.....

i.....

50965

พ.ศ. 2547

เลขทะเบียน.....
วัน,เดือน,ปี 26 พ.ค. 2547

ISBN 974-324-989-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TUNING FUZZY P²ID CONTROLLER USING GENETIC ALGORITHMS



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

2004

ISBN 974-324-989-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2004

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

การปรับแต่งค่าพารามิเตอร์ตัวควบคุมแบบฟัซซีฟัสแควรีไอดี
ด้วยวิธีทางพันธุศาสตร์

นักศึกษา

นายทศพล อ่อนบุญเชื้อ

รหัสนักศึกษา

44061812

ปริญญา

วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา

วิศวกรรมสารสนเทศ

พ.ศ.

2547

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร. ปิติเขต สู้รักษา

บทคัดย่อ

วิทยานิพนธ์เสนอการปรับแต่งค่าพารามิเตอร์ ของตัวควบคุมฟัซซีฟัสแควรีไอดีโดยใช้วิธีทางพันธุศาสตร์ โดยใช้ตัวควบคุม 2 ชนิด คือตัวควบคุมฟัซซีฟัวและฟัซซีฟิดี ซึ่งทำงานขนานกัน และเลือกตัวควบคุมโดยใช้สวิตช์เป็นตัวเลือก โดยอาศัยเงื่อนไขเสถียรภาพพอเพียง (sufficient condition) หน้าที่ที่สำคัญของตัวควบคุมแบบฟัซซีฟิดี คือทำให้เอาต์พุตที่ได้นั้นมีเวลาไต่ขึ้นที่ดี และเมื่อถึงเวลาที่ได้ออกแบบไว้ ก็จะเปลี่ยนไปใช้ตัวควบคุมแบบฟัซซีฟัว ซึ่งมีความคงทน (robustness) ขณะสถานะคงตัว (steady state) การปรับแต่งพารามิเตอร์ของตัวควบคุมฟัซซีฟัสแควรีไอดีและเวลาสวิตช์ จะอาศัยวิธีทางพันธุศาสตร์ ผลการออกแบบนำไปใช้กับแบบจำลองที่ไม่เป็นเชิงเส้น และแบบจำลองของหุ่นยนต์เคลื่อนที่ ผลการจำลองที่ได้แสดงว่าตัวควบคุมที่ได้ออกแบบมีประสิทธิภาพ

Thesis Title	Tuning Fuzzy P ² ID Controller Using Genetic Algorithms
Student	Mr.Tossapon Onboonare
Student ID.	44061812
Degree	Master of Engineering
Programme	Information Engineering
Year	2004
Thesis Advisor	Assoc.Prof.Dr. Pitikhate Sooraksa

ABSTRACT

This thesis deals with tuning parameters for a new fuzzy P²ID controller using genetic algorithms. This type of controller composes of two parts, which are a fuzzy PD controller and a fuzzy PI controller working in parallel processing. A sub-controller is selected one at a time to control the system using switching mechanism performed under given sufficient conditions for stability. Functionally, the fuzzy PD controller is used to launch the output for good rise-time, then, at a designed time, it will switch to the fuzzy PI controller for robustness and good steady state correction. The designed switching-time and control parameters are tuned by using Genetic Algorithms (GAs). Given a nonlinear plant and a mobile robot model, the simulation results show effectiveness of the designed system.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับแนวคิด หลักการ และการประยุกต์ใช้งานของระบบควบคุมให้มีความถูกต้อง ข้อคิดเห็น คำเสนอแนะอันมีค่า และคำปรึกษาต่าง ๆ ที่เป็นประโยชน์แก่ข้าพเจ้า ตลอดจนช่วยตรวจทานแก้ไขวิทยานิพนธ์ฉบับนี้จนเสร็จสมบูรณ์จาก รศ.ดร.ปิติเชต สุรักษา ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ทุกท่านที่ได้ประสาทความรู้ให้แก่ข้าพเจ้าตลอดระยะเวลาของการเรียนจนกระทั่งข้าพเจ้ามีโอกาสประสบความสำเร็จ

ขอกราบขอบพระคุณบิดา มารดาของข้าพเจ้า ที่ให้การสนับสนุน และให้กำลังใจในการเรียน และการทำวิจัยเป็นอย่างดีตลอดมา

สุดท้ายขอขอบคุณเพื่อน ๆ พี่ ๆ นักศึกษาปริญญาโทที่ให้คำแนะนำเป็นอย่างดีตลอดมา คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ทศพล อ่อนบุญเชื้อ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	2
1.6 ขั้นตอนการศึกษา.....	3
บทที่ 2 ตัวควบคุมแบบฟัซซีฟัสแควร์ไอดี.....	4
2.1 ตัวควบคุมแบบฟัซซีฟิดี.....	4
2.1.1 การออกแบบตัวควบคุมแบบฟัซซีฟิดี.....	4
2.1.2 การทำฟัซซีฟิเคชัน.....	6
2.1.3 การกำหนดกฎของฟัซซีฟิดี.....	7
2.1.4 การทำดีฟัซซีฟิเคชันของฟัซซีฟิดี.....	9
2.2 ตัวควบคุมแบบฟัซซีฟีโอ.....	11
2.2.1 การออกแบบตัวควบคุมแบบฟัซซีฟีโอ.....	11
2.2.2 การทำฟัซซีฟิเคชัน.....	13
2.2.3 การกำหนดกฎของฟัซซีฟีโอ.....	13
2.2.4 การทำดีฟัซซีฟิเคชันของฟัซซีฟีโอ.....	15
2.3 ตัวควบคุมแบบฟัซซีฟัสแควร์ไอดี.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตัด IV อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีการทางพันธุศาสตร์.....	20
3.1 การแทนที่ประชากร.....	21
3.2 ฟังก์ชันเป้าหมายและฟังก์ชันฟิตเนส.....	22
3.3 การคัดเลือก.....	23
3.3.1 การคัดเลือกแบบวงล้อ roulette.....	24
3.3.2 การคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม.....	25
3.4 การรวมตัวของยีน.....	25
3.4.1 การรวมตัวของยีนโดยใช้ค่าจริง.....	25
3.4.1.1 การรวมตัวของยีนแบบไม่ต่อเนื่อง.....	25
3.4.1.2 การรวมตัวของยีนแบบชั้นกลาง.....	26
3.4.1.3 การรวมตัวของยีนแบบเส้น.....	28
3.4.2 การรวมตัวของยีนโดยใช้ค่าไบนารี.....	29
3.4.2.1 การครอสโอเวอร์แบบจุดเดียว.....	29
3.4.2.2 การครอสโอเวอร์แบบหลายจุด.....	30
3.4.2.3 การครอสโอเวอร์แบบยูนิฟอร์ม.....	31
3.5 การกลายพันธุ์.....	31
3.6 การแทนที่.....	31
บทที่ 4 การออกแบบและหลักการทำงาน.....	33
4.1 การออกแบบโปรแกรมทางพันธุศาสตร์.....	33
4.1.1 การแทนที่ประชากรและการกำหนดค่าตัวแปรเริ่มต้น.....	34
4.1.2 การคัดเลือก.....	35
4.1.3 การครอสโอเวอร์.....	35
4.1.4 การกลายพันธุ์.....	36
4.1.5 การสำเนาประชากร.....	36
4.1.6 ฟังก์ชันเป้าหมาย.....	36

สารบัญ (ต่อ)

	หน้า
4.2 การทดสอบโปรแกรมทางพันธุศาสตร์.....	36
4.3 โปรแกรม GA พืชที่พืชแคร์ไอดีแบบ 8 ตัวแปร.....	39
4.3.1 โปรแกรม GA พืชที่พืชดี.....	40
4.3.2 โปรแกรม GA พืชที่พืชไอ.....	41
4.3.3 โปรแกรมพืชที่พืชแคร์ไอดี.....	42
4.4 โปรแกรม GA พืชที่พืชแคร์ไอดีแบบ 10 ตัวแปร.....	44
บทที่ 5 ผลการจำลองและวิจารณ์ผลการจำลอง.....	46
5.1 การจำลองหาค่าพารามิเตอร์ของตัวควบคุมแบบพืชที่พืชแคร์ไอดีแบบ 8 ตัวแปร	46
5.1.1 แบบจำลองที่ไม่เป็นเชิงเส้น.....	46
5.1.2 แบบจำลองของหุ่นยนต์เคลื่อนที่.....	62
5.2 การจำลองหาค่าพารามิเตอร์ของตัวควบคุมแบบพืชที่พืชแคร์ไอดีแบบ 10 ตัวแปร	79
5.2.1 แบบจำลองที่ไม่เป็นเชิงเส้น.....	79
5.2.2 แบบจำลองของหุ่นยนต์เคลื่อนที่.....	86
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	93
6.1 สรุปผลการจำลอง.....	93
6.2 ปัญหาที่พบในงานวิจัย.....	94
6.3 แนวทางการพัฒนาในอนาคต.....	94
เอกสารอ้างอิง.....	96
ภาคผนวก.....	98
ภาคผนวก ผลงานวิจัยที่ได้ตีพิมพ์.....	99
ภาคผนวก ผลงานวิจัยที่ได้ตีพิมพ์.....	113

ประวัติผู้เขียน.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตี VI อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 กฎของตัวควบคุมแบบพีชชีพีดี.....	7
2.2 กฎของตัวควบคุมแบบพีชชีพีไอ.....	14
3.1 ความน่าจะเป็นของการคัดเลือกและค่าของฟิตเนส.....	24
4.1 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 1.....	37
4.2 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 2.....	37
4.3 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 3.....	37
4.4 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 4.....	38
4.5 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 5.....	38
4.6 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 6.....	38
5.1 ค่าพารามิเตอร์ของตัวควบคุมแบบพีชชีพีดีที่ได้จากวิธีทางพันธุศาสตร์ของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	46
5.2 ค่าพารามิเตอร์ของตัวควบคุมแบบพีชชีพีไอที่ได้จากวิธีทางพันธุศาสตร์ของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	52
5.3 ค่าพารามิเตอร์ที่ปรับโดยผู้ให้ และปรับโดยวิธีทางพันธุศาสตร์ของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	58
5.4 ประสิทธิภาพของการปรับแต่งของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	61
5.5 ค่าพารามิเตอร์ของตัวควบคุมแบบพีชชีพีดีที่ได้จากวิธีทางพันธุศาสตร์ของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	62
5.6 ค่าพารามิเตอร์ของตัวควบคุมแบบพีชชีพีไอที่ได้จากวิธีทางพันธุศาสตร์ของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	68
5.7 ค่าพารามิเตอร์ที่ปรับโดยผู้ให้ และปรับโดยวิธีทางพันธุศาสตร์ของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	74
5.8 ประสิทธิภาพของการปรับแต่งของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	78
5.9 ค่าพารามิเตอร์ของตัวควบคุมแบบพีชชีพีสแควร์ไอดีที่ได้จากวิธีทางพันธุศาสตร์ ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	79

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
5.10 ประสิทธิภาพของการปรับแต่งของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร เปรียบเทียบกับแบบ 8 ตัวแปร.....	85
5.11 ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีสแควร์ไอดีที่ได้จากวิธีทางพันธุศาสตร์ ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	86
5.12 ประสิทธิภาพของการปรับแต่งของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร เปรียบเทียบกับแบบ 8 ตัวแปร.....	92



สารบัญรูป

รูปที่	หน้า
2.1 ระบบควบคุมอนาล็อกพีดีแบบดั้งเดิม.....	4
2.2 ระบบควบคุมดิจิตอลพีดีแบบดั้งเดิม.....	6
2.3 ระบบควบคุมแบบพีซีพีดี.....	6
2.4 ฟังก์ชันสมาชิกของอินพุตของตัวควบคุมแบบพีซีพีดี.....	6
2.5 ฟังก์ชันสมาชิกทางเอาต์พุตของตัวควบคุมแบบพีซีพีดี.....	7
2.6 ผลตอบสนองที่ใช้กำหนดกฎของพีซีพีดี.....	7
2.7 พื้นที่ทั้งหมดของตัวควบคุมพีซีพีดี.....	9
2.8 ระบบควบคุมอนาล็อกพีไอแบบดั้งเดิม.....	11
2.9 ระบบควบคุมดิจิตอลพีไอแบบดั้งเดิม.....	12
2.10 ระบบควบคุมแบบพีซีพีไอ.....	12
2.11 ฟังก์ชันสมาชิกของอินพุตของตัวควบคุมแบบพีซีพีไอ.....	13
2.12 ฟังก์ชันสมาชิกทางเอาต์พุตของตัวควบคุมแบบพีซีพีไอ.....	13
2.13 ผลตอบสนองที่ใช้กำหนดกฎของพีซีพีไอ.....	14
2.14 พื้นที่ทั้งหมดของตัวควบคุมพีซีพีดี.....	15
2.15 ระบบควบคุมแบบพีซีพีเอสแควร์ไอดี.....	17
2.16 ผลตอบสนองของเอาต์พุตเทียบกับเวลา.....	18
2.17 วิธีคำนวณผิดพลาดบนระนาบเฟส.....	19
3.1 โครงสร้างกระบวนการทางพันธุศาสตร์.....	20
3.2 การคัดเลือกแบบวงล้อ roulette.....	24
3.3 วิธีการคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม.....	25
3.4 ตำแหน่งที่เป็นไปได้ที่จะเกิดประชากรลูกหลานหลังจากการรวมยีน.....	26
3.5 พื้นที่ขอบเขตของประชากรลูกหลานที่เทียบกับประชากรพ่อแม่.....	27
3.6 ตำแหน่งที่เป็นไปได้ที่จะเกิดประชากรลูกหลานหลังจากการรวมยีนแบบตัวกลาง.....	28
3.7 ตำแหน่งที่เป็นไปได้ที่จะเกิดประชากรลูกหลานหลังจากการรวมยีนแบบเส้น.....	29
3.8 การครอสโอเวอร์แบบจุดเดียว.....	29
3.9 การครอสโอเวอร์แบบหลายจุด.....	30
4.1 โฟลว์ชาร์ตโปรแกรมทางพันธุศาสตร์.....	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และดัดแปลงอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.2 การแทนตัวแปร X และ Y ของประชากร.....	35
4.3 โพล์ชาร์ตโปรแกรมทางพันธุศาสตร์ของตัวควบคุมพืชซีพีสแควร์ไอดีแบบ 8 ตัวแปร.....	40
4.4 โพล์ชาร์ตโปรแกรมพืชซีพีสแควร์ไอดี.....	43
4.5 โพล์ชาร์ตการทำงานของโปรแกรมทางพันธุศาสตร์ของตัวควบคุมแบบ พืชซีพีสแควร์ไอดีแบบ 10 ตัว.....	44
5.1 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	47
5.2 ผลตอบสนองของตัวควบคุมแบบพืชซีพีดีโดยใช้ประชากรเริ่มต้น 15 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	47
5.3 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	48
5.4 ผลตอบสนองของตัวควบคุมแบบพืชซีพีดีโดยใช้ประชากรเริ่มต้น 30 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	48
5.5 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	49
5.6 ผลตอบสนองของตัวควบคุมแบบพืชซีพีดีโดยใช้ประชากรเริ่มต้น 60 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	49
5.7 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	50
5.8 ผลตอบสนองของตัวควบคุมแบบพืชซีพีดีโดยใช้ประชากรเริ่มต้น 90 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	50
5.9 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	51
5.10 ผลตอบสนองของตัวควบคุมแบบพืชซีพีดีโดยใช้ประชากรเริ่มต้น 120 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	51
5.11 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	53

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.12 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 15 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	53
5.13 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	54
5.14 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 30 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	54
5.15 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	55
5.16 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 60 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	55
5.17 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	56
5.18 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 90 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	56
5.19 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	57
5.20 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 120 ตัวของ แบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	57
5.21 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้ในการปรับแต่งโดย GA-15 ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	59
5.22 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้ในการปรับแต่งโดย GA-30 ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	59
5.23 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้ในการปรับแต่งโดย GA-60 ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	60
5.24 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้ในการปรับแต่งโดย GA-90 ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	60

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.25 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-120 ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร.....	61
5.26 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	63
5.27 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 15 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	63
5.28 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	64
5.29 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 30 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	64
5.30 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	65
5.31 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 60 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	65
5.32 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	66
5.33 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 90 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	66
5.34 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	67
5.35 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 120 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	67
5.36 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	69
5.37 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟีดโดยใช้ประชากรเริ่มต้น 15 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	69

สารบัญญรูป (ต่อ)

รูปที่	หน้า
5.38 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	70
5.39 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 30 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	70
5.40 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	71
5.41 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 60 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	71
5.42 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	72
5.43 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 90 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	72
5.44 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	73
5.45 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 120 ตัวของ แบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	73
5.46 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-15 ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	75
5.47 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-30 ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	75
5.48 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-60 ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	76
5.49 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-90 ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	76
5.50 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-120 ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร.....	77

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.51 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	80
5.52 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 15 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น.....	80
5.53 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	81
5.54 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 30 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น.....	81
5.55 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	82
5.56 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 60 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น.....	82
5.57 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	83
5.58 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 90 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น.....	83
5.59 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง ที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร.....	84
5.60 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น.....	84
5.61 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	87
5.62 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 15 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่.....	87
5.63 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลอง หุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	88

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.64 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 30 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่.....	88
5.65 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	89
5.66 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่.....	89
5.67 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	90
5.68 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่.....	90
5.69 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร.....	91
5.70 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งโดย GA 10 ตัวแปรที่ประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่.....	91

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันตัวควบคุมแบบฟuzzy ใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรม เพราะว่าเป็นตัวควบคุมที่ปรับแต่งค่าพารามิเตอร์ได้ง่าย แต่ตัวควบคุมแบบฟuzzy นี้จะมีข้อจำกัด คือ จะไม่สามารถใช้งานได้ดีกับระบบที่ไม่เป็นเชิงเส้นหรือระบบอันดับสูง (high order) ต่อมา Chen และคณะ[1,2] ปรับปรุงการปรับแต่งค่าพารามิเตอร์ ด้วยการประยุกต์ใช้วิธีการของตรรกศาสตร์ฟuzzy มาใช้กับระบบควบคุม ซึ่งใช้ชื่อว่า ฟuzzy พีดี (Fuzzy PD) และฟuzzy พีไอ (Fuzzy PI) โดยตัวควบคุมทั้งสองจะมีคุณสมบัติที่แตกต่างกัน ต่อมาได้มีการพัฒนาโดยการใช้ประโยชน์จากจุดเด่นของตัวควบคุมทั้งสองแบบมารวมกัน โดยเรียกว่าตัวควบคุมแบบฟuzzy พีดีพีไอ (Fuzzy P²ID) [3]

โดยที่ตัวควบคุมแบบฟuzzy พีดีพีไอ เป็นการประสานการทำงานกันระหว่างตัวควบคุมแบบฟuzzy พีดีกับฟuzzy พีไอ โดยที่ตัวควบคุมทั้งสองแบบจะทำงานขนานกัน โดยมีสิทธิในการเลือกที่จะให้ตัวควบคุมใดทำงาน เมื่อระบบเริ่มทำงานตัวควบคุมแบบฟuzzy พีดีจะเริ่มทำงานก่อน จากนั้นเมื่อถึงจุดที่ได้ออกแบบไว้จะเปลี่ยนให้ตัวควบคุมแบบฟuzzy พีไอทำงานต่อ สำหรับการทำให้ตัวควบคุมทำงานได้นั้น จะต้องมีการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมทั้งสองให้เหมาะสมด้วย ซึ่งการปรับแต่งโดยทั่วไป จะทดลองปรับแบบลองผิดลองถูกและวัดผลที่ได้ทางเอาต์พุต ทำแบบนี้ไปจนกว่าจะได้ผลเป็นที่น่าพอใจ ซึ่งจะใช้เวลาปรับแต่งที่ค่อนข้างนานกว่าจะได้ค่าที่เหมาะสม เนื่องจากตัวควบคุมแบบฟuzzy พีดีพีไอมีค่าพารามิเตอร์ถึงแปดตัว โดยไม่ทราบเลยว่าค่าที่ได้มาจะทำให้ผลของเอาต์พุตที่ได้นั้นดีที่สุดแล้วหรือยัง

จากปัญหาที่เกิดขึ้น จึงได้หาวิธีที่จะมาช่วยในการค้นหาพารามิเตอร์ของตัวควบคุมที่เหมาะสม ผู้วิจัยได้มีการนำแนวความคิดของวิธีการทางพันธุศาสตร์ (genetic algorithm) [4,5,6] เข้ามาประยุกต์ใช้ในการหาค่าเพื่อให้ได้ผลของเอาต์พุตตามที่ผู้ใช้งานต้องการ

งานวิจัยนี้เป็นการนำเสนอวิธีการทางพันธุศาสตร์ เพื่อนำมาประยุกต์ใช้ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟuzzy พีดีพีไอ โดยใช้โปรแกรม Matlab เป็นเครื่องมือในการทดสอบและจำลองผลการออกแบบระบบ

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อให้เป็นวิธีหนึ่งในการหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดี
2. ปรับแต่งค่าพารามิเตอร์เพื่อให้ได้เอาต์พุตเป็นไปตามที่ผู้ใช้งานต้องการด้วยวิธีทางพันธุศาสตร์
3. เพื่อประยุกต์ใช้กับระบบควบคุมแบบอื่นต่อไป

1.3 สมมติฐานของการศึกษา

1. ได้ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอส เพื่อให้ช่วงเวลาไต่ขึ้น (Rise Time) เข้าหาเป้าหมายได้รวดเร็ว
2. ได้ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอส เพื่อให้ช่วงเวลาเข้าที่ (Settle Time) ได้ดี
3. ได้ค่าจุดสลับสวิตช์และค่าเวลาสุ่ม (Sampling Time) ที่เหมาะสม

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

1. ทฤษฎีของตัวควบคุมแบบพีซีพีเอสแควร์ไอดี
2. เทคนิคและวิธีการของการแก้ปัญหาโดยใช้วิธีทางพันธุศาสตร์

1.5 ขอบเขตการวิจัย

1. นำโปรแกรมที่ได้ออกแบบมาทดลองแบบจำลอง โดยใช้โปรแกรม Matlab เป็นเครื่องมือในการเขียนโปรแกรม
2. นำไปให้ทดสอบกับระบบที่ไม่เป็นเชิงเส้น (nonlinear model) และแบบจำลองของหุ่นยนต์เคลื่อนที่ (mobile robot model)
3. หาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีแบบ 8 ตัวแปร คือ $\{K_p, K_d, L_d, K_{u_pd}, K_p, K_i, L_i, K_{u_pi}\}$ เปรียบเทียบกับการหาค่าพารามิเตอร์แบบ 10 ตัวแปร คือ $\{K_p, K_d, L_d, K_{u_pd}, K_p, K_i, L_i, K_{u_pi}, T_s, S_e\}$
4. สรุปรวผลผลกระทบการเพิ่มหรือลดจำนวนประชากรต่อเอาต์พุต และการควบคุม

1.6 ขั้นตอนการศึกษา

1. ค้นคว้า รวบรวม ศึกษาเอกสาร และข้อมูลที่เกี่ยวข้อง
2. ออกแบบและทดสอบโปรแกรมทางพันธุศาสตร์
3. ออกแบบโปรแกรมจำลองของตัวควบคุมแบบพีซีพีสแควร์ไอดี
4. นำโปรแกรมในข้อ 2 และ 3 มารวมกัน
5. ทำการทดลองการหาค่าพารามิเตอร์แบบ 8 ตัวแปรเปรียบเทียบกับผู้ใช้
6. ทำการทดลองการหาค่าพารามิเตอร์แบบ 10 ตัวแปรเปรียบเทียบกับแบบ 8 ตัวแปร
7. สรุปผลวิจารณ์ผลการทดลอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ตัวควบคุมแบบฟัซซีพีเอสแควร์ไอดี

บทนี้กล่าวถึงโครงสร้างของตัวควบคุมแบบฟัซซีพีเอสแควร์ไอดี (Fuzzy P²ID) และเทคนิคการออกแบบซึ่งจะเป็นการประสานการทำงานระหว่างตัวควบคุมแบบฟัซซีพีดี (Fuzzy PD) และตัวควบคุมแบบฟัซซีพีไอ (Fuzzy PI) เนื่องจากตัวควบคุมแบบฟัซซีพีดีให้การตอบสนองได้เร็วกว่าตัวควบคุมแบบฟัซซีพีไอ ในขณะที่ตัวควบคุมแบบฟัซซีพีไอให้ผลที่สถานะอยู่ตัว (steady state) ได้ดีกว่า [7] จึงได้มีการนำข้อดีของตัวควบคุมทั้งสองแบบนี้มาเพื่อให้ผลของเอาต์พุตที่ดีที่สุด ในบทนี้จะแบ่งเนื้อหาออกเป็น 3 ส่วน คือ ตัวควบคุมแบบฟัซซีพีดี ตัวควบคุมแบบฟัซซีพีไอ และตัวควบคุมแบบฟัซซีพีเอสแควร์ไอดี

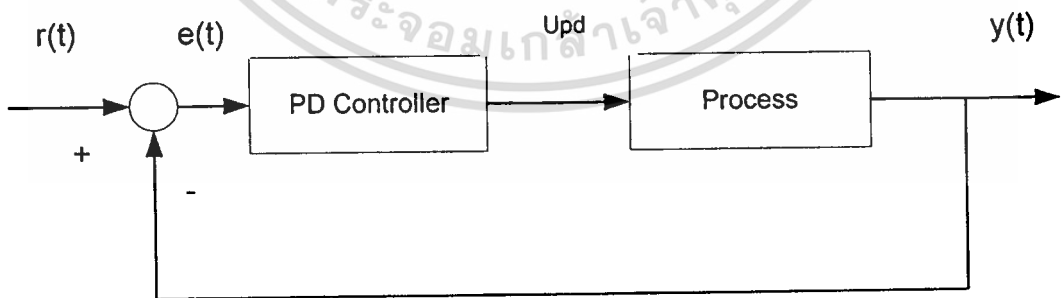
2.1 ตัวควบคุมแบบฟัซซีพีดี

2.1.1 การออกแบบตัวควบคุมแบบฟัซซีพีดี

โครงสร้างพื้นฐานของตัวควบคุมอนาล็อกพีดีแบบดั้งเดิมดังรูปที่ 2.1 จะเขียนให้อยู่ในรูปของโดเมนความถี่ (frequency domain) โดย

$$u_{pd}(s) = (K_p^c + sK_d^c)E(s) \quad (2.1)$$

โดยที่ K_p^c และ K_d^c เป็นค่าเกน และ $E(s)$ เป็นค่าสัญญาณผิดพลาด



รูปที่ 2.1 ระบบควบคุมอนาล็อกพีดีแบบดั้งเดิม

จากสมการที่ (2.1) ใช้การแปลงแบบไบลิเนียร์ $s = \frac{2}{T} \left[\frac{1 - z^{-1}}{1 + z^{-1}} \right]$ เมื่อ $T > 0$ จะได้ (2.1) เป็นแบบไม่ต่อเนื่อง (discrete version) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u_{pd}(z) = \left(K_p + \frac{2K_d}{T} \left[\frac{1-z^{-1}}{1+z^{-1}} \right] \right) E(z)$$

กำหนดให้ $K_p = K_p^c$ และ $K_d = \frac{2}{T} K_d^c$ และทำการแปลงกลับแบบซีได้เป็น

$$u_{pd}(nT) + u_{pd}(nT - T) = K_p[e(nT) + e(nT - T)] + K_d[e(nT) - e(nT - T)] \quad (2.2)$$

หารสมการที่ (2.2) ด้วย T และแทน nT ด้วย n จะได้เป็น

$$\Delta u_{pd}(n) = K_p e_p(n) + K_d e_v(n) \quad (2.3)$$

โดยที่

$$\Delta u_{pd}(n) = \frac{u_{pd}(n) + u_{pd}(n-1)}{T} \quad (2.4)$$

$$e_p(n) = \frac{e(n) - e(n-1)}{T} \quad (2.5)$$

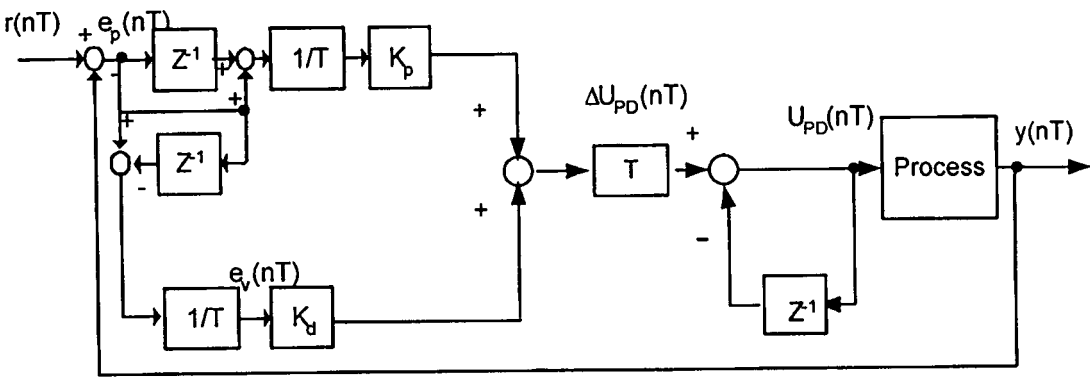
$$e_v(n) = \frac{e(n) + e(n-1)}{T} \quad (2.6)$$

สมการที่ (2.4) เขียนใหม่ได้เป็น

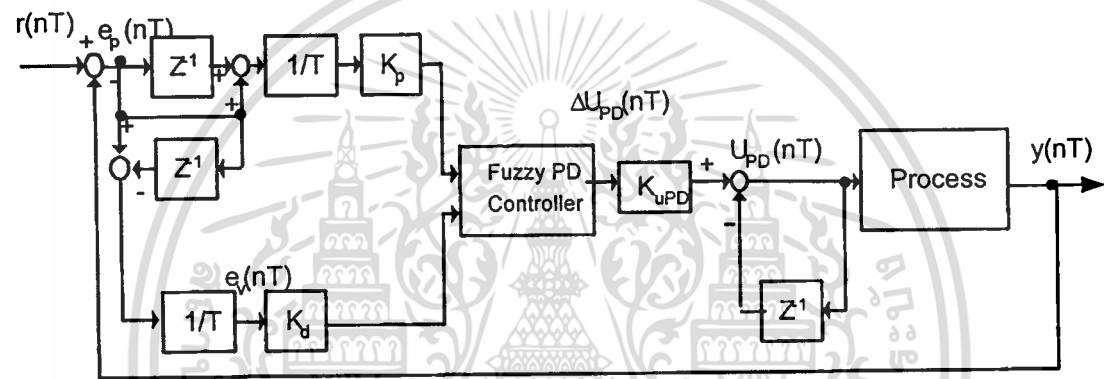
$$u_{pd}(n) = -u_{pd}(n-1) + T\Delta u_{pd}(n) \quad (2.7)$$

แทน $T\Delta u_{pd}(n)$ ด้วยตัวแปรเกนของตัวควบคุมแบบพีซีพีดี $K_{upd} \Delta u_{pd}(n)$ ได้เป็น

$$u_{pd}(n) = -u_{pd}(n-1) + K_{upd} \Delta u_{pd}(n) \quad (2.8)$$



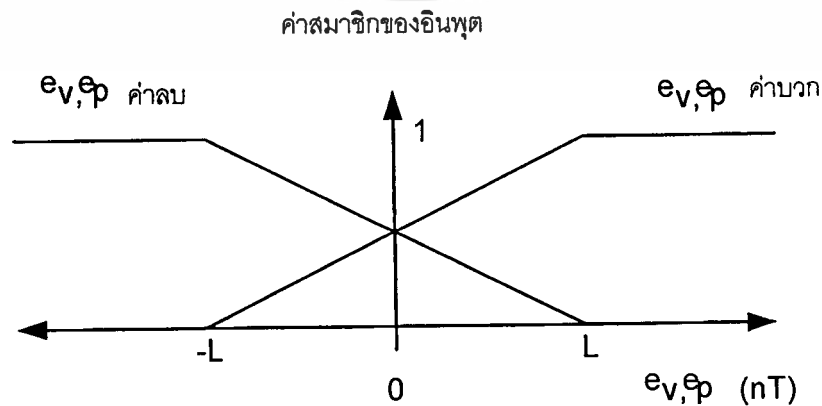
รูปที่ 2.2 ระบบควบคุมดิจิทัลพีดีแบบดั้งเดิม



รูปที่ 2.3 ระบบควบคุมแบบฟัซซีพีดี

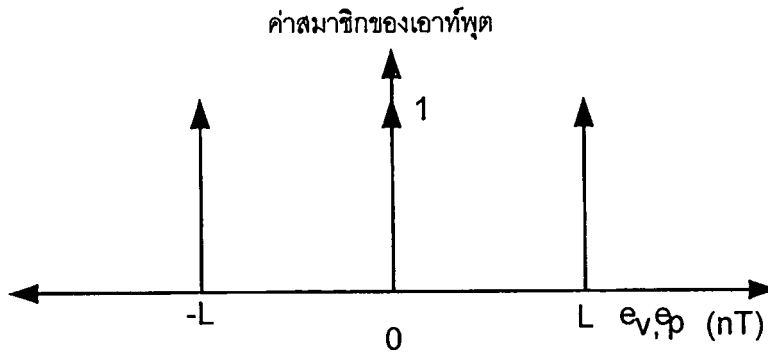
2.1.2. การทำฟัซซีฟิเคชัน

ตัวควบคุมแบบฟัซซีพีดีใช้สองอินพุตประกอบด้วย ค่าเฉลี่ยสัญญาณผิดพลาด ($e_p(nT)$) และค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด ($e_v(nT)$) จากสมการที่ (2.5) และ (2.6) เขียนฟังก์ชันสมาชิกทางอินพุต และฟังก์ชันสมาชิกเอาต์พุตได้ดังรูปที่ 2.4 และ 2.5 ตามลำดับ



เอกสารรูปที่ 2.4 ฟังก์ชันสมาชิกของอินพุตของตัวควบคุมแบบฟัซซีพีดี

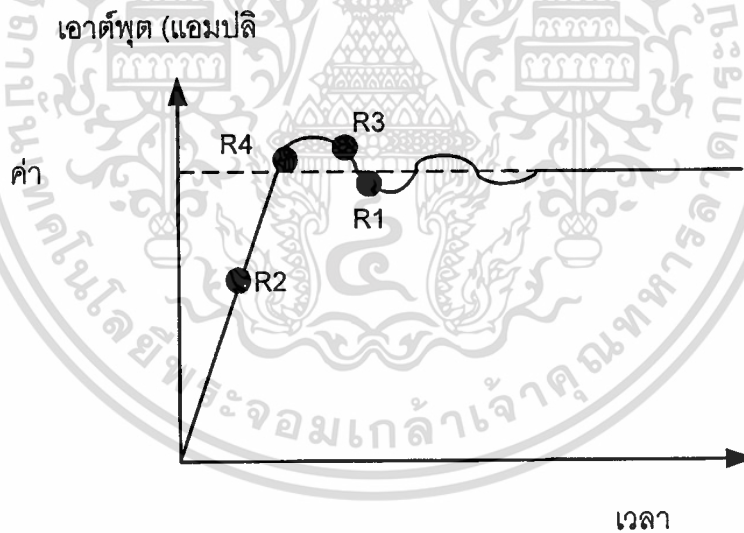
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ฟังก์ชันสมาชิกของเอาต์พุตของตัวควบคุมแบบฟัซซี่พีดี

2.1.3 การกำหนดกฎของฟัซซี่พีดี

การกำหนดกฎของฟัซซี่พีดีนั้น จะขึ้นอยู่กับฟังก์ชันสมาชิกของอินพุตในรูปที่ 2.4 โดยจะกำหนดกฎเป็นไปตามตารางที่ 2.1 และจะเลือกเอาต์พุตจากฟังก์ชันสมาชิกของเอาต์พุตในรูปที่ 2.5 และกำหนดกฎจากรูปที่ 2.6 และจะได้กฎของฟัซซี่พีดีดังตารางที่ 2.1



รูปที่ 2.6 ผลตอบสนองที่ใช้กำหนดกฎของฟัซซี่พีดี

ตารางที่ 2.1 กฎของตัวควบคุมแบบฟัซซี่พีดี

กฎที่	e_p	e_v	เอาต์พุตที่ถูกเลือก
R1	p	p	o.z
R2	p	n	o.p
R3	n	p	o.n
R4	n	n	o.z

จากตารางที่ 2.1 $e_p = r - y$ และ $e_v = 0 - y$ โดย

e_p คือ ค่าสัญญาณผิดพลาด

e_v คือ ค่าอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาด

r คือ ค่าเป้าหมาย

y คือ เอาต์พุต (Δu_{upd})

" $e_p.p$ " คือ ค่าตำแหน่งสัญญาณผิดพลาดที่เป็นบวก และ " $o.p$ " หมายถึงเอาต์พุตที่เป็นบวก จากกฎในตารางที่ 2.1 จะใช้ "AND" หมายถึง ตรรกศาสตร์ซาดเอน (Zadeh's logical "AND")

กำหนดโดย $\mu(A \wedge B) = \min\{\mu_A, \mu_B\}$ จะแบ่งเป็นกฎ 4 ข้อดังนี้

R1 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางบวก และ e_v = ค่าอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาดทางบวก ดังนั้นเอาต์พุตที่ได้เป็น ศูนย์

R2 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางบวก และ e_v = ค่าอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาดทางลบ ดังนั้นเอาต์พุตที่ได้เป็น บวก

R3 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางลบ และ e_v = ค่าอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาดทางบวก ดังนั้นเอาต์พุตที่ได้เป็น ลบ

R4 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางลบ และ e_v = ค่าอัตราการเปลี่ยนแปลงของสัญญาณผิดพลาดทางลบ ดังนั้นเอาต์พุตที่ได้เป็น ศูนย์

2.1.4 การทำดีฟัซซิฟิเคชันของฟัซซีฟิตี

ในขั้นตอนของการทำดีฟัซซิฟิเคชัน จะใช้วิธีศูนย์กลางของมวล (center of mass)

ดังสมการที่ (2.9)

$$\Delta u(n) = \frac{\sum MV_{input} \times MV_{output}}{\sum MV_{input}} \quad (2.9)$$

โดยที่ MV_{input} คือ สมาชิกทางอินพุต และ MV_{output} คือ สมาชิกทางเอาต์พุต

สำหรับตัวควบคุมแบบฟัซซีฟิตีมี 2 อินพุต คือ ค่าสัญญาณผิดพลาด และ ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด จะแบ่งได้ทั้งหมด 20 พื้นที่ และมีพื้นที่ที่มีสมการเหมือนกัน 9 พื้นที่ จึงสามารถแบ่งได้เป็น 9 พื้นที่ดังรูปที่ 2.7

	$K_d e_v(nT)$				
7	4	L 4	8		
5	1	1	3		
-L	2	2	L	$K_p e_p(nT)$	
5	2	2	3		
9	6	-L 6	7		

รูปที่ 2.7 พื้นที่ระนาบเฟสสำหรับสัญญาณผิพลาตที่เป็นไปได้ทั้งหมดของตัวควบคุมพีชชีพีดี

โดยค่า $o.p = L$, $o.n = -L$, $o.z = 0$ และสูตรสมการเส้นตรงจากรูปที่ 2.7 จะได้

$$e_p.p = \frac{K_p d(n) + L}{2L}, e_p.n = -\frac{K_p d(n) + L}{2L}$$

$$e_v.p = \frac{K_p d(n) + L}{2L}, e_v.n = -\frac{K_p d(n) + L}{2L}$$

และจะได้ 9 สมการสำหรับ 9 พื้นที่ดังนี้

$$\Delta u_{pd}(nT) = \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_p |e_p(nT)|]} \quad \text{พื้นที่ 1}$$

$$\Delta u_{pd}(nT) = \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_d |e_v(nT)|]} \quad \text{พื้นที่ 2}$$

$$\Delta u_{pd}(nT) = \frac{1}{2}[-K_d e_v(nT) + L] \quad \text{พื้นที่ 3}$$

$$\Delta u_{pd}(nT) = \frac{1}{2}[K_p e_p(nT) - L] \quad \text{พื้นที่ 4}$$

$$\Delta u_{pd}(nT) = \frac{1}{2}[-K_d e_v(nT) - L] \quad \text{พื้นที่ 5}$$

$$\Delta u_{pd}(nT) = \frac{1}{2}[K_p e_p(nT) + L] \quad \text{พื้นที่ 6}$$

$$\Delta u_{pd}(nT) = 0 \quad \text{พื้นที่ 7}$$

$$\Delta u_{pd}(nT) = -L \quad \text{พื้นที่ 8}$$

$$\Delta u_{pd}(nT) = L \quad \text{พื้นที่ 9}$$

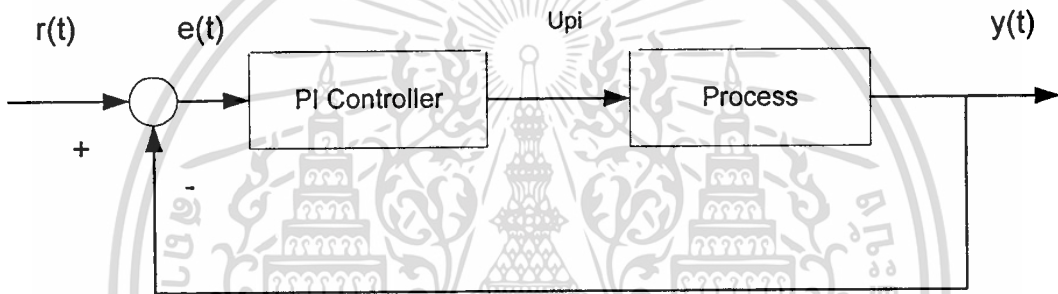
2.2 ตัวควบคุมแบบพีซีพีไอ

2.2.1 การออกแบบตัวควบคุมแบบพีซีพีไอ

โครงสร้างพื้นฐานของตัวควบคุมอนาล็อกพีไอแบบดั้งเดิมดังรูปที่ 2.8 จะเขียนให้อยู่ในรูปของโดเมนความถี่ (frequency domain) โดย

$$u_{pi}(s) = \left(K_p^c + \frac{K_i^c}{s}\right)E(s) \quad (2.10)$$

โดยที่ K_p^c และ K_i^c เป็นค่าเกน(gain) และ $E(s)$ เป็นค่าสัญญาณผิดพลาด



รูปที่ 2.8 ระบบควบคุมอนาล็อกพีไอแบบดั้งเดิม

จากสมการที่ (2.10) ใช้การแปลงแบบไบลีเนีย $s = \frac{2}{T} \left[\frac{1-z^{-1}}{1+z^{-1}} \right]$ เมื่อ $T > 0$ จะได้ (2.10)

เป็นระบบควบคุมแบบไม่ต่อเนื่อง (discrete version) ดังนี้

$$u_{pi}(z) = \left(K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1-z^{-1}} \right) E(z)$$

กำหนดให้ $K_p = K_p^c - \frac{K_i^c T}{2}$ และ $K_d = K_i^c T$ และทำการแปลงกลับแบบซีได้เป็น

$$u_{pi}(nT) - u_{pi}(nT - T) = K_p [e(nT) - e(nT - T)] + K_i T e(nT) \quad (2.11)$$

หารสมการที่ (2.2) ด้วย T

$$u_{pi}(nT) = u_{pi}(nT - T) + T\Delta u_{pi}(nT) \quad (2.12)$$

โดยที่

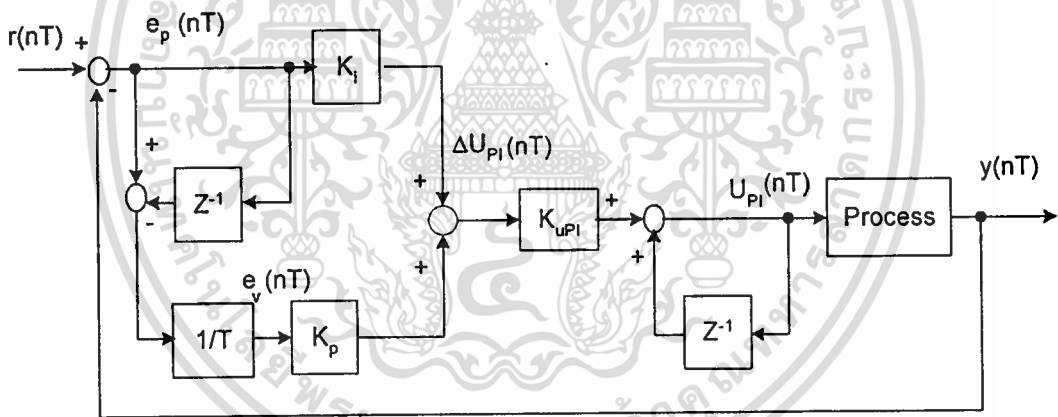
$$\Delta u_{pi}(nT) = K_p e_v(nT) + K_i e_p(nT) \quad (2.13)$$

$$e_v(nT) = \frac{e(nT) - e(nT - T)}{T} \quad (2.14)$$

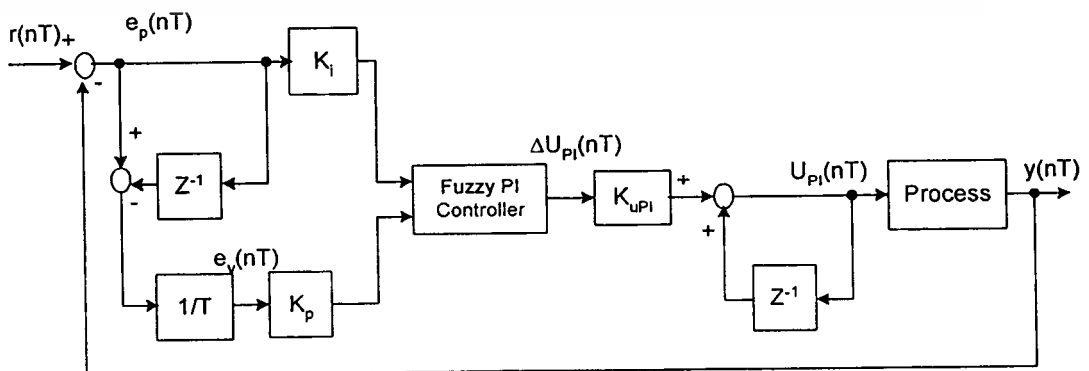
$$e_p(nT) = e(nT) \quad (2.15)$$

แทน $T\Delta u_{pi}(n)$ ด้วยตัวแปรแทนของตัวควบคุมแบบพีซีพีดี $K_{upi}\Delta u_{pi}$ ได้เป็น

$$u_{pi}(nT) = u_{pi}(nT - T) + K_{upi}\Delta u_{pi}(nT) \quad (2.16)$$



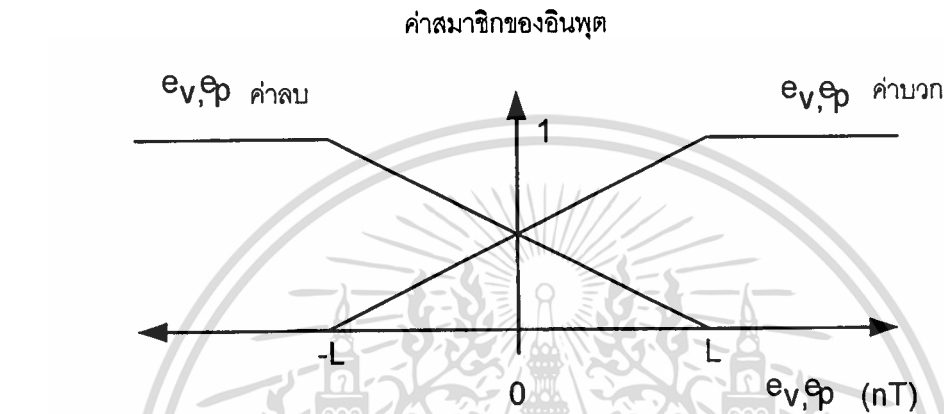
รูปที่ 2.9 ระบบควบคุมดิจิทัลพีไอแบบดั้งเดิม



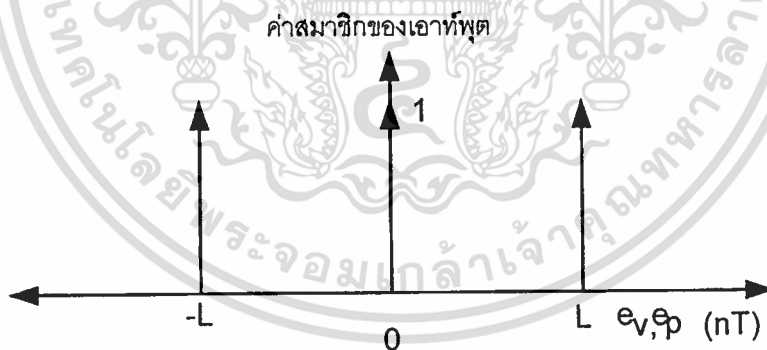
เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยฯ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยฯ จะถือว่าผิดกฎหมายและต้องรับผิดชอบต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2. การทำพีซีพีเคชัน

ตัวควบคุมแบบพีซีพีไอใช้สองอินพุตประกอบด้วย ค่าสัญญาณผิดพลาด ($e_p(nT)$) และค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด ($e_v(nT)$) จากสมการที่ (2.15) และ (2.14) ตามลำดับ เขียนฟังก์ชันสมาชิกทางอินพุต และฟังก์ชันสมาชิกเอาต์พุตได้ดังรูปที่ 2.11 และ 2.12 ตามลำดับ



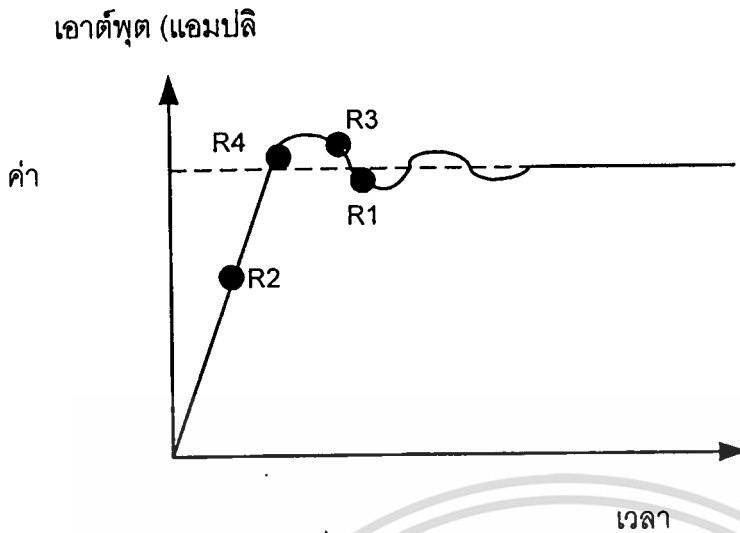
รูปที่ 2.11 ฟังก์ชันสมาชิกของอินพุตของตัวควบคุมแบบพีซีพีไอ



รูปที่ 2.12 ฟังก์ชันสมาชิกของเอาต์พุตของตัวควบคุมแบบพีซีพีไอ

2.2.3 การกำหนดกฎของพีซีพีดี

การกำหนดกฎของพีซีพีไอนั้น จะขึ้นอยู่กับฟังก์ชันสมาชิกของอินพุตในรูปที่ 2.11 โดยจะกำหนดกฎเป็นไปตามตารางที่ 2.2 และจะเลือกเอาต์พุตจากฟังก์ชันสมาชิกของเอาต์พุตในรูปที่ 2.12 และกำหนดกฎจากรูปที่ 2.13 และจะได้กฎของพีซีพีไอดังนี้



รูปที่ 2.13 ผลตอบสนองที่ใช้กำหนดกฎของฟัซซีพีเอ

ตารางที่ 2.2 กฎของตัวควบคุมแบบฟัซซีพีเอ

กฎที่	e_p	e_v	เอาต์พุตที่ถูกเลือก
R1	p	p	o.p
R2	p	n	o.z
R3	n	p	o.z
R4	n	n	o.n

จากตารางที่ 2.1 $e_p = r - y$ และ $e_v = 0 - y$ โดย

e_p คือ ค่าสัญญาณผิดพลาด

e_v คือ ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด

r คือ ค่าเป้าหมาย

y คือ เอาต์พุต (Δu_{upi})

" $e_p.p$ " คือ ค่าตำแหน่งสัญญาณผิดพลาดที่เป็นบวก และ " $o.p$ " หมายถึงเอาต์พุตที่เป็นบวก จาก

กฎในตารางที่ 2.1 จะใช้ "AND" หมายถึง ตรรกศาสตร์ซาดเอน (Zadeh's logical "AND")

กำหนดโดย $\mu(A \wedge B) = \min\{\mu_A, \mu_B\}$ จะแบ่งเป็นกฎเป็น 4 ข้อดังนี้

R1 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางบวก และ e_v = ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาดทางบวก ดังนั้นเอาต์พุตที่ได้เป็น บวก

R2 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางบวก และ e_v = ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาดทางลบ ดังนั้นเอาต์พุตที่ได้เป็น ศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R3 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางลบ และ e_v = ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาดทางบวก ดังนั้นเอาต์พุตที่ได้เป็น ศูนย์

R4 : ถ้า e_p = ค่าสัญญาณผิดพลาดทางลบ และ e_v = ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาดทางลบ ดังนั้นเอาต์พุตที่ได้เป็น ลบ

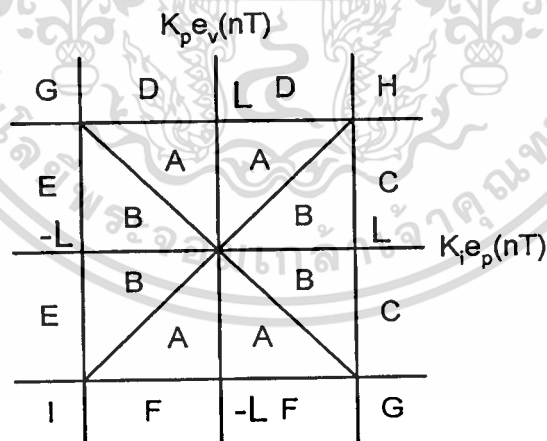
2.2.4 การทำดีฟัซซิฟิเคชันของฟัซซิฟิไอ

ในขั้นตอนของการทำดีฟัซซิฟิเคชัน จะใช้วิธีศูนย์กลางของมวล (center of mass) ดังสมการที่ 2.17

$$\Delta u(n) = \frac{\sum MV_{input} \times MV_{output}}{\sum MV_{input}} \quad (2.17)$$

โดยที่ MV_{input} คือ สมาชิกทางอินพุต และ MV_{output} คือ สมาชิกทางเอาต์พุต

สำหรับตัวควบคุมแบบฟัซซิฟิไอมี 2 อินพุต คือ ค่าสัญญาณผิดพลาด และ ค่าอัตราการเปลี่ยนแปลงของค่าสัญญาณผิดพลาด จะแบ่งได้ทั้งหมด 20 พื้นที่ และมีพื้นที่ที่มีสมการเหมือนกัน 9 พื้นที่ จึงสามารถแบ่งได้เป็น 9 พื้นที่ดังรูปที่ 2.14



รูปที่ 2.14 พื้นที่ระนาบเฟสของสัญญาณผิดพลาดที่เป็นไปได้ทั้งหมดของตัวควบคุมฟัซซิฟิไอ

โดยค่า $o.p = L$, $o.n = -L$, $o.z = 0$ และสูตรสมการเส้นตรงจากรูปที่ 2.14 จะได้

$$e_p.p = \frac{K_i e_p(nT) + L}{2L}, e_p.n = \frac{-K_i e_p(n) + L}{2L}$$

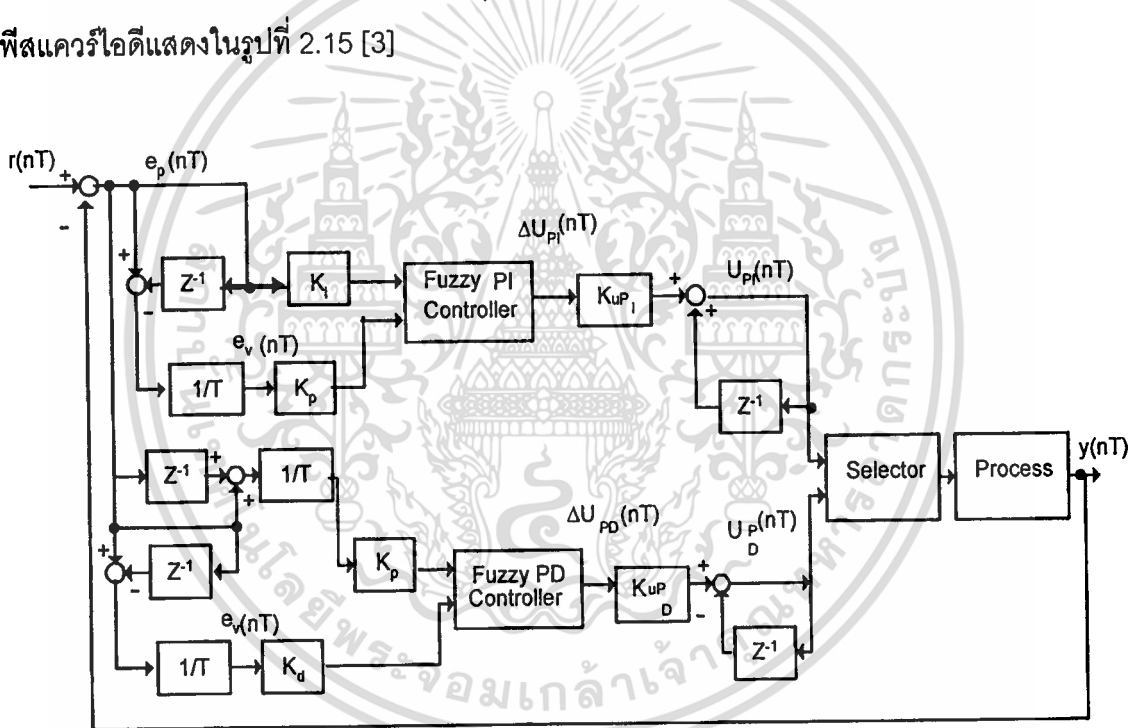
$$e_v.p = \frac{K_p e_v(n) + L}{2L}, e_v.n = \frac{-K_p e_v(n) + L}{2L}$$

และจะได้ 9 สมการสำหรับ 9 พื้นที่ดังนี้

$\Delta u_{pi}(nT) = \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i e_p(nT)]}$	พื้นที่ A
$\Delta u_{pi}(nT) = \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i e_v(nT)]}$	พื้นที่ B
$\Delta u_{pi}(nT) = \frac{1}{2}[K_p e_v(nT) + L]$	พื้นที่ C
$\Delta u_{pi}(nT) = \frac{1}{2}[K_i e_p(nT) + L]$	พื้นที่ D
$\Delta u_{pi}(nT) = \frac{1}{2}[K_p e_v(nT) - L]$	พื้นที่ E
$\Delta u_{pi}(nT) = \frac{1}{2}[K_i e_p(nT) - L]$	พื้นที่ F
$\Delta u_{pi}(nT) = 0$	พื้นที่ G
$\Delta u_{pi}(nT) = L$	พื้นที่ H
$\Delta u_{pi}(nT) = -L$	พื้นที่ I

2.3 ตัวควบคุมแบบฟัซซีฟัสแควร์ไอดี

ตัวควบคุมแบบฟัซซีฟัสแควร์ไอดี เป็นการทำงานร่วมกันระหว่างตัวควบคุมแบบฟัซซีพีไอ และตัวควบคุมแบบฟัซซีพีดี ซึ่งจะมีตัวเลือกทำหน้าที่ในเลือกการทำงาน โดยเริ่มแรกตัวควบคุมแบบฟัซซีพีดีจะทำงานเริ่มต้นกระบวนการเนื่องจากตัวควบคุมแบบฟัซซีพีดีให้การตอบสนองได้เร็วกว่าตัวควบคุมแบบฟัซซีพีไอ แล้วจะสลับการทำงานเป็นตัวควบคุมแบบฟัซซีพีไอ เนื่องจากตัวควบคุมแบบฟัซซีพีไอให้ผลที่สถานะอยู่ตัว (steady state) ดีกว่า เมื่อกระบวนการได้มาถึงจุดที่กำหนดไว้ในย่านสถานะเสถียร ถ้าหากเอาต์พุตของกระบวนการออกไปจากบริเวณสถานะเสถียร ตัวควบคุมแบบฟัซซีพีไอจะหยุดการทำงาน คือ ต่ำกว่า 90 เปอร์เซ็นต์ของค่าเป้าหมาย ตัวควบคุมแบบฟัซซีพีดีจะทำงานอีกครั้งหนึ่ง ด้วยเหตุนี้จะเรียกกระบวนการควบคุมนี้ว่า ตัวควบคุมแบบฟัซซีฟัสแควร์ไอดีแสดงในรูปที่ 2.15 [3]



รูปที่ 2.15 ระบบควบคุมแบบฟัซซีฟัสแควร์ไอดี

สำหรับการสร้างตัวควบคุมแบบฟัซซีฟัสแควร์ไอดีตัวเลือก (Selector) ที่แสดงในรูปที่ 2.15 เป็นเซตของกฎทางตรรกะสำหรับการทำงานของตัวควบคุม และจะได้มีการประกันความมีเสถียรภาพได้จาก

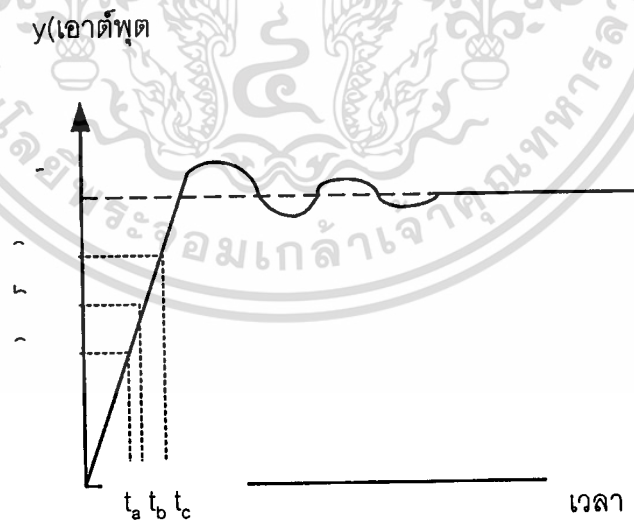
1. ทฤษฎีเกนน้อย (small gain theorem) เสถียรภาพของตัวควบคุมฟัซซีพีไอและฟัซซีพีดีรับประกันโดย [1] และ [8] ตามลำดับ

2. ความมีเสถียรภาพของระบบควบคุมทั้งหมดขึ้นอยู่กับเสถียรภาพของตัวเลือก โดยจะเขียนแบบจำลองทางคณิตศาสตร์สำหรับระบบควบคุมการสลับแบบดิจิทัล ได้ดังสมการที่ (2.18) ไม่ว่าการันตีทางทฤษฎี นอกเหนือจากนี้ การเลือกและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

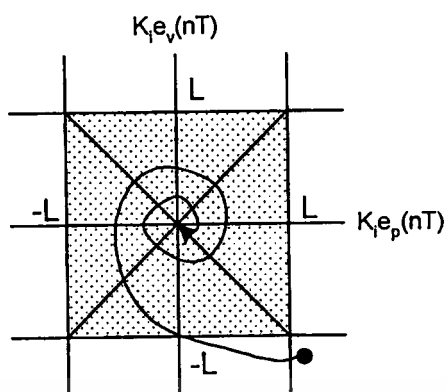
$$x(nT + T) = Gx(nT) + H \sum_{i=1}^2 u_i(nT) \delta_i(nT) \quad (2.18)$$

ซึ่ง $x(nT)$ เป็นเวกเตอร์สถานะ (State Vector) ที่เวลาไม่ต่อเนื่อง (Discrete Time) G เป็นเมตริกซ์ระบบ (System Matrix) และ H เป็นเมตริกซ์อินพุต (Input Matrix) สำหรับระบบเวลาไม่ต่อเนื่อง โดย u เป็นค่าอินพุตควบคุม ซึ่ง $i = 1$ แทนการทำงานของตัวควบคุมแบบพีชชีพีดี และ $i = 2$ แทนการทำงานของตัวควบคุมแบบพีชชีพีไอ $\delta_i(nT)$ เป็นตัวแปรทางตรรกะซึ่งจะมีค่าเท่ากับ 1 เมื่อระบบทำงาน และเท่ากับ 0 หากหยุดทำงาน และจะพบว่าทุกคาบเวลา nT มีเพียงดัชนีเดียวที่ถูกใช้งาน โดยอาจจะใช้สมการที่ (2.18) มาวิเคราะห์เสถียรภาพแบบ (Bounded-Input Bounded-Output, BIBO) ใน [8]

การออกแบบโดยใช้กราฟนำมาใช้ในการออกแบบกฎสลับการควบคุม (Switching Rules) โดยเริ่มจากการพิจารณาวิถีโคจรของเอาต์พุตกับเวลาดังแสดงในรูปที่ 2.16 โดยที่ r เป็นจุดค่าเป้าหมาย a , b , และ c แทนเอาต์พุตที่เวลา t_a , t_b และ t_c ตามลำดับ และเมื่อดูที่วิถีโคจรของค่าความผิดพลาดบนระนาบเฟส (error-phase plane) ที่แสดงในรูป 2.16 จากรูปจะเห็นว่าจำนวนของวงกลมรอบจุดศูนย์กลางของแผนภาพสอดคล้องกับระยะเวลาของสภาวะชั่วคราวคลื่นไซน์ (Transient Sine Wave) เมื่อวิถีเอาต์พุตของระบบในรูป 2.16 มาถึงสภาวะเสถียร วิถีค่าความผิดพลาดจะตกอยู่ในจุดศูนย์กลางของระนาบเฟส (Phase Plane) แสดงในรูป 2.17



รูปที่ 2.16 ผลตอบสนองของเอาต์พุตเทียบกับเวลา



รูปที่ 2.17 วิธีค่าความผิดพลาดบนระนาบเฟส

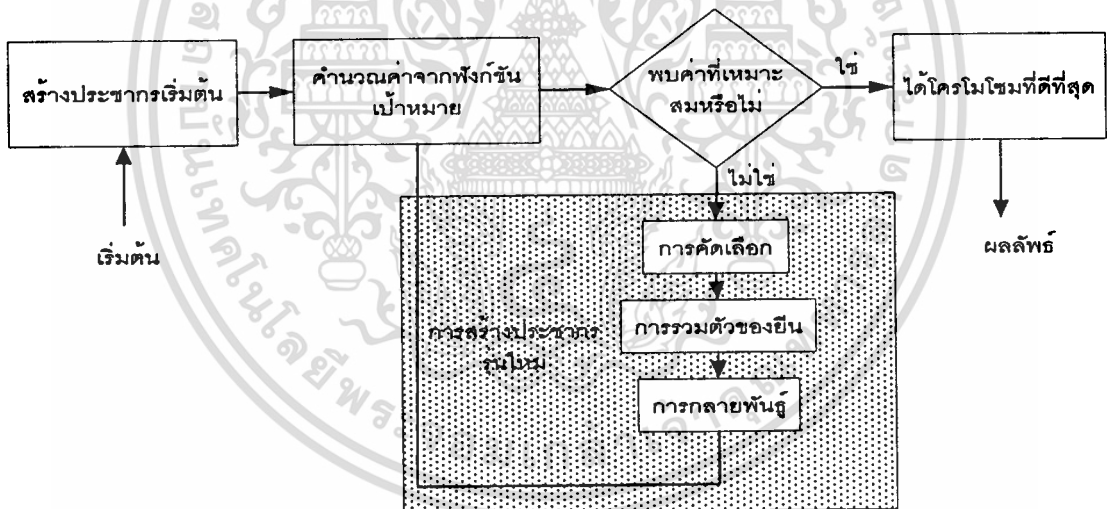
จากรูปที่ 2.16 จะเห็นว่าสามารถเลือกเวลาการสลับได้ตามต้องการ โดยอาจจะใช้เวลาสลับที่ 90/10 โดยให้ตัวควบคุมแบบพีซีพีดีทำงานไปจนถึง 90% ของค่าเป้าหมาย และให้ตัวควบคุมแบบพีซีพีไอทำงานใน 10% ที่เหลือ เพราะช่วงเวลาที่ 10% ถึง 90% ของค่าเป้าหมายเป็นช่วงของเวลาไคขึ้น (Rise Time) เนื่องจากตัวควบคุมแบบพีซีพีดี ให้การตอบสนองที่เร็ว จึงเหมาะที่จะนำมาใช้ในช่วงนี้หลังจากที่เข้าใกล้ 90% ของค่าเป้าหมายแล้วต้องการวิธีที่เข้าเป้าหมายดีที่สุด ดังนั้นจึงใช้คุณสมบัติของตัวควบคุมแบบพีไอจึงทำงานต่อในช่วง 10% ที่เหลือแต่อาจจะเลื่อนจุดได้ตามความเหมาะสม ทั้งหมดนี้เป็นหลักการพื้นฐานของตัวควบคุมแบบพีซีพีสแควร์ไอดีต่อไปจะเป็นการกล่าวในเรื่องวิธีการทางพันธุศาสตร์ที่จะใช้ในการหาจุดสลับและปรับตั้งค่าพารามิเตอร์ในการควบคุม

บทที่ 3

วิธีการทางพันธุศาสตร์

วิธีการทางพันธุศาสตร์ (Genetic Algorithms) Halland [9] เป็นคนแรกที่น่าเสนอ โดยเลียนแบบการวิวัฒนาการทางธรรมชาติ โดยขั้นตอนวิวัฒนาการประยุกต์ใช้ในการแก้ปัญหาได้อย่างมีประสิทธิภาพ โดยความสำคัญนั้นอยู่ที่ประชากรลูกหลานที่อยู่รอดจะเป็นประชากรที่ดี และเหมาะสมกับปัญหานั้น ๆ ประชากรแต่ละรุ่นจะถูกสร้างโดย กระบวนการคัดเลือกประชากร (population) แต่ละตัวที่มีระดับของความเหมาะสมจากขอบเขตของปัญหา และนำมาผสมกันโดยใช้ตัวดำเนินการทางพันธุศาสตร์ (Genetic Operation) โดยคำตอบที่ดีที่สุดจะพบจากประชากรตัวสุดท้ายที่เหลือรอดอยู่

กระบวนการทางพันธุศาสตร์จะประกอบด้วย การคัดเลือก (Selection) การผสม (Recombination) การกลายพันธุ์ (Mutation) และการแทนที่ (Reinsertion)



รูปที่ 3.1 โครงสร้างกระบวนการทางพันธุศาสตร์

จากรูปที่ 3.1 แสดงโครงสร้างของกระบวนการทางพันธุศาสตร์แบบพื้นฐาน ขั้นตอนแรกเริ่มจากการสร้างประชากรเริ่มต้นโดยการสุ่ม ขนาดของประชากรขึ้นอยู่กับปัญหาที่แก้ไข ต่อมาเป็นการคำนวณประชากรแต่ละตัวด้วยฟังก์ชันวัตถุประสงค์ (Objective Function) หรือฟังก์ชันเป้าหมาย ไม่มีการกำหนดตายตัวขึ้นอยู่กับลักษณะของปัญหาที่จะแก้ไข เมื่อผ่านขั้นตอนนี้เป็นการตรวจสอบ ถ้าไม่พบคำตอบที่เหมาะสมจะเริ่มสร้างประชากรรุ่นลูกหลาน โดยนำประชากรรุ่นพ่อแม่ (parent) มาคัดเลือกซึ่งดูจากค่าของฟิตเนส (Fitness) ประชากรพ่อแม่ที่ถูกเลือกนำมาผสมกัน หลังจากนั้นไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ประชากรรุ่นลูกหลาน (Offspring) ประชากรรุ่นลูกหลานทั้งหมดจะถูกทำให้กลายพันธุ์ (Mutated) ด้วยค่าความน่าจะเป็น แล้วจึงคำนวณค่าของประชากรลูกหลานแต่ละตัว แล้วนำประชากรรุ่นลูกหลานที่ดี 3-5 ตัว เข้าไปแทนในประชากรรุ่นพ่อแม่เพื่อนำมาเป็นประชากรในรุ่นใหม่ต่อไป และจะวนรอบเช่นนี้ไปจนกว่าจะพบคำตอบที่เหมาะสม

ในบทนี้กล่าวถึงกระบวนการทางพันธุศาสตร์ซึ่งประกอบด้วย การแทนที่ประชากร การคัดเลือก การรวมตัวของยีน การกลายพันธุ์ ฟังก์ชันเป้าหมาย และฟังก์ชันฟิตเนสอันเป็นพื้นฐานในการเลือกค่าที่เหมาะสมสำหรับพารามิเตอร์ที่ใช้ในการควบคุมในบทต่อไป

3.1 การแทนที่ประชากร (Population Representation)

ก่อนที่จะมีการดำเนินการทางพันธุศาสตร์ โดยจะต้องมีการเข้ารหัสของพารามิเตอร์ในประชากรเสียก่อน โดยประชากรที่สร้างนั้นจะอยู่ในช่วง 30 ถึง 50 ตัว แต่บางครั้งวิธีการของไมโครพันธุศาสตร์ จะใช้ประชากรที่น้อยคือประมาณ 10 ตัว และได้ใช้ความสามารถของการผลิต (Reproduction) และการแทนที่ (Replacement) เพื่อให้สามารถใช้กับเวลาจริง (real-time) [10] ได้

การแทนที่ประชากรหรือโครโมโซมของวิธีทางพันธุศาสตร์นั้น จะเป็นการเข้ารหัสของแต่ละตัวแปรหรือยีน (Gene) ด้วยรหัสแบบไบนารี (Binary Code) และจะมีการเชื่อมตัวแปรแต่ละตัวเข้าเป็นชุดเดียวกันเรียกว่า โครโมโซม (Chromosome) ต่อมาได้มีงานวิจัยได้ทำการทดลองพบว่าการเข้ารหัสแบบเกรย์ (Gray Code) ได้ผลของดีกว่าการเข้ารหัสแบบไบนารี

ในขณะที่มีการใช้การเข้ารหัสแบบไบนารีกันอย่างแพร่หลายนั้น ได้มีอีกทางเลือกหนึ่งที่น่าสนใจ คือการเข้ารหัสแบบค่าจริง (Real-value) ซึ่งการแทนด้วยค่าจริงนี้ได้ถูกนำเสนอโดย Wright [11] โดยข้อดีของการเข้ารหัสด้วยค่าจริงคือ ไม่จำเป็นต้องทำการแปลงจากโครโมโซมเป็นฟีโนไทป์ (Phenotype) ก่อนการคำนวณค่าจากฟังก์ชันเป้าหมาย ต้องการหน่วยความจำน้อย และสามารถนำไปใช้กับคอมพิวเตอร์ได้โดยตรง อีกทั้งยังมีอิสระในการใช้ตัวดำเนินการทางพันธุศาสตร์ (Genetic Operation) ที่แตกต่างกัน

จากวิธีที่กล่าวมาทั้งหมดนี้ การเลือกใช้วิธีการจะขึ้นอยู่กับลักษณะปัญหา เพราะการเลือกวิธีที่เหมาะสมจะช่วยเพิ่มประสิทธิภาพในการทำงานของวิธีทางพันธุศาสตร์

3.2 ฟังก์ชันเป้าหมายและฟังก์ชันฟิตเนส (The Objective and Fitness Function)

ฟังก์ชันเป้าหมายนั้นกำหนดโดยผู้ใช้ตามปัญหาที่จะแก้ ฟังก์ชันเป้าหมายเป็นตัววัดประสิทธิภาพของประชากรแต่ละตัว ในการหาค่าต่ำสุดหรือค่าสูงสุดของปัญหา บางครั้งค่าของฟังก์ชันเป้าหมายอาจมีค่าเป็นลบ ฟังก์ชันฟิตเนสเป็นตัววัดสถานะของประชากรต้องมีค่าบวกเสมอ ดังนั้นถ้าฟังก์ชันเป้าหมายมีค่าเป็นลบจำเป็นต้องเปลี่ยนให้เป็นฟังก์ชันฟิตเนสจากสมการที่ (3.1)

$$F(x) = g(f(x)) \quad (3.1)$$

ซึ่ง $f(x)$ คือ ฟังก์ชันเป้าหมาย g คือ การเปลี่ยนแปลงค่าของฟังก์ชันเป้าหมายเพื่อไม่ให้ค่าเป็นลบ และ $F(x)$ คือ ค่าของฟิตเนสที่ได้ ส่วนมากค่าของฟังก์ชันฟิตเนสจะสอดคล้องกับจำนวนประชากรลูกหลานที่คาดหวังในการผลิตในรุ่นถัดไป จากสมการที่ (3.2) $F(x_i)$ คือ ค่าของฟิตเนส $f(x_i)$ คือ ค่าประสิทธิภาพที่คำนวณได้จากประชากรแต่ละตัว

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N_{ind}} f(x_i)} \quad (3.2)$$

ซึ่ง N_{ind} คือ จำนวนประชากรทั้งหมด และ x_i คือ ค่าของประชากร

ในขณะที่ความเหมาะสมของประชากรแต่ละตัวรับประกันได้จากการถูกเลือก ด้วยความน่าจะเป็น แต่อาจจะเกิดความผิดพลาดจากค่าฟังก์ชันเป้าหมายของประชากรที่มีค่าเป็นลบ และอาจทำให้ลู่เข้าก่อนกำหนด เนื่องจากการยึดครองพื้นที่ของประชากรที่มีค่าฟิตเนสที่สูง ทำให้ประชากรที่มีค่าฟิตเนสต่ำไม่มีโอกาสที่จะถูกเลือก จากปัญหาที่เกิดขึ้นนี้จะแก้ได้ โดยการปรับค่าฟิตเนสซึ่งการปรับค่าฟิตเนสมี 2 วิธี คือ การทำมาตรวัด (scaling) และการกำหนดตำแหน่ง (ranking)

การทำมาตรวัดจะเป็นการแปลงค่าฟิตเนสใหม่โดยใช้สมการเส้นตรงดังสมการที่ (3.3)

$$F(x) = af(x) + b \quad (3.3)$$

โดยที่ $f(x)$ คือค่าของฟิตเนสเก่า $F(x)$ คือ ค่าของฟิตเนสใหม่ a คือ ค่าของแฟคเตอร์สเกล ที่มีค่าเป็นบวก และ b เป็นค่าขดเซย (offset) ที่รับรองว่าค่าของฟิตเนสใหม่มีค่าเป็นบวก ส่วนการกำหนดตำแหน่งนั้น เป็นการเรียงค่าจากฟังก์ชันเป้าหมาย ประชากรที่มีค่าฟิตเนสน้อย

เอกสารนี้เป็นเอกสารตัวอย่าง ไม่สามารถนำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สุดจะอยู่ในลำดับที่ 1 และจะเรียงลำดับต่อไปจนถึงประชากรที่มีค่าฟิตเนสมากที่สุดจะอยู่ลำดับสุดท้ายจากสมการที่ (3.4)

$$Fitness(pos) = 2 - SP + 2(SP - 1) \frac{Pos - 1}{N_{ind} - 1} \quad (3.4)$$

โดย SP คือค่าความลำเอียง (bias) หรือค่าของซีเล็คทีฟ เพเชอร์ (Selective Pressure) Pos คือ ลำดับของประชากรตัวที่ i และ N_{ind} คือจำนวนของประชากรทั้งหมด

3.3 การคัดเลือก (Selection)

การคัดเลือกเป็นขั้นตอนที่จำลองการคัดเลือกทางธรรมชาติ โดยจะคัดเลือกประชากรรุ่นพ่อแม่เพื่อเป็นประชากรต้นแบบในบ่อเพาะพันธุ์ (Mating Pool) เพื่อใช้ในการสร้างประชากรรุ่นลูกหลานต่อไป การคัดเลือกของวิวิธพันธุ์ศาสตร์นั้น จะใช้ค่าความเหมาะสมเป็นตัวตัดสินว่าประชากรมีโอกาสถูกเลือกเป็นประชากรพ่อแม่มากน้อยเพียงใด ประชากรที่มีค่าความเหมาะสมที่ดี จะถูกกำหนดค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้ง จะสูงตามไปด้วยจากสมการที่ 3.5

$$Prob_select = \frac{F_i}{\sum Fitness} \quad (3.5)$$

โดย F_i คือ ค่าของฟิตเนสของประชากรแต่ละตัว และ $\sum Fitness$ คือ ผลรวมของค่าฟิตเนสทั้งหมดที่นำมาคัดเลือกในบ่อเพาะพันธุ์ ซึ่งมีวิธีการคัดเลือกหลายวิธีด้วยกันประสิทธิภาพจะขึ้นอยู่กับ ตัววัดประสิทธิภาพของวิธีการทั้งหมดคือ

Selective pressure คือ ค่าความน่าจะเป็นของประชากรที่ดีที่สุดที่ถูกเลือก โดยเปรียบเทียบกับค่าเฉลี่ยของความน่าจะเป็นของประชากรที่ถูกเลือกทั้งหมด

ความลำเอียง (Bias) คือ ค่าสัมบูรณ์ของความแตกต่างระหว่างค่าจริงกับความน่าจะเป็นที่คาดหวังไว้สำหรับการคัดเลือกของสมาชิกแต่ละตัว

การกระจาย (spread) คือ ช่วงของค่าความเป็นไปได้ที่ประชากรตัวหนึ่งจะถูกเลือก

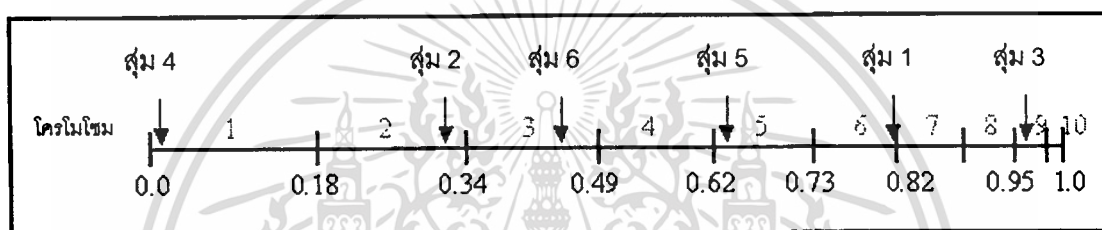
Loss of Diversity คือ สัดส่วนของประชากรตัวหนึ่ง ซึ่งไม่ถูกเลือกในระหว่างขั้นตอนของการคัดเลือก

ความหนาแน่นของการคัดเลือก (Selection Intensity) คือ ค่าเฉลี่ยของค่าฟิตเนสของประชากรหลังจากใช้วิธีการคัดเลือกแบบการกระจายตัวของ Gaussian

ความแปรปรวนของการคัดเลือก (Selection Variance) คือ ความแปรปรวนของการกระจายตัวของค่าฟิตเนสของประชากร หลังจากใช้วิธีการคัดเลือกแบบการกระจายตัวแบบ Gaussian

3.3.1 การคัดเลือกแบบวงล้อรูเลต (roulette wheel)

วิธีการคัดเลือกแบบวงล้อรูเลตอย่างง่ายเรียกว่า การเลือกแบบสุ่มเลือกโดยมีการแทนที่ (Stochastic Sampling with Replacement : SSR) [12] วิธีการคัดเลือกแบบนี้เป็นการกำหนดขนาดแต่ละช่องของวงล้อขึ้นอยู่กับค่าฟิตเนส เส้นรอบวงของวงล้อจะมีค่าความน่าจะเป็นเท่ากับ 1 จำนวนครั้งของการสุ่มจะขึ้นอยู่กับจำนวนของประชากรรุ่นพ่อแม่ที่ต้องการ จะแสดงดังรูปที่ 3.2



รูปที่ 3.2 การคัดเลือกแบบวงล้อรูเลต

จากรูปที่ 3.2 นำมาเขียนได้จากตารางที่ 3.1 โดยตารางจะแสดงถึงค่าความน่าจะเป็นของการถูกเลือก สำหรับ 11 โครโมโซม ลิเนีย แรงคิง และ ซีเล็กทีฟเพเซอร์ มีค่าเท่ากับสอง โครโมโซมที่ 1 มีค่าฟิตเนสมากที่สุดและมีช่วงบนเส้นกว้างที่สุด ในขณะที่เดียวกันโครโมโซมที่ 10 มีค่าฟิตเนสน้อยที่สุด ดังนั้นจึงมีช่วงบนเส้นแคบที่สุด แสดงในรูปที่ 3.2

ตารางที่ 3.1 ความน่าจะเป็นของการคัดเลือกและค่าของฟิตเนส

จำนวนโครโมโซม	1	2	3	4	5	6	7	8	9	10	11
ค่าฟิตเนส	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
ความน่าจะเป็นของการคัดเลือก	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

เมื่อทำการเลือกโดยการสุ่มแล้วจะนำไปเก็บรวมกันใน แมททิง ปอปปูเรชัน (Mating Population) โดยจะสุ่มอยู่ในช่วง 0.0 ถึง 1 ตัวอย่างการสุ่ม 6 ครั้งจะได้ผลดังนี้ 0.81, 0.32, 0.96, 0.01, 0.65, 0.42 หลังจากนั้นใน แมททิง ปอปปูเรชัน จะประกอบด้วยโครโมโซมที่ 1 2 3 5 6 และ 9

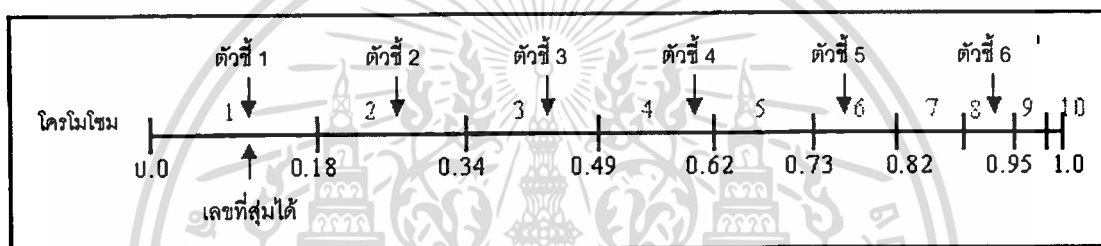
วิธีการคัดเลือกแบบวงล้อรูเลตนี้จะให้ผลของความลำเอียงเป็น 0 แต่จะไม่รับประกันการกระจายว่าจะได้ค่าน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม (Stochastic Universal Sampling)

วิธีการคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่มจะเป็นการแก้ปัญหาของวิธีการคัดเลือกแบบวงล้อรูเลท ซึ่งจะทำให้ได้ค่าความลำเอียงเท่ากับ 0 และจะได้ค่าของการกระจายน้อยที่สุด ดังนั้นแต่ละช่องของโครโมโซมจะเหมือนกันกับวิธีการคัดเลือกแบบวงล้อรูเลท โดยจะเพิ่มตัวชี้ (Pointer) ที่มีจำนวนเท่ากับจำนวนของโครโมโซมที่ต้องการเลือก คือ N-pointer ซึ่งระยะห่างของตัวชี้จะเท่ากับ $1/N$ -pointer โดยตำแหน่งแรกของตัวชี้จะกำหนดจากการสุ่มในช่วง $[0, 1/N$ -pointer] ตัวอย่างการเลือก 6 โครโมโซม ระยะห่างของตัวชี้จะมีค่าเท่ากับ $1/6$ หรือ 0.167 จากรูปที่ 3.3 แสดงตัวอย่างของการคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม

ตัวอย่าง การสุ่ม 1 ครั้งในช่วง $[0, 0.167]$ ได้ค่าเท่ากับ 0.1



รูปที่ 3.3 วิธีการคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม

หลังจากนั้นใน แมททิง ปอปปูเรชัน จะประกอบด้วยโครโมโซมที่ 1 2 3 4 6 และ 8

3.4 การรวมตัวของยีน (Recombination)

การรวมตัวของยีนหรือการครอสโอเวอร์ (Crossover) เป็นการนำประชากรที่ได้จากการคัดเลือกที่อยู่ในป๊อปูเอชันพันธุ์ มาจับคู่แลกเปลี่ยนยีนของประชากรพ่อแม่ ตามการกำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover : P_c) เพื่อสร้างประชากรรุ่นใหม่หรือประชากรลูกหลาน โดยการกระทำนี้จะแบ่งเป็น 2 ประเภท คือ การรวมตัวของยีนโดยใช้ค่าจริง และการรวมตัวของยีนโดยใช้ค่าไบนารี

3.4.1 การรวมตัวของยีนโดยใช้ค่าจริง

3.4.1.1 การรวมตัวของยีนแบบไม่ต่อเนื่อง (Discrete Recombination)

การรวมตัวของยีนแบบไม่ต่อเนื่อง เป็นการเปลี่ยนค่ายีนหรือค่าตัวแปรที่เป็นค่าจำนวนจริง ระหว่างประชากรรุ่นพ่อแม่ ประชากรรุ่นพ่อแม่สองตัวที่อยู่ด้านล่างนี้ แต่ละตัวจะมี 3 ตัวแปร โดยกระทำกันระหว่างค่าตัวแปรที่อยู่หลักเดียวกันเท่านั้น และจะใช้ตัวแปรตัวไหนนั้นขึ้นอยู่กับค่าความน่าจะเป็นที่สุ่มได้ ดังตัวอย่าง

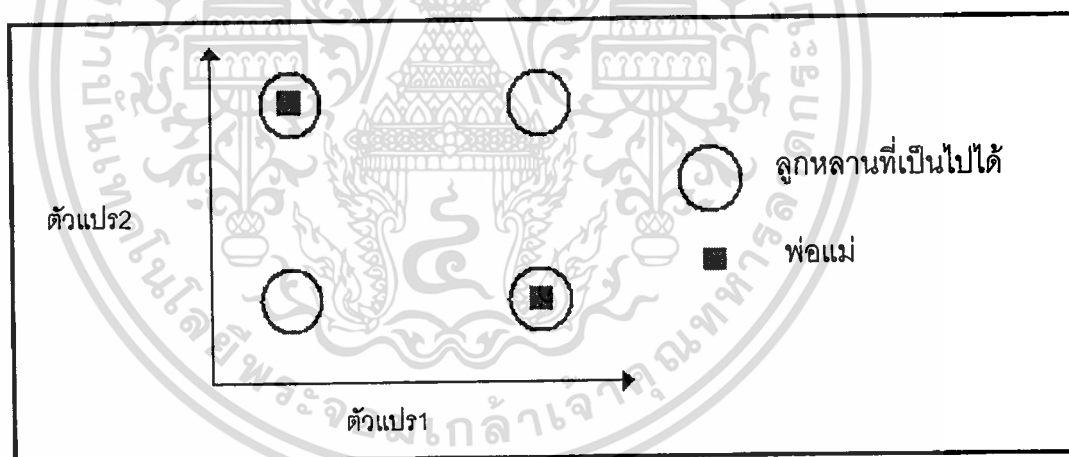
ประชากรพ่อแม่ 1	12	25	5
ประชากรพ่อแม่ 2	123	4	34

สำหรับแต่ละตัวแปรของประชากรพ่อแม่ ซึ่งมีส่วนทำให้เกิดประชากรลูกหลาน จะถูกสุ่มจากค่าความน่าจะเป็น

ตัวอย่าง 1	2	2	1
ตัวอย่าง 2	1	2	1

หลังจากการครอสโอเวอร์แล้วจะได้ประชากรรุ่นลูกหลานดังนี้

ประชากรลูกหลาน 1	123	4	5
ประชากรลูกหลาน 2	12	4	5



รูปที่ 3.4 ตำแหน่งที่เป็นไปได้ที่จะเกิดประชากรลูกหลานหลังจากการรวมยีน

การรวมตัวของยีนแบบไม่ต่อเนื่อง จะสร้างมุมโดยกำหนดจากประชากรพ่อแม่ และการรวมตัวแบบนี้สามารถใช้กับตัวแปรที่เป็นค่าจริงและค่าไบนารี

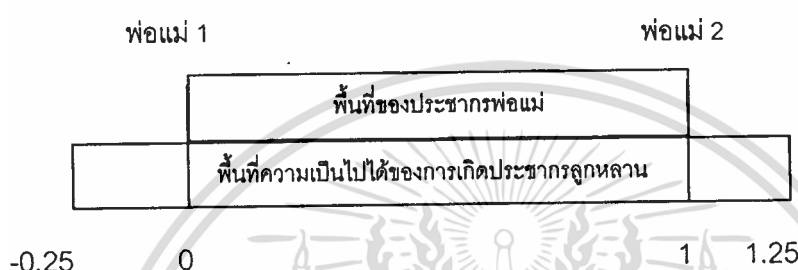
3.4.1.2 การรวมตัวของยีนแบบชั้นกลาง (Intermediate recombination)

การรวมตัวของยีนแบบชั้นกลาง [13] นี้จะใช้กับตัวแปรที่เข้ารหัสแบบจำนวนจริงเท่านั้น ซึ่งค่าตัวแปรของประชากรลูกหลานจะเกิดขึ้นรอบ ๆ ระหว่างค่าตัวแปรของประชากรพ่อแม่ โดยประชากรลูกหลานที่เกิดขึ้นจะได้จากสมการที่ (3.6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประชากรลูกหลาน = ประชากรพ่อแม่ 1 + α (ประชากรพ่อแม่ 2 – ประชากรพ่อแม่ 1) 3.5

ซึ่ง α เป็นแฟกเตอร์สเกล (scaling factor) ที่ถูกเลือกโดยการสุ่มค่าที่อยู่ระหว่าง $[-d, 1+d]$ การรวมตัวของยีนแบบชั้นกลางจะมีค่า $d = 0$ สำหรับระยะการขยายออก $d > 0$ ค่าที่ดีที่สุดที่ถูกเลือกจะมีค่า $d = 0.25$ ค่าของตัวแปรประชากรลูกหลานที่ได้ จะมีขอบเขตที่มากกว่าค่าของตัวแปรประชากรพ่อแม่เท่ากับค่าของ α ที่ถูกเลือกดังรูปที่ 3.5



รูปที่ 3.5 พื้นที่ขอบเขตของประชากรลูกหลานที่เทียบกับค่าของประชากรพ่อแม่

พิจารณาประชากรพ่อแม่ 2 ตัวที่อยู่ข้างล่างนี้ โดยที่ประชากรพ่อแม่แต่ละตัวจะมี 3 ตัวแปร

ประชากรพ่อแม่ 1 12 25 5

ประชากรพ่อแม่ 2 123 4 34

ค่าของของ α ที่สุ่มได้เท่ากับ

ตัวอย่าง 1 0.5 1.1 -0.1

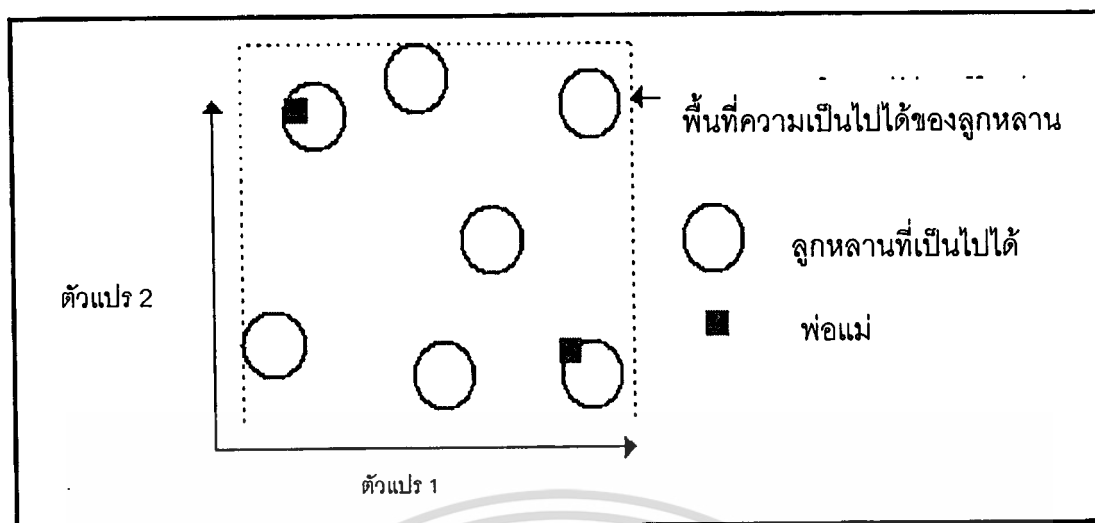
ตัวอย่าง 2 0.1 0.8 0.5

ค่าของประชากรลูกหลานใหม่ที่คำนวณได้

ประชากรลูกหลาน 1 67.5 1.9 2.1

ประชากรลูกหลาน 2 23.1 8.2 19.5

การรวมตัวของยีนแบบชั้นกลางสามารถสร้างจุดภายในไฮเปอร์คิวบ์ (Hypercube) โดยที่ความกว้างจะขึ้นอยู่กับตัวประชากรพ่อแม่ซึ่งเป็นผู้กำหนดในรูปที่ 3.6 แสดงพื้นที่ของประชากรลูกหลานหลังจากทำการรวมตัวของยีนแบบชั้นกลางเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ตำแหน่งที่เป็นไปได้ของการเกิดประชากรลูกหลานของการรวมตัวยีนแบบชั้นกลาง

3.4.1.3 การรวมตัวของยีนแบบเส้น (Line Recombination)

การรวมตัวของยีนแบบเส้นจะมีวิธีการที่คล้ายกับการรวมตัวของยีนแบบชั้นกลาง แต่จะมีความแตกต่างที่จะใช้ค่า α ค่าเดียวกับค่าตัวแปรของประชากรทั้งหมดดังตัวอย่าง

ประชากรพ่อแม่ 1	12	25	5
ประชากรพ่อแม่ 2	123	4	34

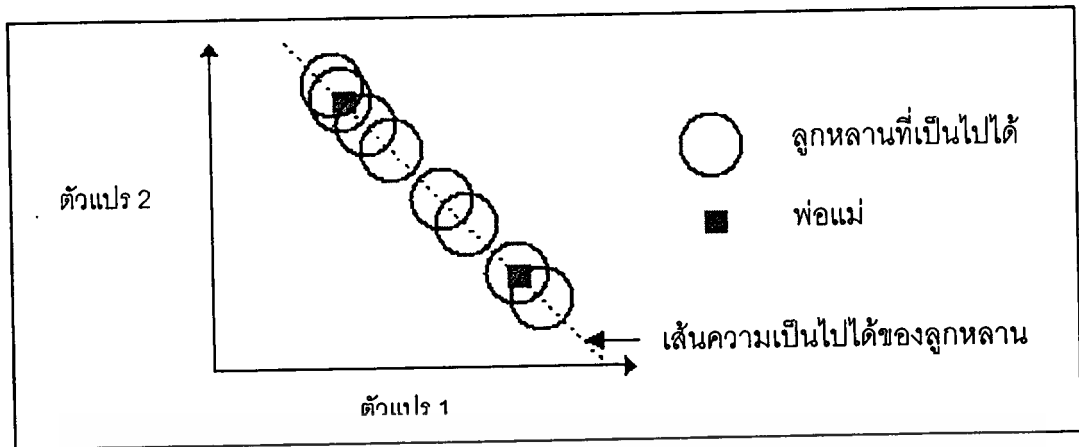
ค่าของของ α ที่สุ่มได้เท่ากับ

ตัวอย่าง 1	0.5
ตัวอย่าง 2	0.1

ค่าของประชากรลูกหลานใหม่ที่คำนวณได้

ประชากรลูกหลาน 1	67.5	14.5	19.5
ประชากรลูกหลาน 2	23.1	22.9	7.9

การรวมตัวของยีนแบบเส้นจะสร้างประชากรลูกหลานเรียงเป็นเส้นที่กำหนดโดยประชากรพ่อแม่ ดังรูปที่ 3.7

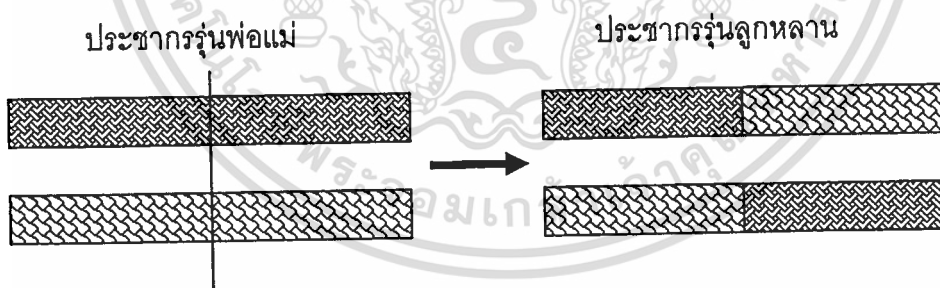


รูปที่ 3.7 ตำแหน่งที่เป็นไปได้ของการเกิดประชากรลูกหลานของการรวมตัวของยีนแบบเส้น

3.4.2 การรวมตัวของยีนโดยใช้ค่าไบนารี (Binary Valued Recombination)

3.4.2.1 การครอสโอเวอร์แบบจุดเดียว (Single - point Crossover)

การครอสโอเวอร์แบบจุดเดียวจะทำการครอสโอเวอร์หนึ่งจุดในช่วง $[1, 2, \dots, N\text{-var} - 1]$ โดยที่ $N\text{-var}$ คือจำนวนตัวแปรของแต่ละโครโมโซม หรือประชากร ซึ่งจะทำการเลือกจุดที่จะทำการครอสโอเวอร์โดยการสุ่ม และตัวแปรแลกเปลี่ยนกันระหว่างประชากรพ่อแม่ทั้งสองตัวหลังจุดที่สุ่มได้ ดังนั้นประชากรลูกหลานสองตัวที่ได้ จะแสดงในรูปที่ 3.8



รูปที่ 3.8 การครอสโอเวอร์แบบจุดเดียว

พิจารณาประชากรพ่อแม่ 2 ตัวที่อยู่ข้างล่าง โดยที่ประชากรพ่อแม่แต่ละตัวจะมี 11 ตัวแปร

ประชากรพ่อแม่ 1	0	1	1	1	0	0	1	1	0	1	0
ประชากรพ่อแม่ 2	1	0	1	0	1	1	0	0	1	0	1

ตำแหน่งของการครอสโอเวอร์ที่สุ่มได้เท่ากับ 5

หลังจากการครอสโอเวอร์แล้วจะได้ประชากรรุ่นลูกหลานดังนี้

ประชากรลูกหลาน 1	0	1	1	1	0	1	0	0	1	0	1
ประชากรลูกหลาน 2	1	0	1	0	1	0	1	1	0	1	0

3.4.2.2 การครอสโอเวอร์แบบหลายจุด (Multi - point Crossover)

สำหรับการครอสโอเวอร์แบบหลายจุด จะประกอบด้วย m ตำแหน่งในการครอสโอเวอร์ที่อยู่ในช่วง $[1, 2, \dots, N\text{-var} - 1]$ $N\text{-var}$ คือจำนวนตัวแปรของประชากรแต่ละตัว ซึ่งจะทำให้การเลือกจุดที่จะทำการครอสโอเวอร์โดยการสุ่มตามจำนวนของ m ต่อจากนั้นตัวแปรจะแลกเปลี่ยนกันระหว่างประชากรพ่อแม่ทั้งสองตัว และจะไม่มีแลกเปลี่ยนกันระหว่างตัวแปรตัวแรกและตำแหน่งแรกของการครอสโอเวอร์ แสดงในรูปที่ 3.9



รูปที่ 3.9 การครอสโอเวอร์แบบหลายจุด

พิจารณาประชากรพ่อแม่ 2 ตัวที่อยู่ข้างล่าง โดยที่ประชากรพ่อแม่แต่ละตัวจะมี 11 ตัวแปร

ประชากรพ่อแม่ 1	0	1	1	1	0	0	1	1	0	1	0
ประชากรพ่อแม่ 2	1	0	1	0	1	1	0	0	1	0	1

ตำแหน่งของการครอสโอเวอร์ที่สุ่มได้ ($m=3$) ประกอบด้วย 2

2

6

10

หลังจากการครอสโอเวอร์แล้วจะได้ประชากรรุ่นลูกหลานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประชากรลูกหลาน 1 0 1 | 1 0 1 1 | 1 1 0 1 | 1
 ประชากรลูกหลาน 2 1 0 | 1 1 0 0 | 0 0 1 0 | 0

3.4.2.3 การครอสโอเวอร์แบบยูนิฟอร์ม (Uniform crossover)

การครอสโอเวอร์แบบยูนิฟอร์ม [14] ซึ่งเป็นวิธีการสร้างประชากรลูกหลาน โดยขึ้นอยู่กับการสร้างการระบุตำแหน่ง (mask) โดยการสุ่ม

พิจารณาประชากรพ่อแม่ 2 ตัวที่อยู่ข้างล่าง โดยที่ประชากรพ่อแม่แต่ละตัวจะมี 11 ตัวแปร

ประชากรพ่อแม่ 1	0	1	1	1	0	0	1	1	0	1	0
ประชากรพ่อแม่ 2	1	0	1	0	1	1	0	0	1	0	1
Mask	1	0	0	1	1	0	1	0	1	0	1
ประชากรลูกหลาน 1	0	0	1	1	0	1	1	0	0	0	0
ประชากรลูกหลาน 2	1	1	1	0	1	0	0	1	1	1	1

จะเห็นว่าประชากรพ่อแม่ 1 และ ประชากรพ่อแม่ 2 จะสลับบิตกันก็ต่อเมื่อ mask บิตที่ตำแหน่งนั้นเป็น 0 และจะไม่สลับบิตเมื่อ mask บิตที่ตำแหน่งนั้นเป็น 1 หลังจากการสลับจะได้เป็นประชากรลูกหลาน 1 และประชากรลูกหลาน 2

3.5 การกลายพันธุ์ (Mutation)

การกลายพันธุ์เป็นวิธีการสร้างความหลากหลายให้กับประชากร หรือเป็นการผ่าเหล่าหลังจากการครอสโอเวอร์ โดยทั่วไปการกลายพันธุ์จะออกแบบไว้สำหรับประชากรที่เข้ารหัสแบบไบนารีเท่านั้น แต่ในปัจจุบันมีการกลายพันธุ์แบบสุ่มซึ่งเป็นวิธีที่ใช้กับประชากรที่เข้ารหัสแบบจำนวนจริง

3.6 การแทนที่ (Reinsertion)

หลังจากที่สร้างประชากรลูกหลานแล้ว จะมีการแทนที่ประชากรพ่อแม่ปัจจุบันด้วย

ประชากรลูกหลานทุกตัว เช่น ถ้าหากขนาดประชากรเท่ากับ N วิธีนี้จะสร้างประชากรลูกหลาน N ไม่ว่าการันใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ N ด้วย วิธีนี้อาจจะทำให้ประชากรที่ดีเสียหายในรุ่นถัดไป ดังนั้นจึงมีการสำเนาประชากร วิธีนี้จะทำสำเนาประชากรที่ดีที่สุด 2-5 ตัว ในรุ่นถัดไป วิธีการสำเนาประชากรจะช่วยเพิ่มความเร็วในการหาคำตอบที่ดีที่สุดถ้าหากกำหนดจำนวนสำเนาประชากรได้อย่างเหมาะสม

เนื่องจากการสร้างประชากรลูกหลานที่มีขนาดใหญ่ จะทำให้การคำนวณในแต่ละรอบของวิวิธพันธุศาสตร์นานขึ้น จึงมีวิธีการแทนที่แบบอื่นที่สร้างประชากรลูกหลานจำนวนน้อยลงเพื่อช่วยเพิ่มความเร็วในการคำนวณ เช่น การแทนที่ประชากรพ่อแม่ด้วยประชากรลูกหลานโดยตรง คือ จะเลือกแทนที่ประชากรที่เลวที่สุดเมื่อมีการแทรกประชากรรุ่นใหม่เข้ามา หรืออาจแทนที่ประชากรที่อาศัยอยู่เป็นเวลานาน ในบทต่อไปจะเป็นการนำเอาวิธีต่าง ๆ ในวิวิธพันธุศาสตร์ มาออกแบบกระบวนการการปรับค่าของระบบควบคุม



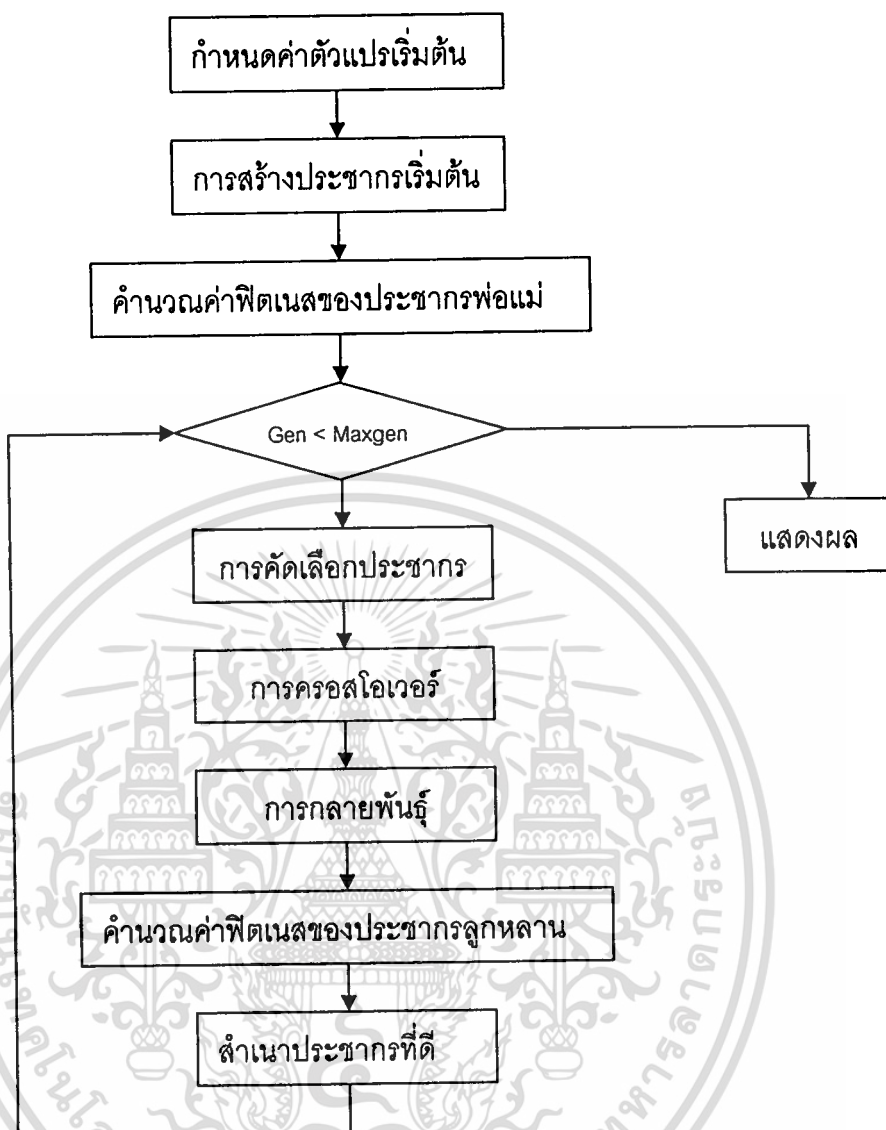
บทที่ 4

การออกแบบและหลักการทำงาน

บทนี้อธิบายขั้นตอนการออกแบบโปรแกรม โดยเริ่มแรกเป็นการออกแบบโปรแกรมทางพันธุศาสตร์ เพื่อทดสอบประสิทธิภาพของโปรแกรมทางพันธุศาสตร์ที่ได้ออกแบบไว้ ต่อมาจะเป็นโปรแกรมจำลองตัวควบคุมแบบฟัซซีฟีด ฟัซซีฟิว และฟัซซีฟิสแควรีไอดี ซึ่งจะนำมาใช้เป็นฟังก์ชันเป้าหมายในโปรแกรมทางพันธุศาสตร์ และโปรแกรมการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟิสแควรีไอดีแบบ 8 ตัวแปร ซึ่งจะประกอบด้วยตัวแปรของตัวควบคุมแบบฟัซซีฟีด 4 ตัว {Kp, Kd, Ld, Ku_pd} ตัวควบคุมแบบฟัซซีฟิว 4 ตัว {Kp, Ki, Li, Ku_pi} และโปรแกรมการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟิสแควรีไอดีแบบ 10 ตัวแปร {Kp, Kd, Ld, Ku_pd, Kp, Ki, Li, Ku_pi, Ts, Sel}แ

4.1 การออกแบบโปรแกรมทางพันธุศาสตร์

เนื่องจากการใช้งานวิธีทางพันธุศาสตร์ส่วนมาก จะต้องอาศัยเวลาในการประมวลผลที่ค่อนข้างนาน กว่าจะได้ค่าที่เหมาะสมหรือค่าที่ดีที่สุดและกระบวนการทางพันธุศาสตร์ แต่ละขั้นตอนนั้นมีหลายวิธีการ ดังนั้นการออกแบบโปรแกรมจึงมีความสำคัญมาก เพื่อให้โปรแกรมที่ได้ออกแบบมีประสิทธิภาพที่ดีที่สุด ดังนั้นการออกแบบโปรแกรมจึงต้องคำนึงถึงปัญหาเหล่านี้ด้วย โดยการออกแบบโปรแกรมทางพันธุศาสตร์ที่เขียนขึ้นแสดงในโฟลว์ชาร์ตรูปที่ 4.1 โดยเริ่มแรกจะเป็นการกำหนดค่าพารามิเตอร์ต่าง ๆ ด้วยวิธีทางพันธุศาสตร์ และขนาดของประชากรเริ่มต้น ต่อมาเมื่อได้ประชากรเริ่มต้นแล้ว จะคำนวณค่าของประชากรพ่อแม่ที่ได้ หลังจากนั้นเป็นการตรวจสอบเงื่อนไข โดย Gen คือ ตัวนับจำนวนที่ทำงานอยู่ ส่วน Maxgen คือ จำนวนรอบสูงสุดที่โปรแกรมทำงาน ต่อจากนั้นจะเป็นการคัดเลือกประชากร เมื่อได้ประชากรพ่อแม่ที่เลือกมาแล้วจะเป็นการผสมกันระหว่างประชากรแต่ละคู่ โดยการครอสโอเวอร์ และการกลายพันธุ์ จะได้เป็นประชากรรุ่นลูกหลาน หลังจากนั้นจะคำนวณค่าของประชากรลูกหลานที่ได้ โดยจะใช้ฟังก์ชันเดียวกับที่คำนวณประชากรพ่อแม่ หลังจากนั้นจะเป็นการสำเนาประชากรที่ดี 5 ตัวในประชากรรุ่นพ่อแม่ เพื่อใช้เป็นประชากรรุ่นใหม่ต่อไป และเมื่อครบจำนวนรอบสูงสุดที่ได้กำหนดไว้ จะแสดงผลของประชากรที่ดีที่สุดออกมา



รูปที่ 4.1 ฟลอร์ชาร์ตโปรแกรมทางพันธุศาสตร์

4.1.1 การแทนที่ประชากรและการกำหนดค่าตัวแปรเริ่มต้น

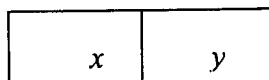
เพื่อลดความซับซ้อน และสะดวกในการคำนวณของโปรแกรมทางพันธุศาสตร์ จึงได้มีการเข้ารหัสประชากรแต่ละตัวเป็นค่าจำนวนจริง จากฟังก์ชันที่ใช้ในการทดสอบจากสมการที่ (4.1) เพื่อตรวจสอบและสาธิตให้เห็นวิธีการที่ได้กล่าวไว้ในข้างต้น

$$f(x, y) = \frac{1}{(x + 0.5)^2 + 2(y - 0.5)^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.8} \quad (4.1)$$

โดยที่ $-1 \leq x \leq 1, -1 \leq y \leq 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ (4.1) จะเห็นว่ามีตัวแปรสองตัวคือ x และ y ประชากรหรือโครโมโซมหนึ่งตัวจะประกอบด้วย x และ y ต่อกันดังรูปที่ 4.2



รูปที่ 4.2 การแทนตัวแปร x และ y ของประชากร

โดยจะแทนค่าตัวแปร x และ y ด้วยค่าจำนวนจริง ซึ่งมีขอบเขตอยู่ระหว่าง -1 ถึง 1 และจะกำหนดจำนวนของประชากร คือ 30 ตัว และจำนวนรอบของการเกิดสูงสุด (Max generation) เท่ากับ 30 รอบ

4.1.2 การคัดเลือก

การคัดเลือกในการออกแบบโปรแกรมมี 2 วิธี คือ การคัดเลือกแบบวงล้อรูเล็ต และการคัดเลือกสุ่มแบบยูนิเวอร์ซอล (SUS) ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายเนื่องจากมีค่าความลำเอียงเป็นศูนย์และมีค่าการกระจายที่ดี โดยที่การคัดเลือกแบบวงล้อรูเล็ตจะแบ่งส่วนของพื้นที่รูเล็ตตามสัดส่วนของค่าฟิตเนส เส้นรอบวง คือ ผลรวมของค่าฟิตเนสของประชากรแต่ละตัว และจะสุ่มตามจำนวนของประชากรที่ต้องการ ส่วนการคัดเลือกสุ่มแบบยูนิเวอร์ซอลจะแบ่งพื้นที่เหมือนกับการคัดเลือกแบบวงล้อรูเล็ต ต่างกันตรงการคัดเลือก คือ จะใช้ตัวชี้ (Pointer) จำนวน N ตัว โดยระยะห่างของตัวชี้จะเท่ากันทุกตัว ประชากรทั้งหมดจะถูกเลือกจากการสุ่มตัวชี้ตัวแรกเพียงครั้งเดียว สำหรับการเลือกคู่ประชากรมาทำการครอสโอเวอร์นั้น ถ้าประชากรตัวไหนที่ไม่ได้ถูกเลือกทำการครอสโอเวอร์ ประชากรตัวนั้นจะถูกนำไปเป็นประชากรรุ่นต่อไปทันที

4.1.3 การครอสโอเวอร์

การครอสโอเวอร์เป็นการนำประชากรที่ได้คัดเลือกไว้แล้วในบ่อเพาะพันธุ์ แล้วนำมาจับคู่แลกเปลี่ยนยีนของประชากรพ่อแม่ ตามการกำหนดอัตราความน่าจะเป็นของการครอสโอเวอร์เพื่อสร้างประชากรรุ่นใหม่ ในการออกแบบโปรแกรมจะใช้วิธีการครอสโอเวอร์ที่ใช้กับจำนวนจริงเท่านั้น เนื่องจากข้อมูลของประชากรที่สร้างขึ้นเป็นค่าจำนวนจริง ซึ่งการครอสโอเวอร์แบบใช้ค่าจริงมี 3 วิธี คือ การรวมตัวของยีนแบบไม่ต่อเนื่อง การรวมตัวของยีนแบบชั้นกลาง และการรวมตัวของยีนแบบเส้น ส่วนการจะเลือกใช้วิธีไหนนั้น จะมีการทดสอบโปรแกรมในลำดับต่อไป

4.1.4 การกลายพันธุ์

การกลายพันธุ์เป็นการสร้างความหลากหลายให้กับประชากร หรือเป็นการผ่าเหล่า หลังจากการครอสโอเวอร์ ในการออกแบบโปรแกรมนั้นจะเลือกใช้การกลายพันธุ์แบบสุ่ม ซึ่งเป็นวิธีที่ใช้กับการเข้ารหัสแบบจำนวนจริง

4.1.5 การสำเนาประชากร

เป็นขั้นตอนในการแทนที่ประชากรลูกหลานในประชากรพ่อแม่ โดยจะสำเนาประชากรลูกหลานที่ดี 5 ตัวในประชากรพ่อแม่ เพื่อนำไปใช้เป็นประชากรในรุ่นถัดไป วิธีนี้จะเป็นการเพิ่มความเร็วในการค้นหาคำตอบ

4.1.6 ฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมายเป็นตัววัดคุณสมบัติของประชากร โดยค่าฟังก์ชันที่ใช้ในการออกแบบและทดสอบโปรแกรมจะเป็นไปตามสมการที่ (4.1) ซึ่งเป็นการหาค่าตัวแปร x และ y เพื่อให้สมการ (4.1) ได้ค่าของผลลัพธ์ที่สูงสุด

4.2 การทดสอบโปรแกรมทางพันธุศาสตร์

ในหัวข้อนี้จะเป็นการทดสอบโปรแกรมทางพันธุศาสตร์ ที่ได้ออกแบบว่ามีประสิทธิภาพแค่ไหน ซึ่งจะพิจารณาจากความเร็วและความถูกต้องในการหาคำตอบ โดยจะทำการทดสอบด้วยการหาค่าสูงสุดของฟังก์ชันทดสอบดังสมการที่ 4.1 [15] ในโปรแกรมทางพันธุศาสตร์ จะแทนตัวแปร x และ y ด้วยค่าจำนวนจริง ขนาดของประชากรเท่ากับ 30 ตัว และเจนเนอเรชันสูงสุดเท่ากับ 30 รอบ ที่เป็นเช่นนี้ เพราะจำนวนประชากรเริ่มต้นที่ 30 มีขนาดที่ไม่น้อยและไม่มากเกินไป ดังจะให้เห็น จากผลการทดลองในลำดับต่อไป

การทดสอบโปรแกรมทางพันธุศาสตร์จะทดสอบ 6 แบบ คือ

1. ใช้การคัดเลือกสุ่มแบบยูนิเวอร์ซอล และการครอสโอเวอร์แบบไม่ต่อเนื่อง
2. ใช้การคัดเลือกสุ่มแบบยูนิเวอร์ซอล และการครอสโอเวอร์แบบชั้นกลาง
3. ใช้การคัดเลือกสุ่มแบบยูนิเวอร์ซอล และการครอสโอเวอร์แบบเส้น
4. ใช้การคัดเลือกแบบวงล้อรูเท และการครอสโอเวอร์แบบไม่ต่อเนื่อง
5. ใช้การคัดเลือกแบบวงล้อรูเท และการครอสโอเวอร์แบบชั้นกลาง
6. ใช้การคัดเลือกแบบวงล้อรูเท และการครอสโอเวอร์แบบเส้น

ตารางที่ 4.1 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 1

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6548	0.4999	17	7.9468
2	-0.6541	0.5000	30	7.9460
3	-0.6551	0.4999	13	7.9468
4	-0.6550	0.4996	28	7.9464
5	-0.6551	0.5004	28	7.9465
6	-0.6547	0.5000	30	7.9467
ค่าเฉลี่ย	-0.6548	0.4999	24.33	7.9465

ตารางที่ 4.2 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 2

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6566	0.4999	12	7.9467
2	-0.6572	0.5007	12	7.9467
3	-0.6549	0.5001	22	7.9468
4	-0.6553	0.4999	24	7.9467
5	-0.6547	0.4999	27	7.9467
6	-0.6549	0.5001	28	7.9468
ค่าเฉลี่ย	-0.6556	0.5001	20.83	7.9467

ตารางที่ 4.3 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 3

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6552	0.5000	17	7.9468
2	-0.6550	0.5001	28	7.9468
3	-0.6551	0.5000	16	7.9468
4	-0.6550	0.5001	13	7.9468
5	-0.6552	0.5000	20	7.9468
6	-0.6551	0.5000	17	7.9468
ค่าเฉลี่ย	-0.6551	0.5	18.5	7.9468

ตารางที่ 4.4 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 4

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6549	0.5000	26	7.9467
2	-0.6537	0.5021	30	7.9359
3	-0.6549	0.5004	28	7.9464
4	-0.6546	0.5001	30	7.9467
5	-0.6551	0.5001	24	7.9488
6	-0.6555	0.4971	29	7.9285
ค่าเฉลี่ย	-0.6548	0.4999	27.83	7.9418

ตารางที่ 4.5 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 5

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6549	0.5000	26	7.9468
2	-0.6537	0.5021	30	7.9359
3	-0.6549	0.5004	28	7.9464
4	-0.6546	0.5001	30	7.9467
5	-0.6551	0.5001	24	7.9468
6	-0.6555	0.4971	29	7.9285
ค่าเฉลี่ย	-0.6548	0.4999	27.83	7.94185

ตารางที่ 4.6 ผลการแก้ปัญหาฟังก์ชันทดสอบด้วยวิธีที่ 5

ครั้งที่	x	y	รอบ gen	ค่าฟิตเนส
1	-0.6552	0.5000	19	7.9468
2	-0.6552	0.4999	16	7.9468
3	-0.6552	0.5000	25	7.9468
4	-0.6548	0.5001	28	7.9467
5	-0.6543	0.5013	30	7.9429
6	-0.6551	0.5000	13	7.9468
ค่าเฉลี่ย	-0.6549	0.5002	21.83	7.94613

จากตารางที่ 4.1 จะเห็นได้ว่าค่าของจำนวนรอบมีค่าเฉลี่ยอยู่ที่ 24.33 และผลของค่าฟิตเนสที่ได้ไม่ค่อยแน่นอน เช่นในการทดลองครั้งที่ 2 และครั้งที่ 4 ไม่ใช่ค่าที่ดีที่สุด หรือเรียกอีกอย่างว่าเป็นการลู่เข้าก่อนกำหนด ซึ่งค่าที่ได้ยังไม่ใช่ค่าที่ดีที่สุด

จากตารางที่ 4.2 จะเห็นว่าค่าของจำนวนรอบมีค่าเฉลี่ยอยู่ที่ 20.83 จะได้ผลที่ดีกว่าวิธีที่ใช้ในตารางที่ 4.1 ค่าฟิตเนสที่ได้มีค่ามากที่สุด แต่บางครั้งยังได้ค่าที่ไม่มากที่สุด

จากตารางที่ 4.3 จะเห็นว่าค่าของจำนวนรอบมีค่าเฉลี่ยอยู่ที่ 18.5 ซึ่งเป็นการลู่เข้าที่เร็วที่สุดใน 6 วิธี และยังได้ค่าฟิตเนสที่ดีที่สุดของทุกครั้งที่ทดลอง

จากตารางที่ 4.4 ค่าเฉลี่ยจำนวนรอบอยู่ที่ 27.83 ซึ่งเป็นค่าที่มากที่สุด ส่วนค่าของฟิตเนสจะได้ค่าที่ไม่ค่อยดีเมื่อเทียบกับตารางอื่น

จากตารางที่ 4.5 ได้ค่าเฉลี่ยจำนวนรอบ และค่าเฉลี่ยของค่าฟิตเนสเท่ากับตารางที่ 4.4

จากตารางที่ 4.6 มีค่าเฉลี่ยจำนวนรอบอยู่ที่ 21.83 ซึ่งดีกว่าของตารางที่ 4.4 และ 4.5 แต่ค่าฟิตเนสที่ได้ยังไม่ดีเมื่อเทียบกับตารางที่ 4.1 4.2 และ 4.3

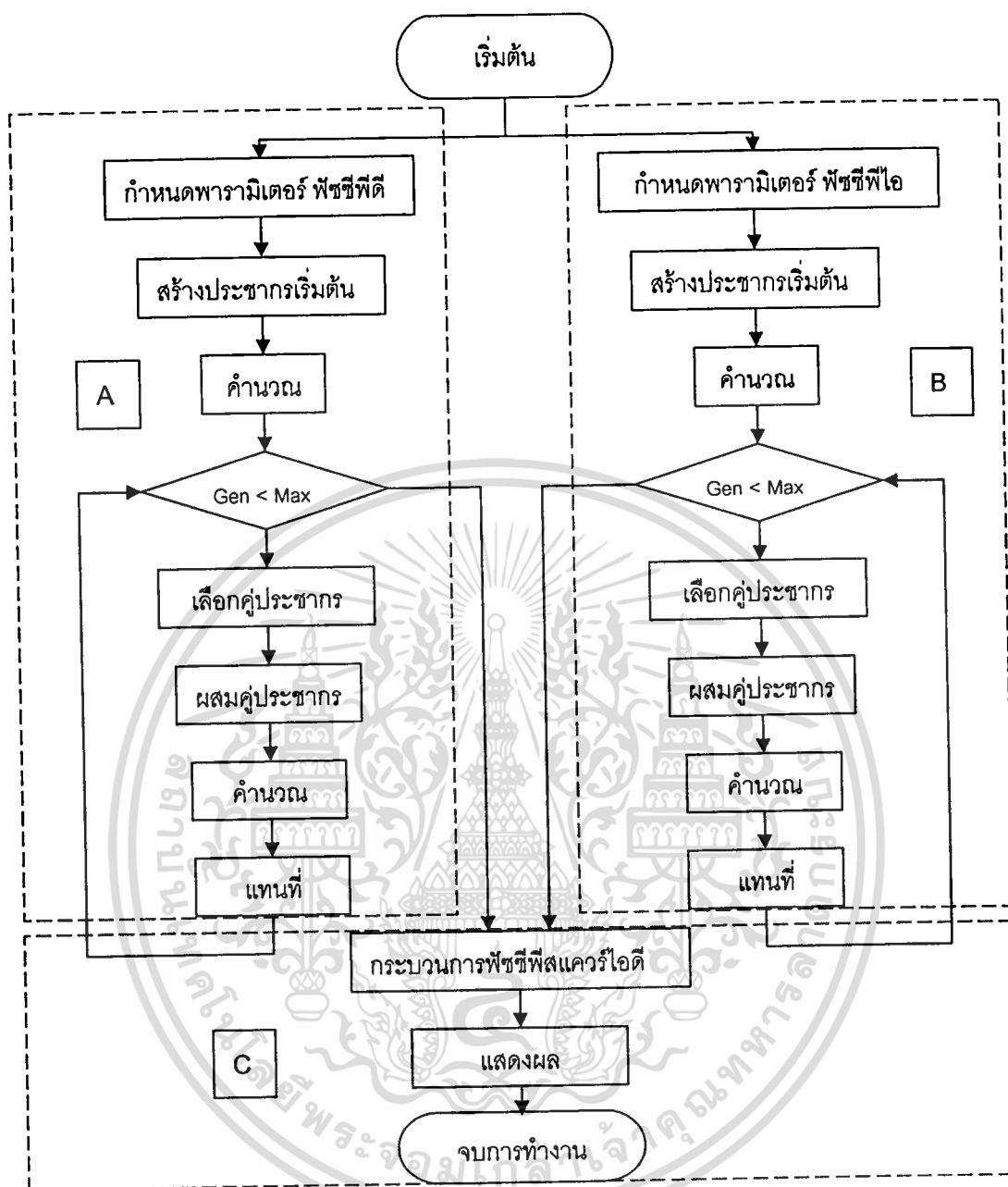
จากผลการทดสอบโปรแกรมทางพันธุศาสตร์ดังกล่าว พบว่าการใช้วิธีที่ 2 และ 3 มีความสามารถที่ใกล้เคียงกัน ในการหาค่าสูงสุดของฟังก์ชันทดสอบ แต่วิธีที่ 3 จะมีการลู่เข้าคำตอบเร็วกว่าวิธีที่ 2 ทำให้ลดเวลาในการประมวลผลได้ดีกว่า และยังได้ค่าที่สูงสุดทุกครั้งที่การทดลอง เมื่อได้โปรแกรมทางพันธุศาสตร์ที่เหมาะสมแล้ว จึงนำไปประยุกต์ใช้ในการหาคำตอบที่ดีที่สุดต่อไป

4.3 โปรแกรม GA ฟัชชีฟิสแควร์ไอดีแบบ 8 ตัวแปร

โปรแกรม GA ฟัชชีฟิสแควร์ไอดีแบบ 8 ตัวแปร เป็นโปรแกรมที่ใช้ในการค้นหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัชชีฟิสแควร์ไอดี ซึ่งจะแบ่งออกเป็น 3 ส่วน คือ

1. โปรแกรม GA ฟัชชีฟิตี
2. โปรแกรม GA ฟัชชีฟิไอ
3. โปรแกรมฟัชชีฟิสแควร์ไอดี

โดยโปรแกรมที่ 1 และ 2 เป็นการใช้วิธีทางพันธุศาสตร์ค้นหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัชชีฟิตี และ ตัวควบคุมแบบฟัชชีฟิไอดีตามลำดับ ส่วนโปรแกรมที่ 3 เป็นการนำค่าพารามิเตอร์ที่ได้จากโปรแกรมที่ 1 และ 2 มาทดสอบ [15,18] กับโปรแกรมที่ 3 ดังรูปที่ 4.3



รูปที่ 4.3 โฟลว์ชาร์ตโปรแกรมทางพันธุศาสตร์ของตัวควบคุมฟิตเนสฟิตแควรี่ไอดีแบบ 8 ตัวแปร

จากรูปที่ 4.3 เป็นกระบวนการทั้งหมดของโปรแกรม GA ฟิตเนสฟิตแควรี่ไอดีแบบ 8 ตัวแปร ซึ่งแบ่งการทำงานออกเป็น 3 ส่วน คือ โปรแกรม GA ฟิตเนสฟิตแสดงในบล็อกที่ A โปรแกรม GA ฟิตเนสฟิตไอแสดงในบล็อกที่ B และโปรแกรมฟิตเนสฟิตแควรี่ไอดีแสดงในบล็อก C ซึ่งแต่ละบล็อกมีรายละเอียดดังนี้คือ

4.3.1 โปรแกรม GA ฟิตเนสฟิต

โปรแกรมนี้จะเป็นการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟิตเนสฟิต โดยใช้วิธีการทางพันธุศาสตร์ ซึ่งจะประกอบด้วย 4 พารามิเตอร์ คือ Kp, Kd, Ku_pd และ Ld ซึ่งจะต้องมีการไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่าพารามิเตอร์ต่าง ๆ ของวิธีการทางพันธุศาสตร์ และการเลือกใช้วิธีการของแต่ละขั้นตอนดังต่อไปนี้

1. ความยาวของโครโมโซม

เนื่องจากตัวควบคุมแบบฟัซซี่ฟิตีมีค่าพารามิเตอร์ที่ต้องการหา คือ K_p , K_d , K_u_{pd} , L_d จะกำหนดโครโมโซม I ดังนี้

$$I = \{K_p, K_d, K_u_{pd}, L_d\}$$

โดยจะแทนค่าพารามิเตอร์ด้วยเลขจำนวนจริง ซึ่งจะกำหนดขอบเขตของพารามิเตอร์แต่ละตัวอยู่ที่ 0.0001 ถึง 10 และจะมีการกำหนดค่าพารามิเตอร์เริ่มต้นของโปรแกรม คือ ค่าของประชากรเริ่มต้นอยู่ที่ 15 30 60 90 120 ตัวตามลำดับ จำนวนการเกิด (generation) สูงสุดอยู่ที่ 30 รอบ และค่าเวลาในการสุ่ม อยู่ที่ 0.01 ถึง 0.001 วินาที

2. ฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมายเป็นการหาค่าของเวลาไต่ขึ้น (Rise Time) ที่น้อยที่สุด โดยจะวัดผลที่ได้จากเอาต์พุต ที่คำนวณจากสมการที่ (4.2)

$$\text{Rise Time} = [t(90\%) - t(10\%)] \times T_s \quad (4.2)$$

โดยที่ Rise Time เป็นเวลาที่วัดจากผลตอบสนองที่มีขนาดเพิ่มขึ้นจาก 10 % ถึง 90 % ของค่าเป้าหมาย (Set Point) $t(90\%)$ เป็นเวลาที่ผลตอบสนองที่ 90 % ของค่าเป้าหมาย $t(10\%)$ เป็นเวลาที่ผลตอบสนองที่ 10 % ของค่าเป้าหมาย และ T_s เป็นช่วงเวลาในการสุ่ม

3. การดำเนินการทางพันธุศาสตร์

ผลของการทดสอบโปรแกรมเจเนติกอัลกอริทึมในหัวข้อ 4.2 ใช้การคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม (Stochastic Universal Sampling) และการครอสโอเวอร์แบบเส้น (line recombination) แสดงในตารางที่ 4.3 ส่วนการมิวเตชันเป็นการมิวเตชันแบบจำนวนจริง

4.3.2 โปรแกรม GA ฟัซซี่ฟีโอ

โปรแกรมนี้จะเป็นการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซี่ฟีโอ โดยใช้วิธีการทางพันธุศาสตร์ ซึ่งจะประกอบด้วย 4 พารามิเตอร์เช่นกัน คือ K_p , K_i , K_u_{pi} และ L_i ซึ่งจะต้องมีการกำหนดค่าพารามิเตอร์ต่าง ๆ ของวิธีทางพันธุศาสตร์ และการเลือกใช้วิธีการของแต่ละขั้นตอนดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความยาวของโครโมโซม

เนื่องจากตัวควบคุมแบบพีซีพีไอมีค่าพารามิเตอร์ที่ต้องการหา คือ K_p , K_i , K_{u_pi} , L_i จะกำหนดโครโมโซม I ดังนี้

$$I = \{K_p, K_i, K_{u_pi}, L_i\}$$

โดยจะแทนค่าพารามิเตอร์ด้วยเลขจำนวนจริง ซึ่งจะกำหนดขอบเขตของพารามิเตอร์แต่ละตัวอยู่ที่ 0.0001 ถึง 10 และจะมีการกำหนดค่าพารามิเตอร์เริ่มต้นของโปรแกรม คือ ค่าของประชากรเริ่มต้นอยู่ที่ 15 30 60 90 120 ตัวตามลำดับ จำนวนการเกิด (generation) สูงสุดอยู่ที่ 30 รอบและค่าเวลาในการ sampling อยู่ที่ 0.01 ถึง 0.001 วินาที

2. ฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมายของตัวควบคุมแบบพีซีพีไอ ต้องการผลตอบสนองที่มีความเรียบของสัญญาณมากที่สุดหรือค่าพุ่งเกิน (overshoot) น้อยที่สุด คำนวณได้จากสมการที่ (4.3)

$$M_p = C_{\max} - C_{ss} \quad (4.3)$$

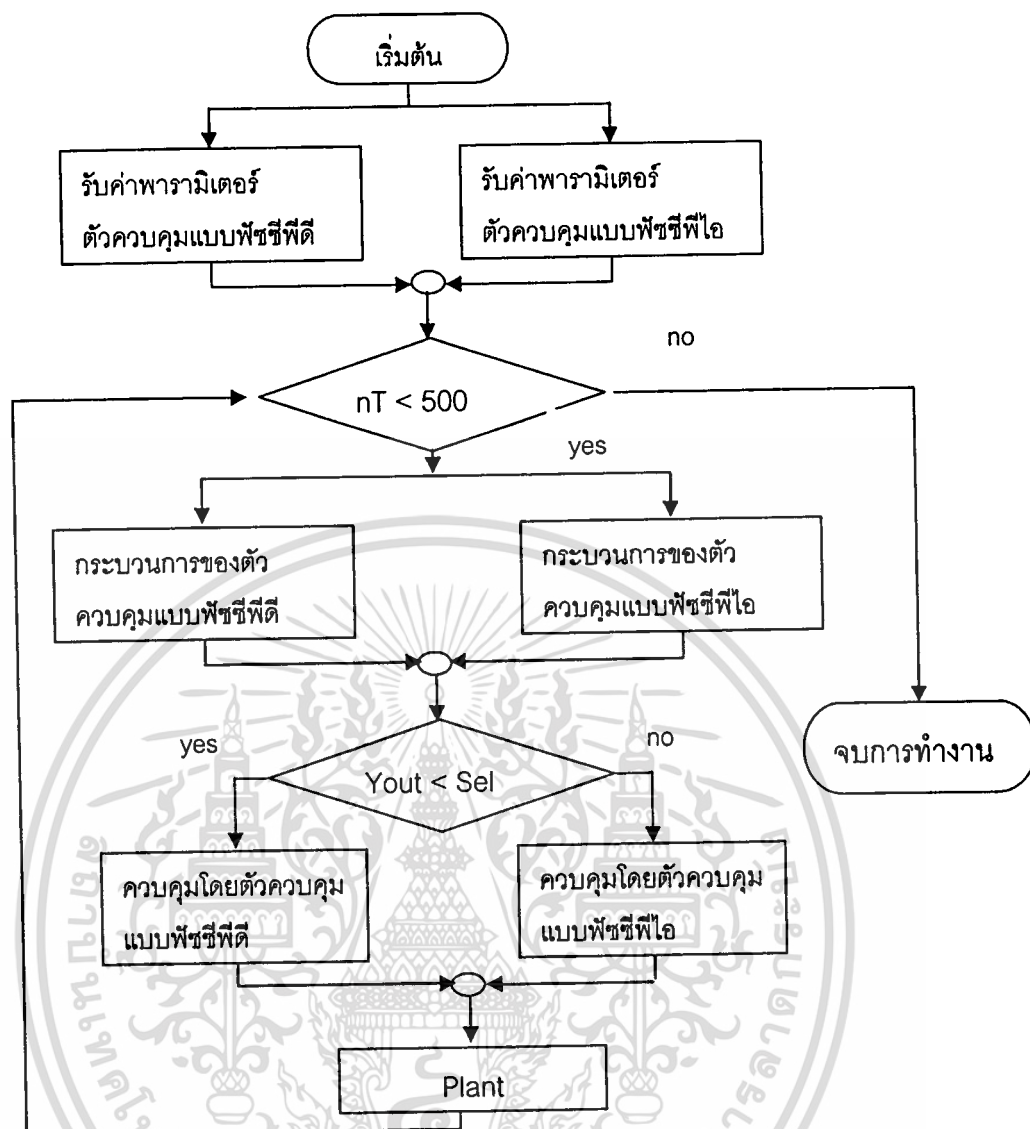
โดยที่ M_p เป็นค่าพุ่งเกินสูงสุด $C_{\max}$ คือ เอาต์พุตที่มีค่าสูงสุดของระบบ C_{ss} คือ เอาต์พุตที่สถานะอยู่ตัว

3. การดำเนินการทางพันธุศาสตร์

จากผลของการทดสอบโปรแกรมทางพันธุศาสตร์ในหัวข้อ 4.2 โดยใช้การคัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม (Stochastic Universal Sampling) และการครอสโอเวอร์แบบเส้น (line recombination) ในตารางที่ 4.3 ส่วนการมิวเตชันเป็นการมิวเตชันแบบจำนวนจริง

4.3.3 โปรแกรมพีซีพีสแควร์ไอดี

เมื่อได้ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีดี และตัวควบคุมแบบพีซีพีไอ จากโปรแกรมในหัวข้อที่ 4.2.1 และ 4.2.2 จะนำค่าพารามิเตอร์มาใส่ในโปรแกรมพีซีพีสแควร์ไอดี ซึ่งเป็นกระบวนการของตัวควบคุมแบบพีซีพีสแควร์ไอดี ตามโฟลว์ชาร์ตโปรแกรมพีซีพีสแควร์ไอดีในรูปที่ 4.4

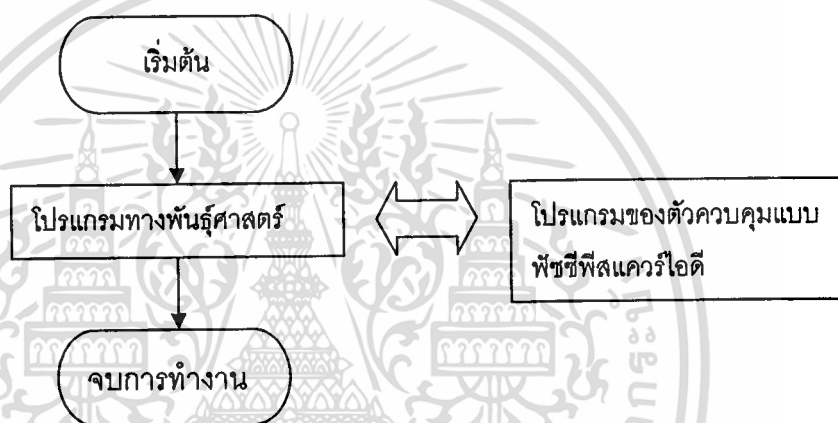


รูปที่ 4.4 ฟลว์ชาร์ตโปรแกรมฟัซซี่ฟีสแควรี่ไอดี

เริ่มต้นโปรแกรมจะเป็นการรับค่าพารามิเตอร์ของตัวควบคุมทั้ง 2 แบบ ซึ่งมี 8 ตัวแปรมา กำหนดค่าให้กับตัวควบคุมทั้ง 2 แบบ โดยตัวควบคุมทั้ง 2 แบบจะทำงานขนานกัน ซึ่งจะมีสวิตช์ เป็นตัวเลือกใช้ตัวควบคุมในการควบคุม Plant จากรูปที่ 4.4 จะเห็นได้ว่าการเลือกใช้ตัวควบคุมจะเป็นการตรวจสอบผลตอบสนองที่ออกจาก Plant ถ้าน้อยกว่าจุดเลือกตัวควบคุม (selected point) จะใช้ตัวควบคุมแบบฟัซซี่ฟีโอในการควบคุม แต่ถ้ามากกว่าจะใช้ตัวควบคุมแบบฟัซซี่ฟีด ในการควบคุม โดยโปรแกรมที่เขียนขึ้นนี้จะแสดงอยู่ในภาคผนวก

4.4 โปรแกรม GA ฟัชชีฟัสแควร์ไอดีแบบ 10 ตัวแปร

โปรแกรม GA ฟัชชีฟัสแควร์ไอดีแบบ 10 ตัวแปร จะแตกต่างกับโปรแกรมในหัวข้อ 4.4 ตรงที่ไม่มีการแบ่งค่าพารามิเตอร์ ซึ่งการหาค่าพารามิเตอร์ของตัวควบคุมแบบก่อนจะเป็นการหาค่าพารามิเตอร์แยกกันระหว่างตัวควบคุมแบบฟัชชีฟัสแควร์ไอดีกับตัวควบคุมแบบฟัชชีฟัสแควร์ไอดี โดยให้ฟังก์ชันเป้าหมายที่แตกต่างกันขึ้นอยู่กับคุณสมบัติของแต่ละตัวควบคุม ซึ่งโปรแกรม GA ฟัชชีฟัสแควร์ไอดีแบบ 10 ตัวแปรจะประกอบด้วย 2 ส่วน ส่วนแรกเป็นโปรแกรมหลักเป็นโปรแกรมทางพันธุศาสตร์ และส่วนที่สองเป็นโปรแกรมย่อยในรูปที่ 4.4 เป็นโปรแกรมของตัวควบคุมแบบฟัชชีฟัสแควร์ไอดี ซึ่งจะใช้เป็นฟังก์ชันเป้าหมายของโปรแกรมทางพันธุศาสตร์ ดังรูปที่ 4.5



รูปที่ 4.5 ไฟลว์ชาร์ตการทำงานของโปรแกรม GA ฟัชชีฟัสแควร์ไอดีแบบ 10 ตัวแปร

โปรแกรมทางพันธุศาสตร์ดังรูปที่ 4.1 เริ่มต้นจะทำงานโดยจะเรียกโปรแกรมย่อย ซึ่งเป็นโปรแกรมของตัวควบคุมแบบฟัชชีฟัสแควร์ไอดีดังรูปที่ 4.4 และมีการกำหนดค่าต่าง ๆ ดังนี้

1. ความยาวของโครโมโซม

ความยาวของโครโมโซมประกอบด้วย 10 ตัวแปรประกอบด้วย ตัวควบคุมแบบฟัชชีฟัสแควร์ไอดีที่มีค่าพารามิเตอร์ที่ต้องการหา คือ K_p , K_d , L_d , K_{u_pd} ตัวควบคุมแบบฟัชชีฟัสแควร์ไอดี K_p , K_i , L_i , K_{u_pi} จุดเลือก (select) และ เวลาในการสุ่ม (T_s) จะกำหนดโครโมโซม I ดังนี้

$$I = \{ K_p, K_d, L_d, K_{u_pd}, K_p, K_i, L_i, K_{u_pi}, sel, T_s \}$$

โดยจะแทนค่าพารามิเตอร์ด้วยเลขจำนวนจริง ซึ่งจะกำหนดขอบเขตของพารามิเตอร์แต่ละตัวอยู่ที่ 0.0001 ถึง 10 และจะมีการกำหนดค่าพารามิเตอร์เริ่มต้นของโปรแกรม คือ ค่าของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประชากรเริ่มต้นอยู่ที่ 15 30 60 90 และ 120 ตัวตามลำดับ จำนวนการเกิด (generation) สูงสุด อยู่ที่ 30 รอบและค่าเวลาในการสุ่ม อยู่ที่ 0.01 ถึง 0.001 วินาที

2. ฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมายของตัวควบคุมแบบฟัซซีพีไอ ต้องการผลตอบสนองที่มีความเรียบของ สัญญาณมากที่สุดหรือค่าเฉลี่ยผิดพลาดของสัญญาณน้อยที่สุด (IAE : Integral Absolute Error) จึงใช้สมการที่ (4.4) [16]

$$IAE = \int_0^{\infty} |e(t)| dt \quad (4.4)$$

โดยที่ IAE เป็นค่าเฉลี่ยผิดพลาดของสัญญาณ $e(t)$ เป็นสัญญาณผิดพลาดแต่ละ คาบเวลาหาได้จาก ค่าของผลตอบสนอง - ค่าของเป้าหมาย สมการที่ (4.4) สามารถแปลงในรูป เวลาไม่ต่อเนื่อง นำไปใช้ในการคำนวณต่อไปได้ดังสมการที่ (4.5)

$$IAE = \sum_0^{\infty} |e(t)| \quad (4.5)$$

3. การดำเนินการทางพันธุศาสตร์

ผลของการทดสอบโปรแกรมทางพันธุศาสตร์ได้นำเสนอไปแล้วในหัวข้อ 4.2 โดยใช้การ คัดเลือกแบบยูนิเวอร์ซอลโดยสุ่ม (Stochastic Universal Sampling) และการครอสโอเวอร์แบบ เส้น (line recombination) ตามตารางที่ 4.3 ส่วนการมิวเตชันเป็นการมิวเตชันแบบจำนวนจริง ส่วนการจำลองจะนำเสนอในบทที่ 5 ต่อไป

บทที่ 5

ผลการจำลองและวิจารณ์ผลการจำลอง

เนื้อหาในบทนี้จะเป็นการจำลองของโปรแกรมที่ได้ออกแบบไว้ในบทที่ 4 โดยแบ่งออกเป็น 2 ส่วน คือ ส่วนแรกจะจำลองหาค่าพารามิเตอร์จากโปรแกรม GA ฟัซซีฟิสแควร์ไอดีแบบ 8 ตัวแปร และส่วนที่สองจะเป็นการจำลองหาค่าพารามิเตอร์จากโปรแกรม GA ฟัซซีฟิสแควร์ไอดีแบบ 10 ตัวแปร โดยจะแบ่งออกเป็น 4 การจำลองดังนี้

5.1 การจำลองหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟิสแควร์ไอดีแบบ 8 ตัวแปร

การจำลองจะแบ่งเป็นสองการจำลอง โดยการจำลองแรกจะใช้แบบจำลองที่ไม่เป็นเชิงเส้น ส่วนการจำลองที่ 2 ใช้แบบจำลองหุ่นยนต์เคลื่อนที่

5.1.1 แบบจำลองที่ไม่เป็นเชิงเส้น (Nonlinear Model)

การจำลองนี้จะเป็นการทดสอบหาค่าแนวโน้มของการลู่เข้าของโปรแกรม โดยใช้จำนวนของประชากรเริ่มต้นที่แตกต่างกัน และอีกส่วนจะเป็นการเปรียบเทียบกันระหว่างการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ สมการที่ใช้ดังสมการที่ (5.1)

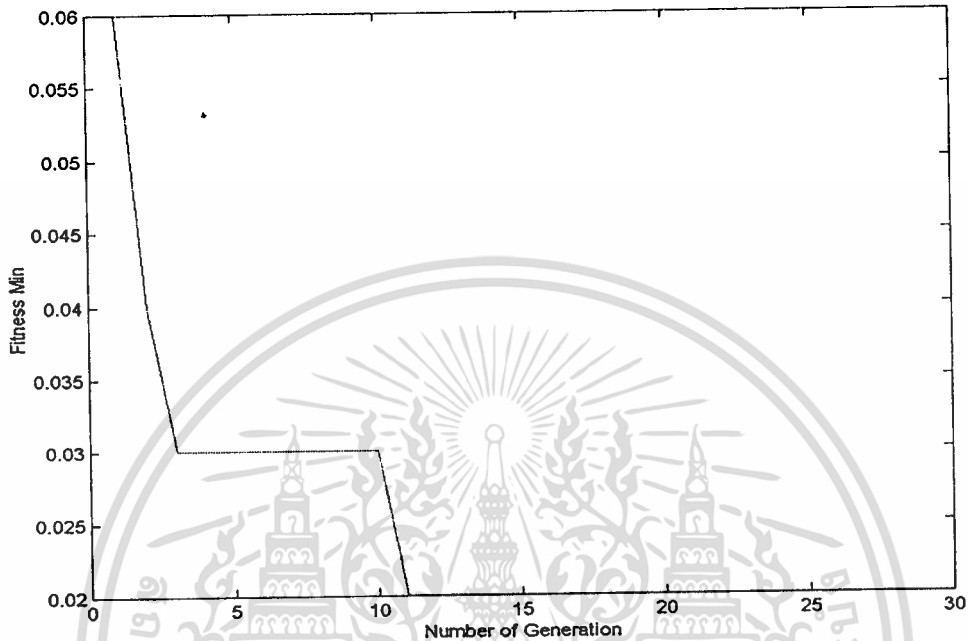
$$\dot{y}(t) = 0.0001|y(t)| + u(t) \quad (5.1)$$

และกำหนดค่า set-point = 5 และ $T_s=0.005$ วินาที โดยผลที่ได้จะแสดงดังตารางที่ 5.1

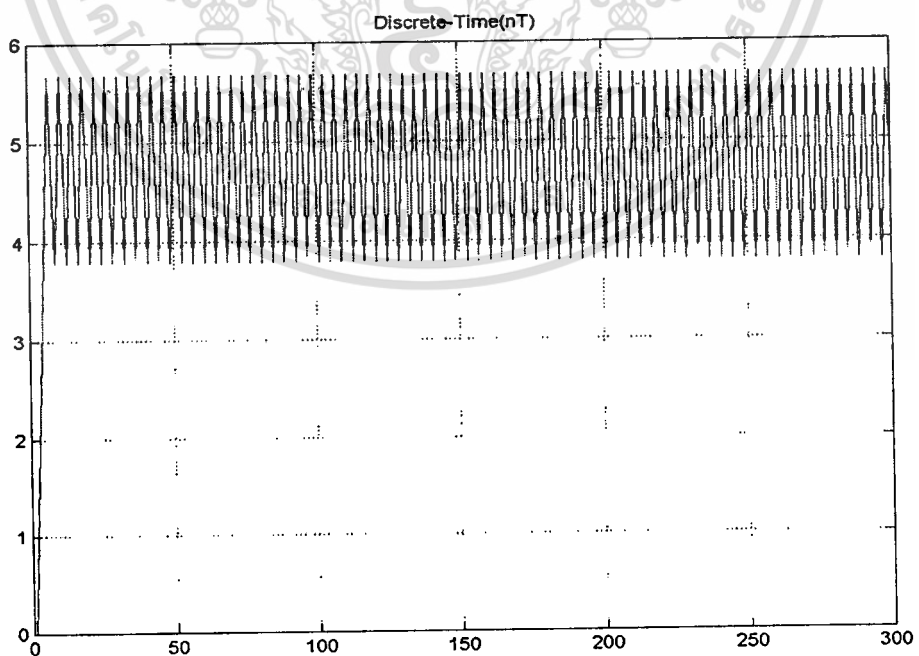
ตารางที่ 5.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟิดีที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

จำนวนประชากรเริ่มต้น	Kp	Kd	Ld	Ku_pd	Rise time (วินาที)	รอบ gen
15	3.0315	3.9040	19.4053	19.5078	0.02	11
30	2.8720	3.9071	19.5692	19.5072	0.02	5
60	2.5781	3.9042	19.0478	19.5175	0.02	5
90	3.1620	3.5211	19.1560	18.8847	0.02	6
120	2.7762	3.1111	18.8214	19.2384	0.02	6

จากตารางที่ 5.1 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีพีดีโดยวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร พบว่า ที่จำนวนประชากรเริ่มต้นเป็น 15 30 60 90 และ 120 ตัวจะมีค่าเวลาได้ขึ้นเท่ากัน คือ 0.02 วินาที ซึ่งผลที่ได้แสดงดังรูปข้างล่างนี้



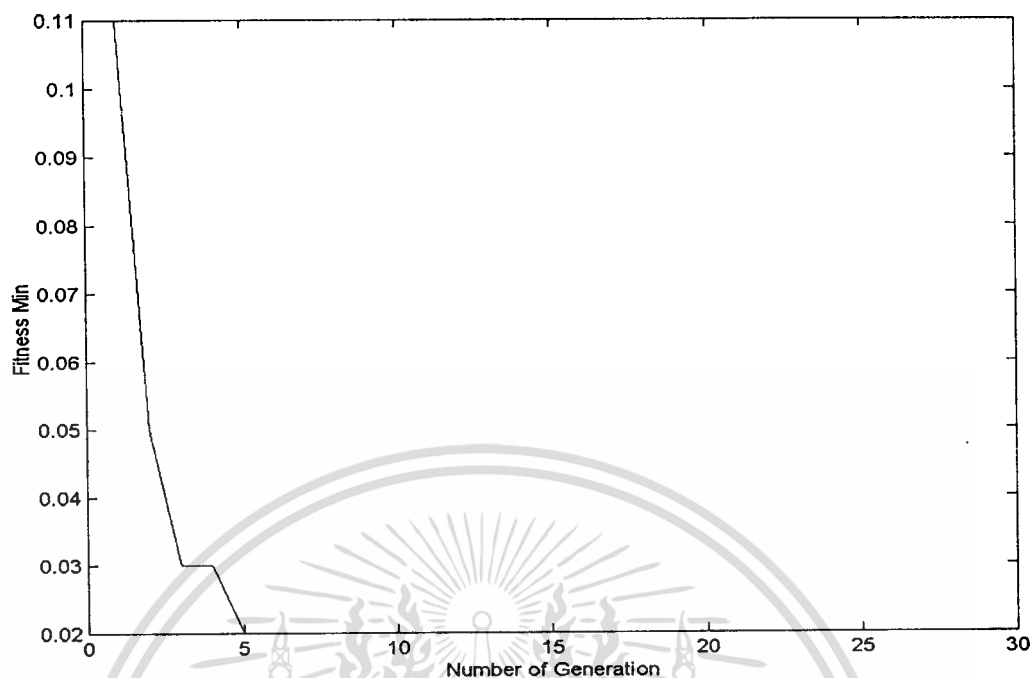
รูปที่ 5.1 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



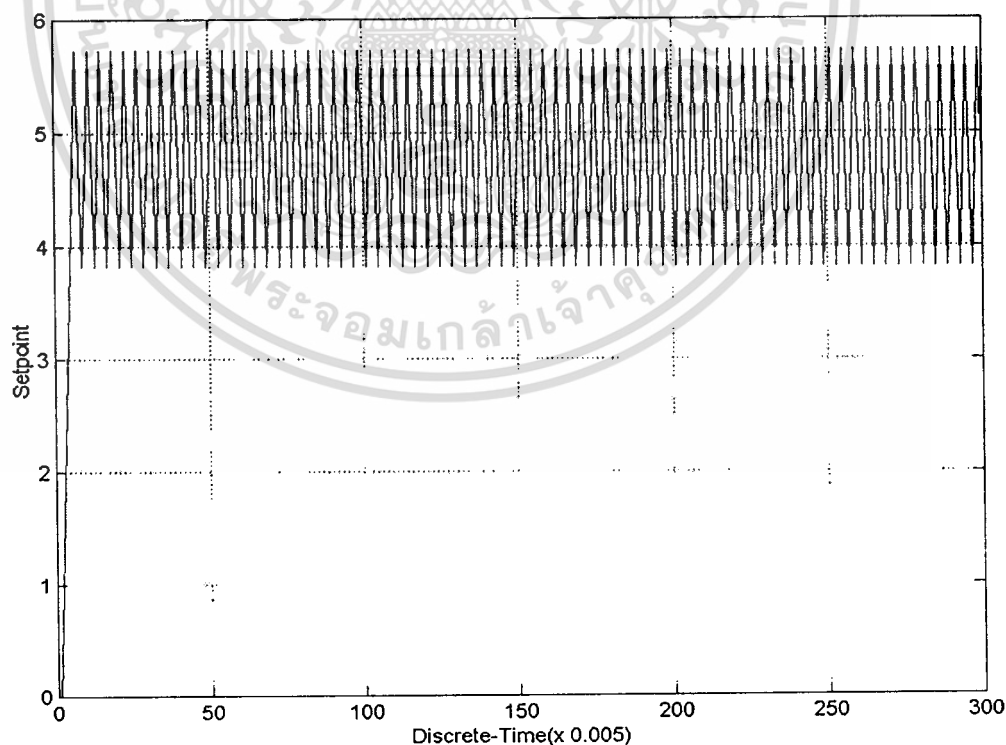
รูปที่ 5.2 ผลตอบสนองของตัวควบคุมแบบฟัซซีพีดีโดยใช้ประชากรเริ่มต้น 15 ตัว ของแบบจำลอง

ที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



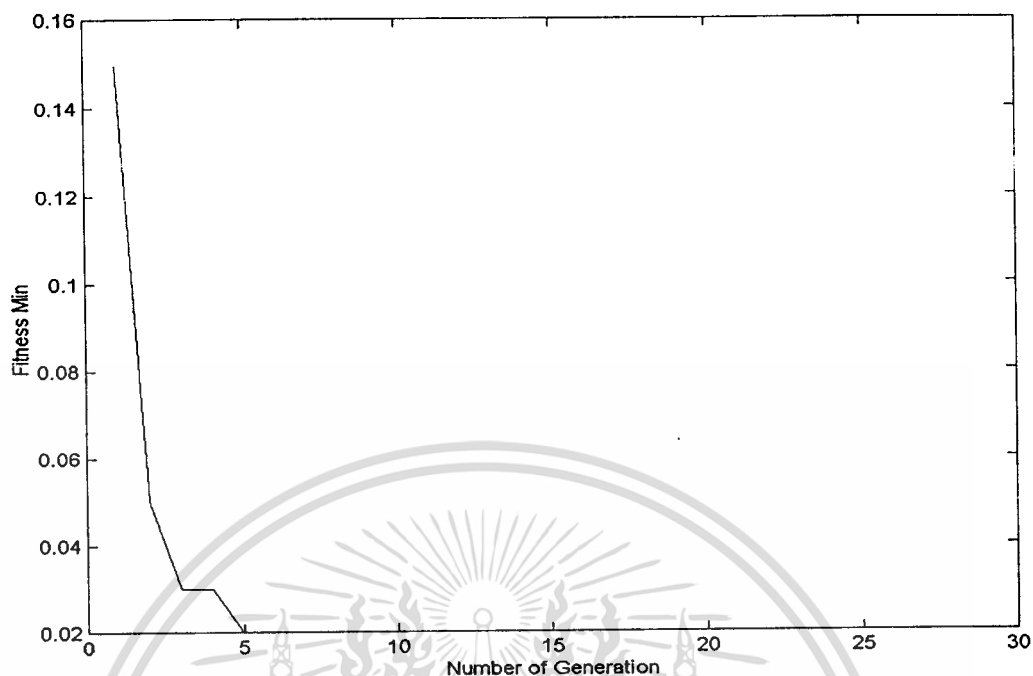
รูปที่ 5.3 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



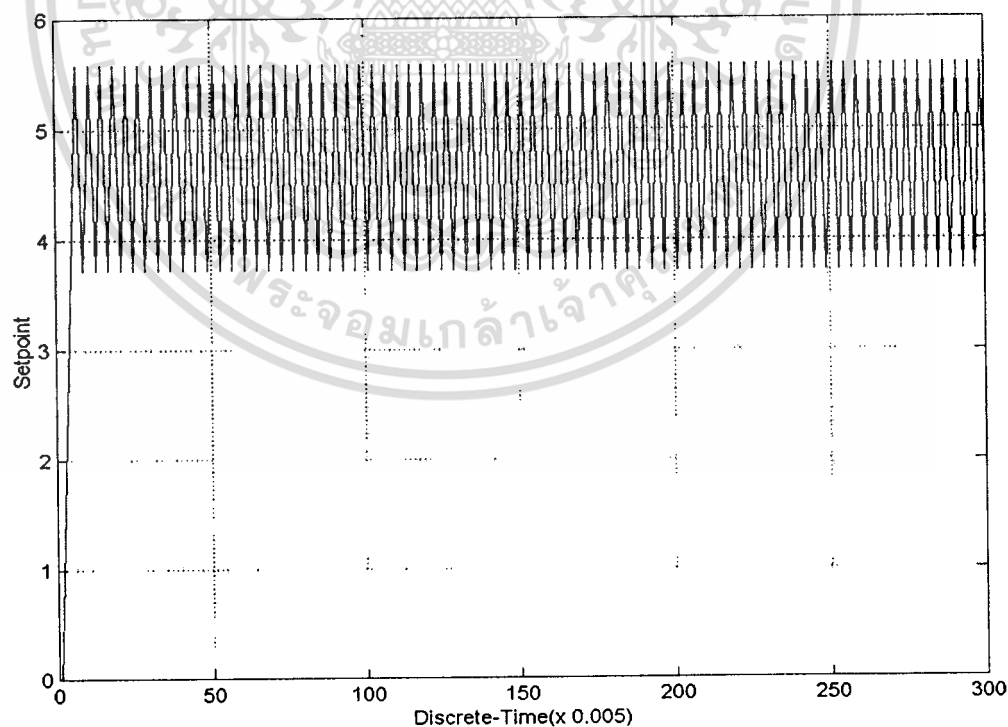
รูปที่ 5.4 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีที่ใช้ประชากรเริ่มต้น 30 ตัวของแบบ

จำลอง ที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

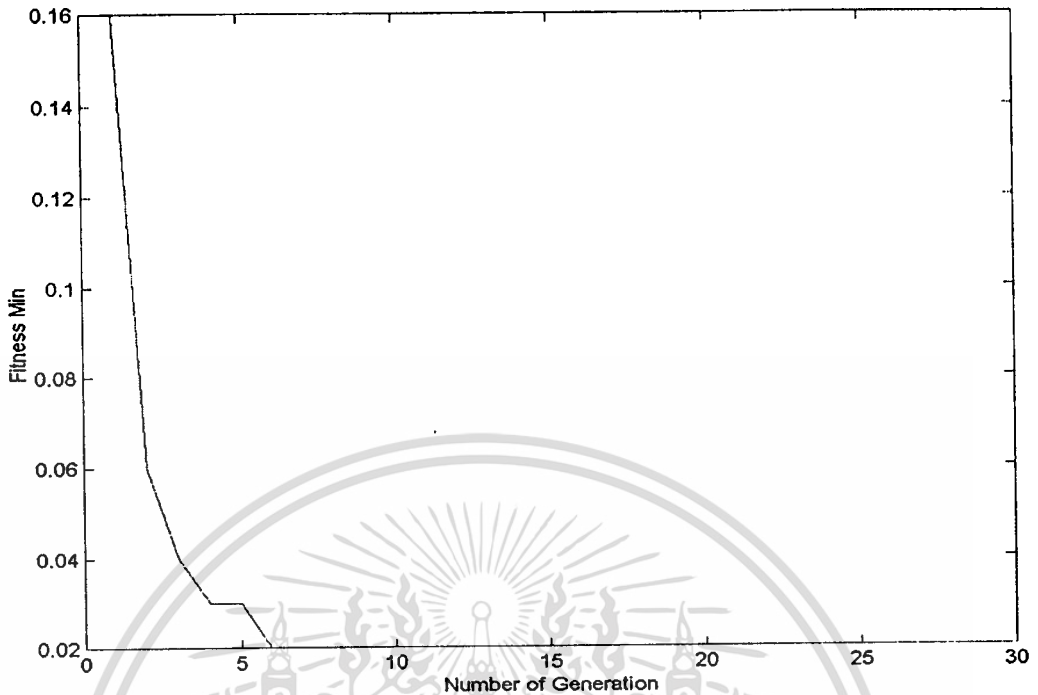


รูปที่ 5.5 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

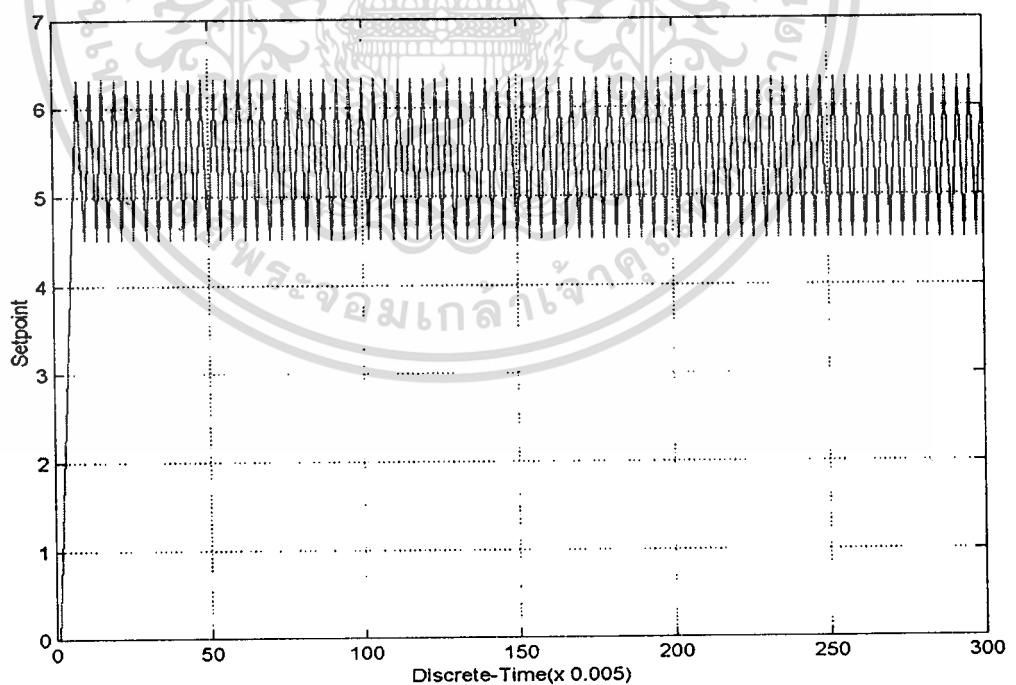


รูปที่ 5.6 ผลตอบสนองของตัวควบคุมแบบฟัซซี่พีดีโดยใช้ประชากรเริ่มต้น 60 ตัวของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

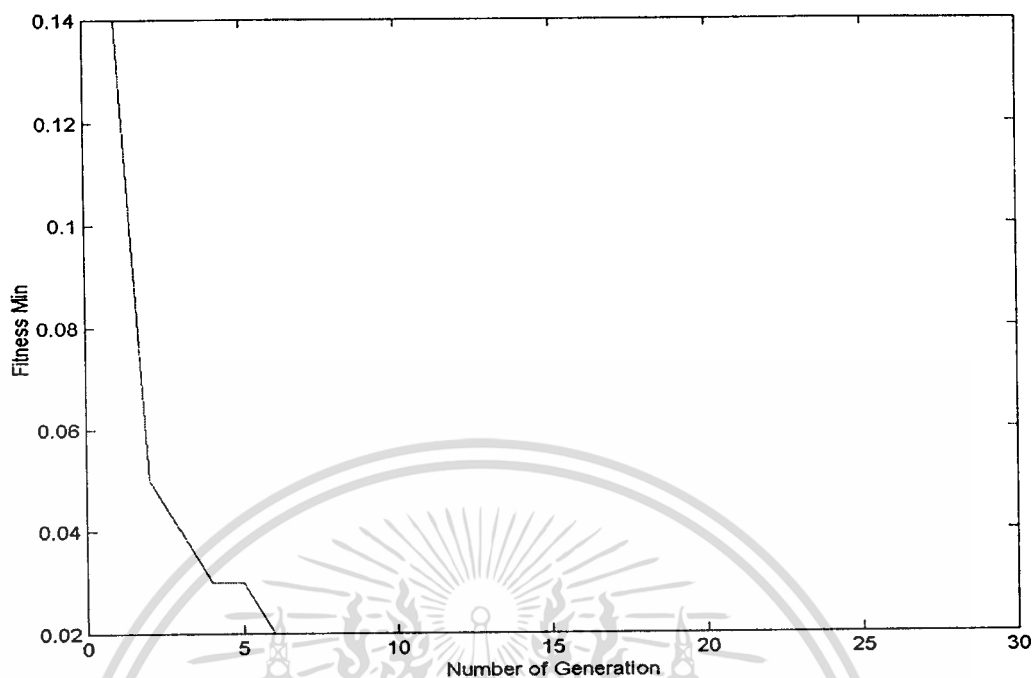


รูปที่ 5.7 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

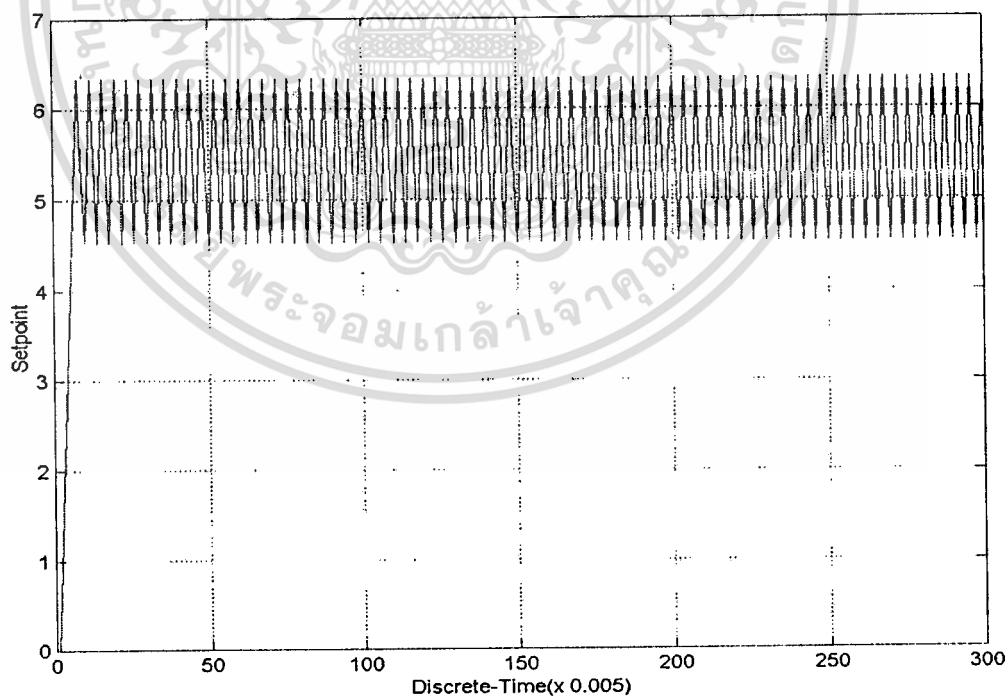


รูปที่ 5.8 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีโดยใช้ประชากรเริ่มต้น 90 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



รูปที่ 5.10 ผลตอบสนองของตัวควบคุมแบบฟัซซี่พีดีโดยใช้ประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

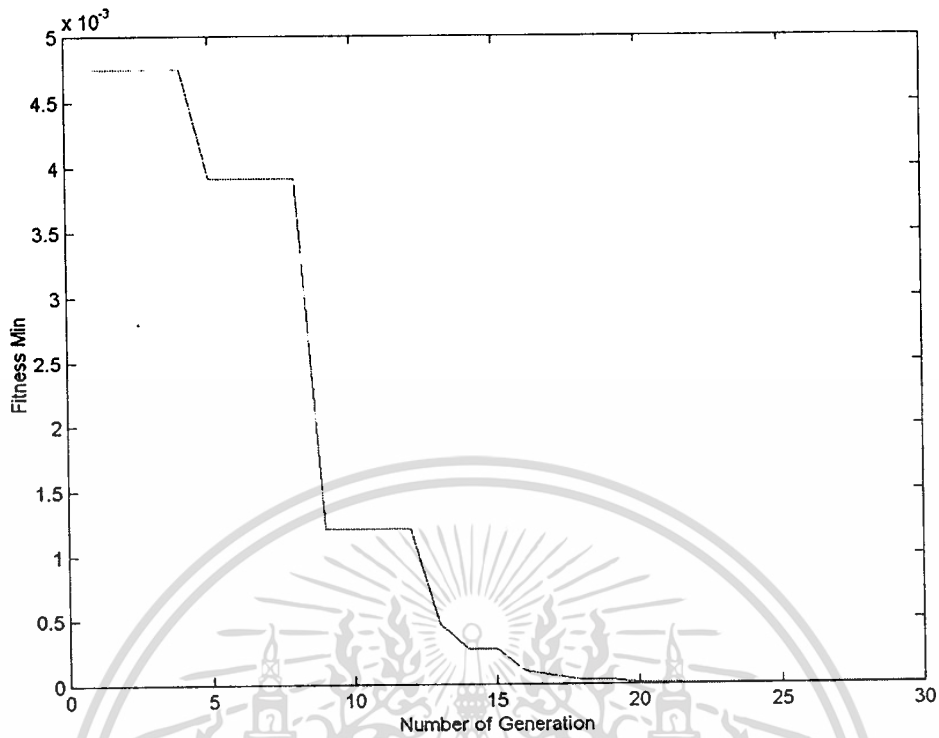
จากรูปที่ 5.1 5.3 5.5 5.7 และ 5.9 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซี่ที่ดีของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัว ใช้จำนวนรอบในการเกิด (generation) มากถึง 11 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 90 และ 120 ตัว พบว่าจำนวนรอบของการเกิดลดลง และที่จำนวนประชากรเป็น 30 ตัวและ 60 ตัวมีจำนวนรอบของการเกิดน้อยที่สุดคือ 5 รอบ การเลือกใช้จำนวนประชากรเริ่มต้นที่เหมาะสมควรใช้ที่ 30 ตัว เนื่องจากจะลดเวลาในการประมวลผล

จากรูปที่ 5.2 5.4 5.6 5.8 และ 5.10 เป็นผลตอบสนองของตัวควบคุมแบบฟัซซี่ที่ดีที่ได้จากการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ พบว่าที่จำนวนประชากรเริ่มต้น 15 30 60 90 และ 120 ตัว จะมีค่าของเวลาไต่ขึ้น (Rise Time) ที่เท่ากันคือ 0.02 วินาที

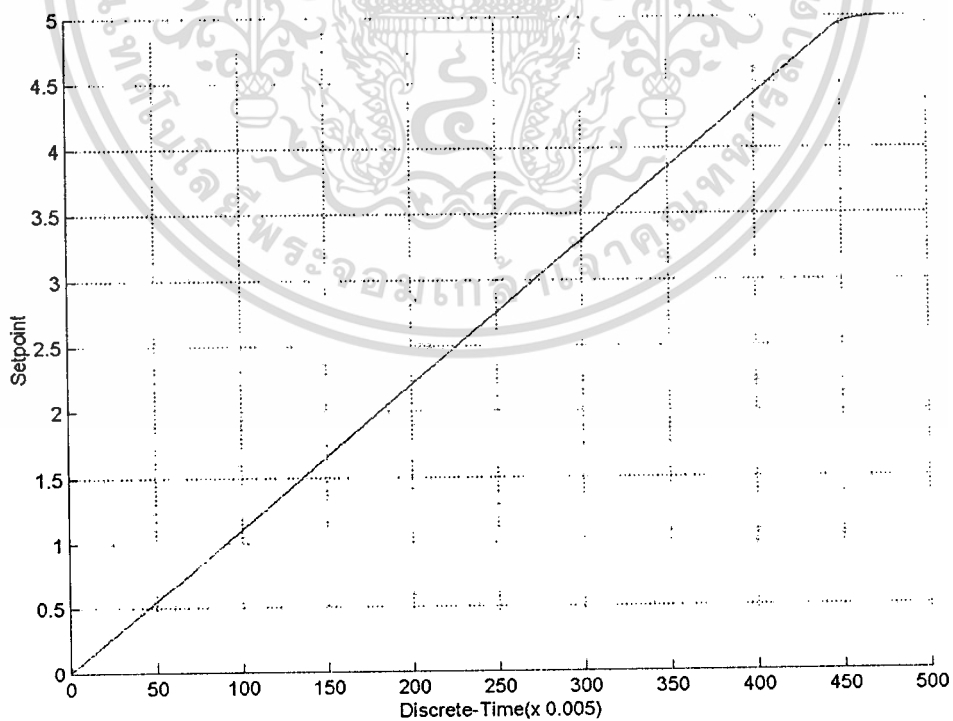
ตารางที่ 5.2 ค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซี่พีไอที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

จำนวนประชากร เริ่มต้น	Kp	Ki	Li	Ku_pi	Max Overshoot	รอบ gen
15	1.0180	0.0392	18.672	1.1792	0	20
30	0.9376	0.0341	2.1019	1.5345	0	3
60	0.9122	0.0328	2.0834	1.3707	0	3
90	0.9212	0.0331	2.0591	1.3711	0	1
120	0.9703	0.0211	2.0182	1.3158	0	1

จากตารางที่ 5.2 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซี่พีไอโดยวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร พบว่าที่จำนวนประชากรเริ่มต้น 15 30 60 90 และ 120 ตัวจะมีค่าฟุ้งเกินที่เท่ากับ 0

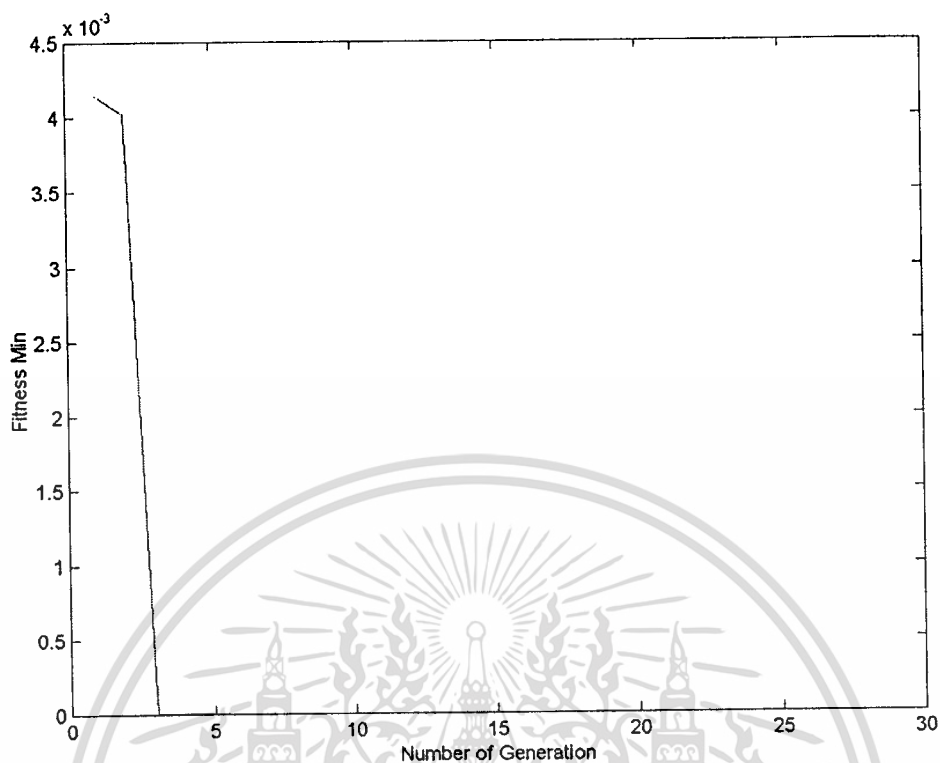


รูปที่ 5.11 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

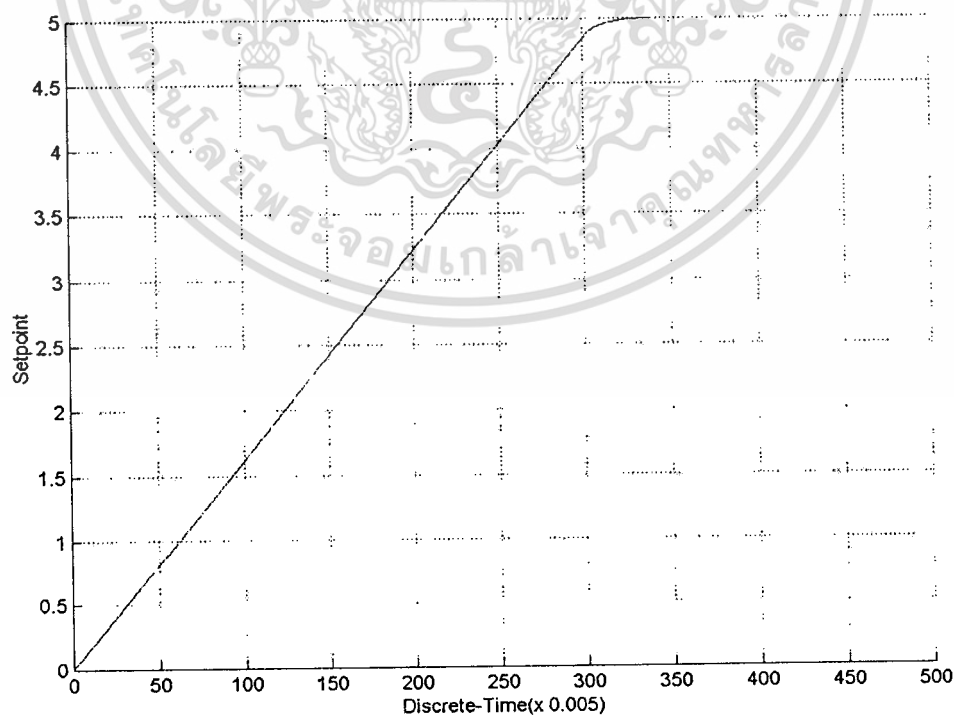


รูปที่ 5.12 ผลตอบสนองของตัวควบคุมแบบฟัซซีพีไอโดยใช้ประชากรเริ่มต้น 15 ตัว ของ

เอกสารนี้เป็นเอกสารแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

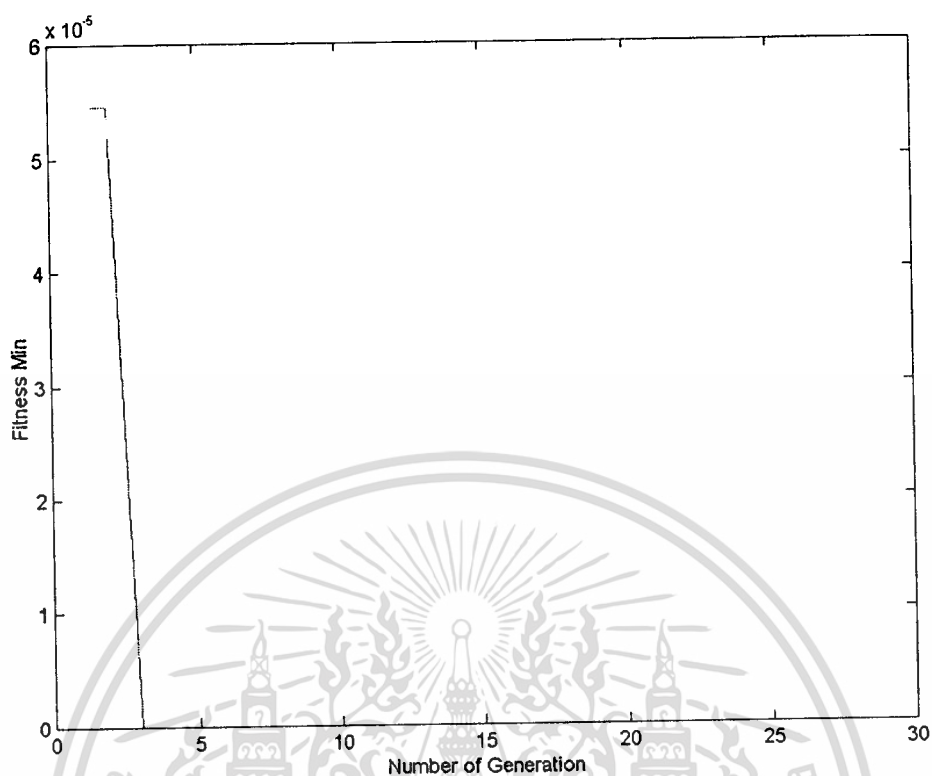


รูปที่ 5.13 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของ แบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

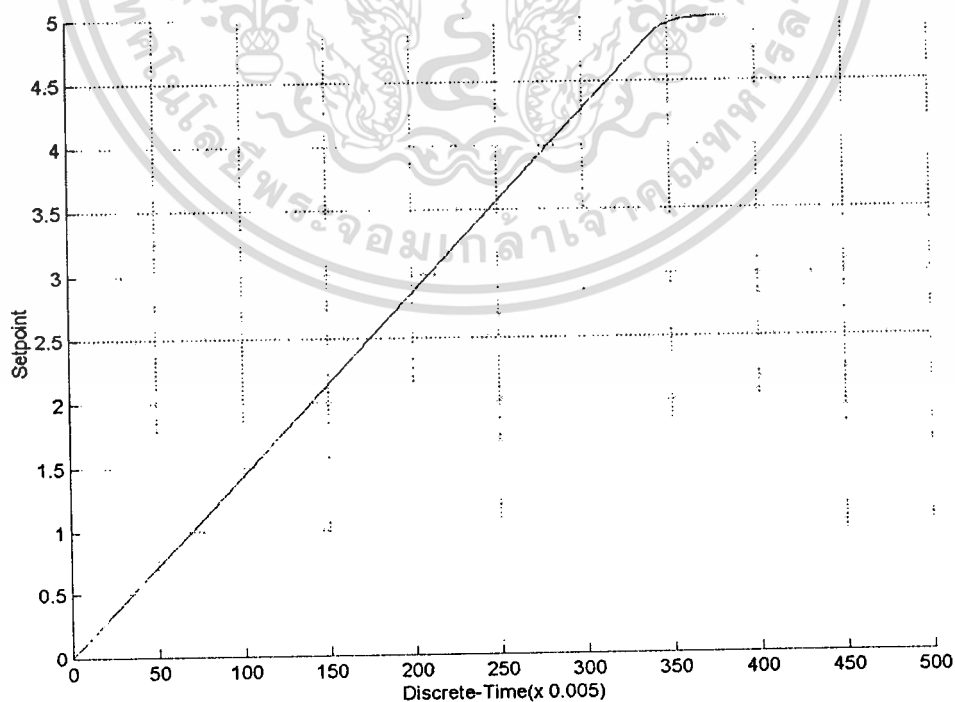


รูปที่ 5.14 ผลตอบสนองของตัวควบคุมแบบฟัซซีพีโอโดยให้ประชากรเริ่มต้น 30 ตัว ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

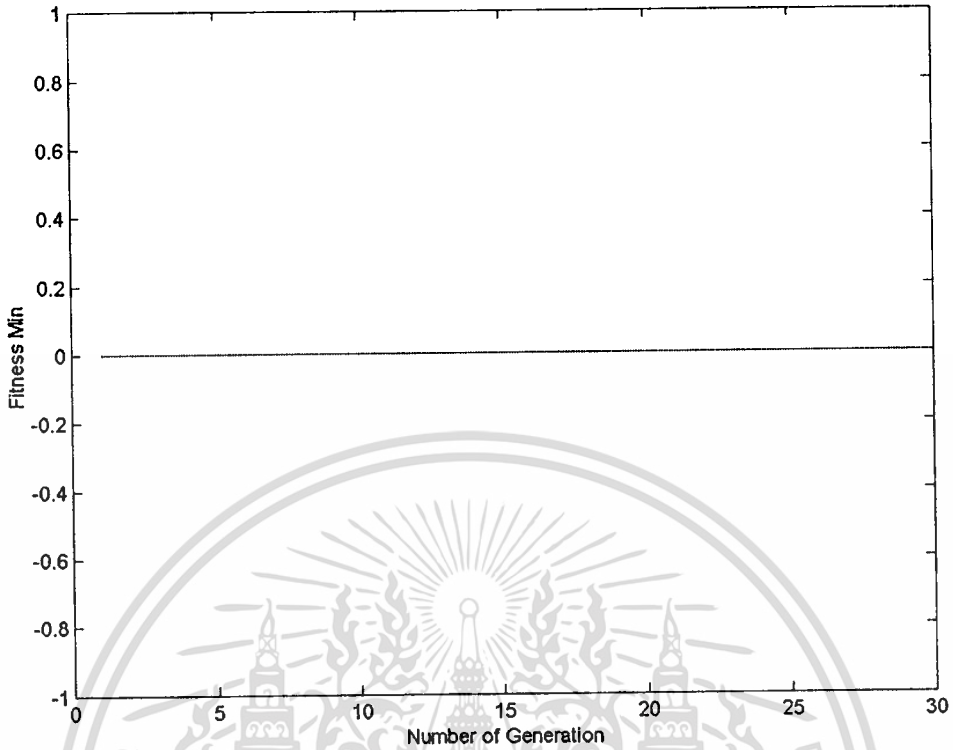


รูปที่ 5.15 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

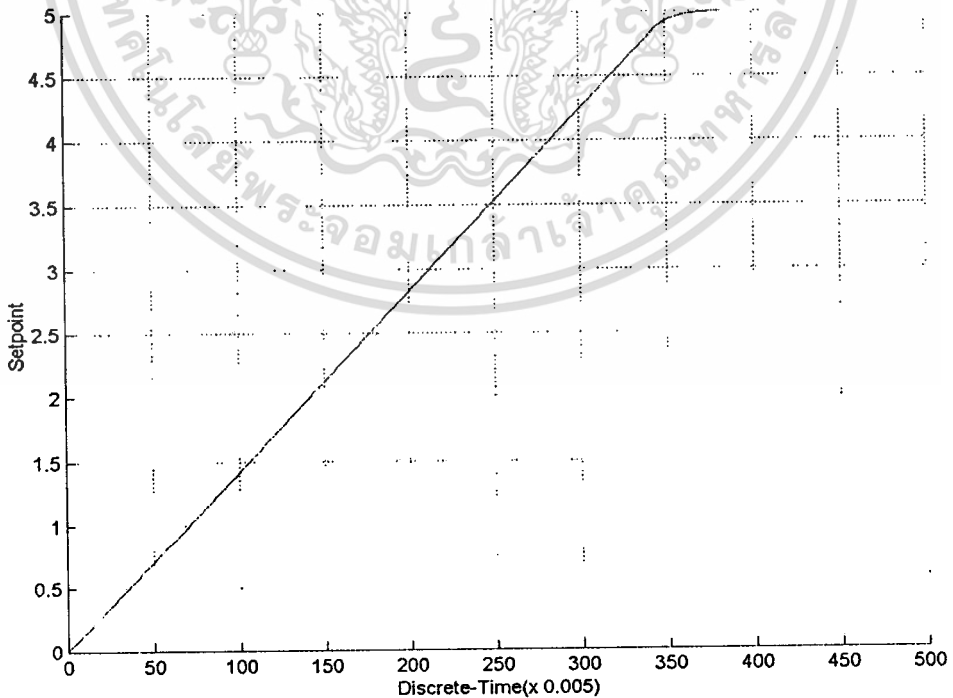


รูปที่ 5.16 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 60 ตัว ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

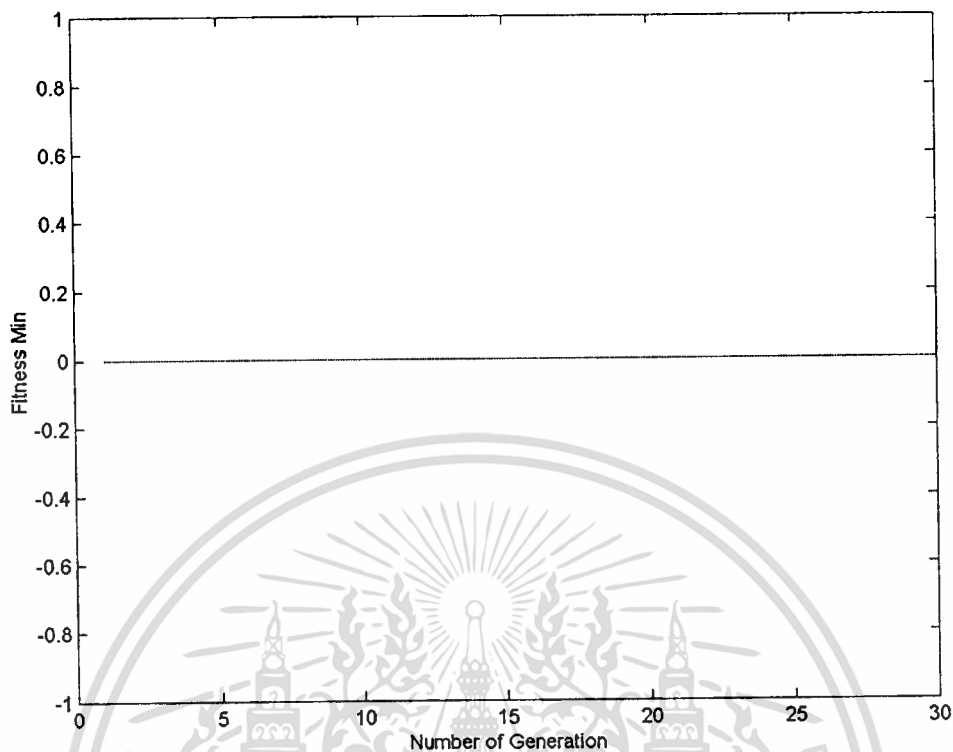


รูปที่ 5.17 แนวโน้มของการดูเข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

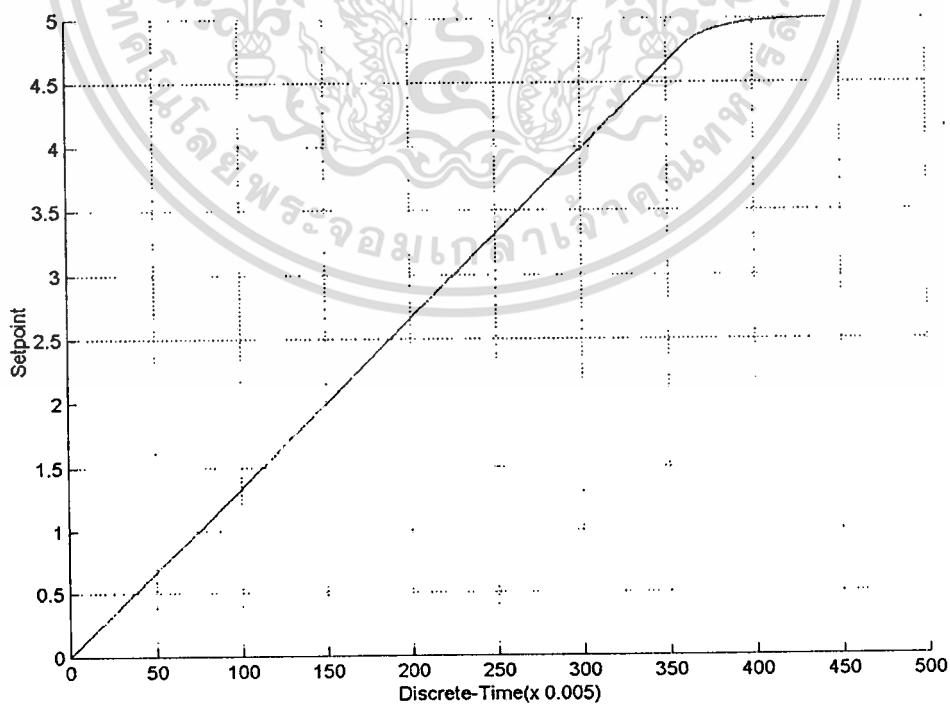


รูปที่ 5.18 ผลตอบสนองของตัวควบคุมแบบฟัซซีพีไอโดยใช้ประชากรเริ่มต้น 90 ตัว ของ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.19 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



รูปที่ 5.20 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 120 ตัว ของ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
แบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

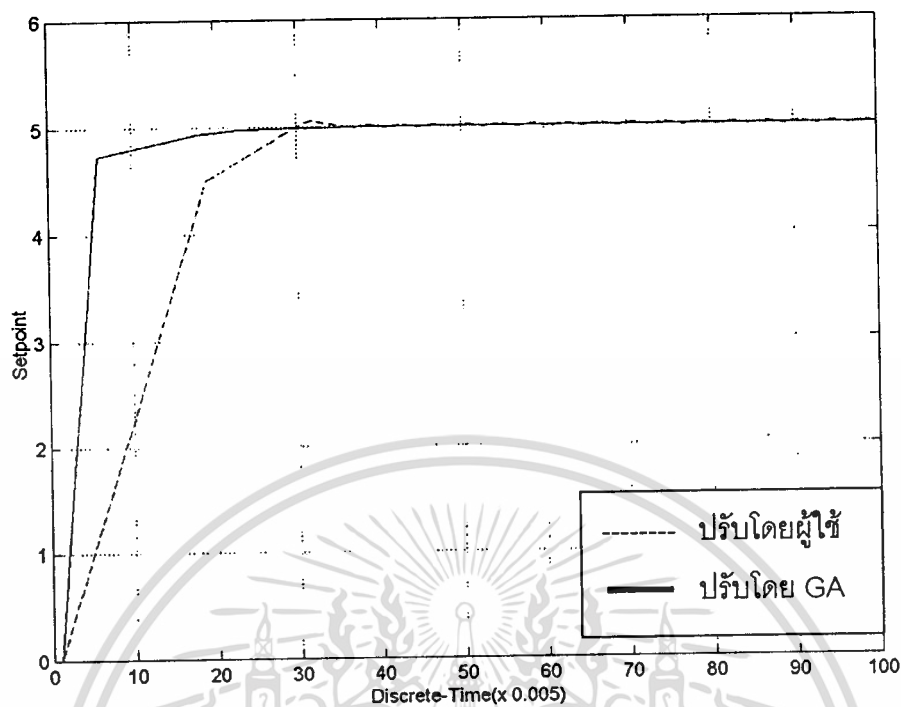
รูปที่ 5.11 5.13 5.15 5.17 และ 5.19 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีพีไอ ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัว จะใช้จำนวนรอบในการเกิด (generation) มากถึง 20 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 90 และ 120 ตัวจะพบว่าจำนวนรอบของการเกิดจะลดลง ซึ่งจำนวนรอบของการเกิดไม่มีความแตกต่างกัน ดังนั้นจำนวนประชากรเริ่มต้นที่เหมาะสมควรจะใช้ที่ 30 เนื่องจากจะลดเวลาในการประมวลผล

จากรูปที่ 5.12 5.14 5.16 5.18 และ 5.20 เป็นผลตอบสนองของตัวควบคุมแบบฟัซซีพีไอ ที่ได้จากการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ พบว่าที่จำนวนประชากรเริ่มต้น 15 30 60 90 และ 120 ตัวจะมีค่าของ overshoot ที่เท่ากันคือ 0 ซึ่งเป็นค่าที่ดีที่สุด

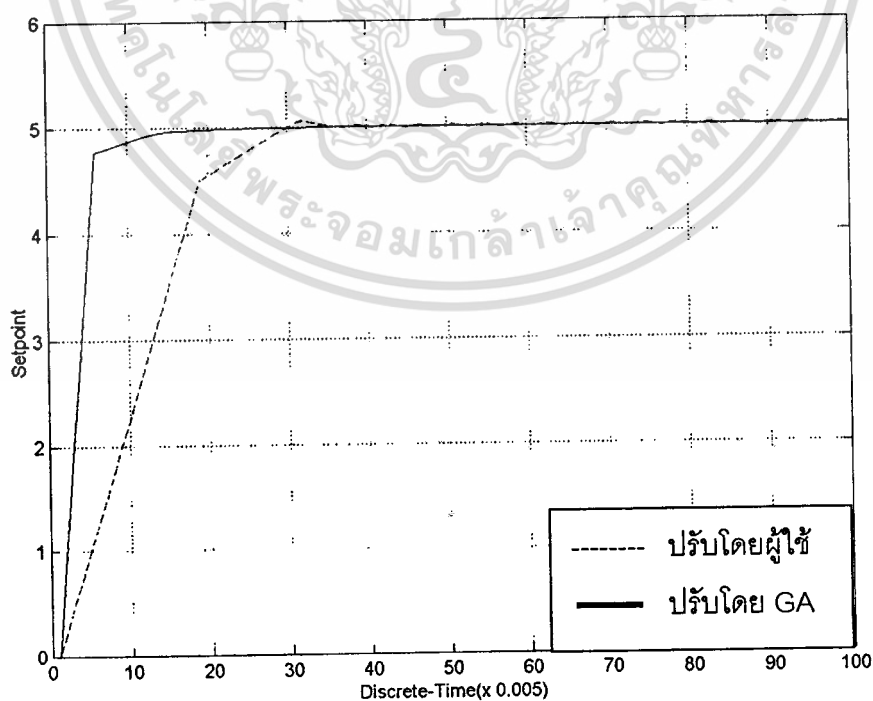
ตารางที่ 5.3 ค่าพารามิเตอร์ที่ปรับโดยผู้ให้ และปรับโดยวิธีทางพันธุศาสตร์ ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

ค่าพารามิเตอร์	Fuzzy PD				Fuzzy PI			
	Kp	Kd	Ku_pd	Ld	Kp	Ki	Ku_pi	Li
โดยผู้ให้	2	3	10	10	1	1	3	2
GA-15	3.0315	3.9040	19.4053	19.5078	1.0180	0.0392	1.8672	1.1792
GA-30	2.8720	3.9071	19.5692	19.5072	0.9376	0.0341	2.1019	1.5345
GA-60	2.5781	3.9042	19.0478	19.5175	0.9122	0.0328	2.0834	1.3707
GA-90	3.1620	3.5211	19.1560	18.8847	0.9212	0.0331	2.0591	1.3717
GA-120	2.7762	3.1111	18.8214	19.2384	0.9703	0.0211	2.0182	1.3158

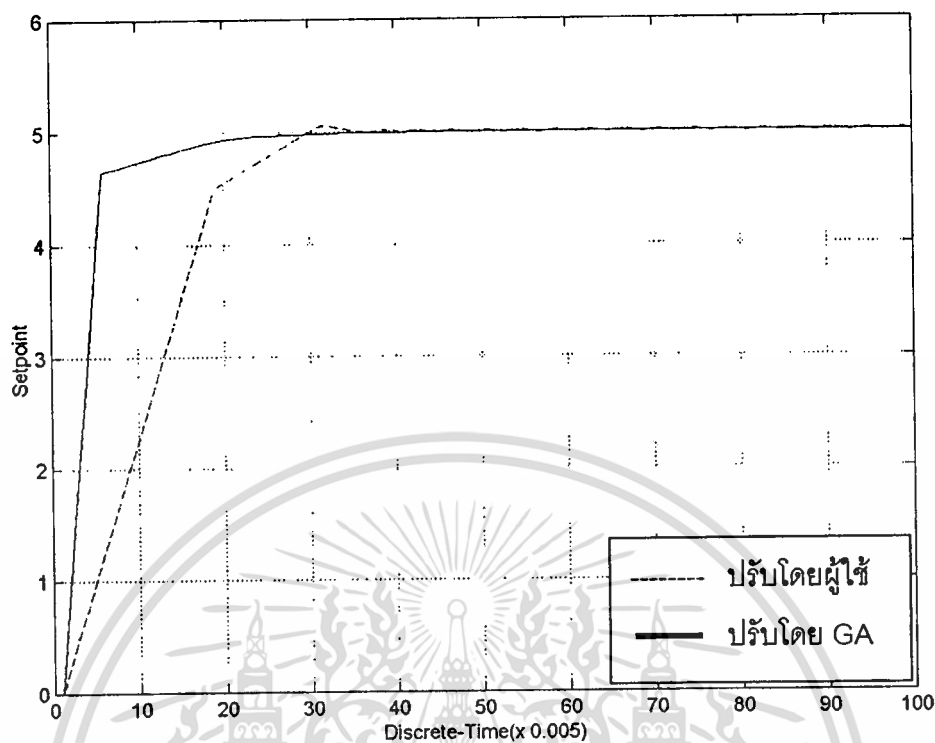
จากตารางที่ 5.3 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีพีเอสแควร์ไอดี เปรียบเทียบกับการปรับโดยผู้ให้ ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร ด้วยวิธีทางพันธุศาสตร์ พบว่าการปรับแต่งด้วยวิธีทางพันธุศาสตร์จะมีค่าพารามิเตอร์ที่ละเอียดมากกว่า



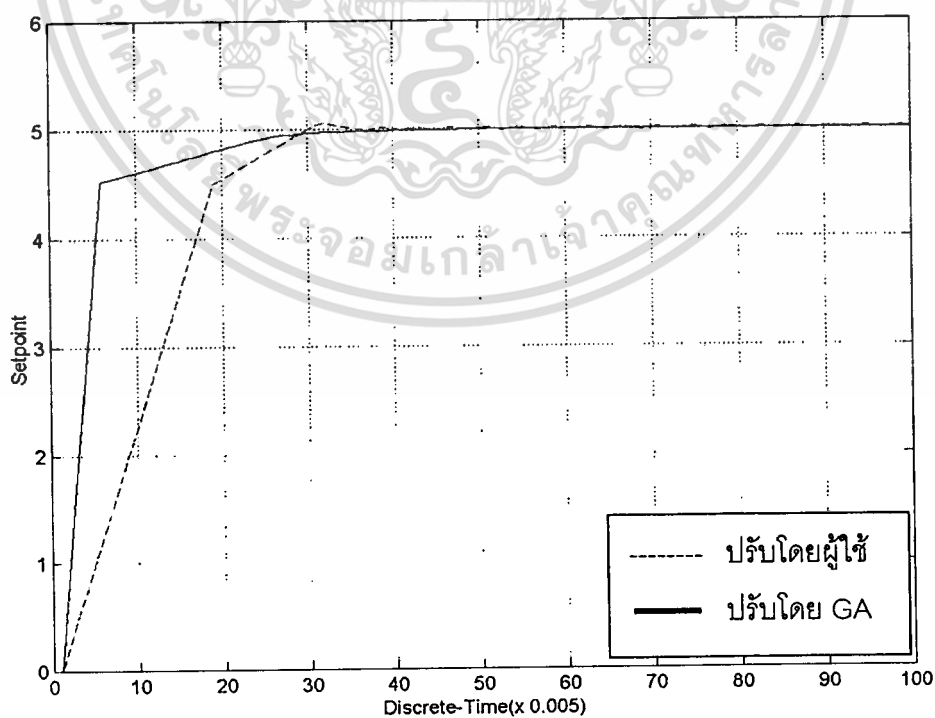
รูปที่ 5.21 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-15 ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



รูปที่ 5.22 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-30

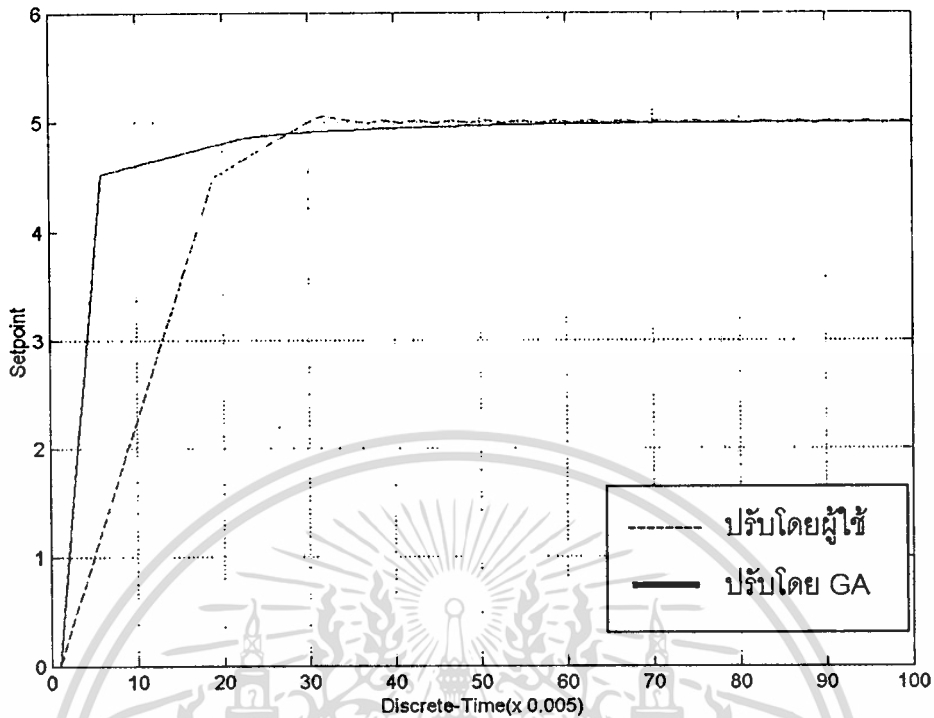


รูปที่ 5.23 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-60 ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร



รูปที่ 5.24 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.25 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-120 ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

จากรูปที่ 5.21 5.22 5.23 5.24 และ 5.25 เป็นการเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดยวิธีทางพันธุศาสตร์ ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 8 ตัวแปร พบว่าผลตอบสนองของการแต่งโดยวิธีทางพันธุศาสตร์จะดีกว่าการปรับแต่งโดยผู้ใช้ และประสิทธิภาพที่ได้จะแสดงดังตารางที่ 5.4

ตารางที่ 5.4 ประสิทธิภาพของการปรับแต่งของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 8 ตัวแปร

วิธีการ	Rise-time(sec)	Setting-time(sec)	Overshoot (%)
ผู้ใช้	0.08	0.1394	0.011
GA-15	0.02	0.0809	0
GA-30	0.02	0.0568	0
GA-60	0.02	0.0895	0
GA-90	0.02	0.12	0
GA-120	0.02	0.1425	0

จากตารางที่ 5.4 พบว่า การปรับแต่งโดยวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น จะมีค่า Rise-time , Setting-time และ Overshoot ที่น้อยกว่าการปรับโดยผู้ใช้ แสดงว่าการใช้วิธีทางพันธุศาสตร์จะดีกว่า

5.1.2 แบบจำลองหุ่นยนต์เคลื่อนที่ (Mobile Robot Model)

การจำลองนี้จะเป็นการทดสอบหาค่าแนวโน้มของการลู่เข้าของโปรแกรม โดยใช้ประชากรเริ่มต้นที่แตกต่างกัน และอีกส่วนจะเป็นการเปรียบเทียบกันระหว่างการปรับแต่งโดยผู้ใช้งานกับการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ ซึ่งเป็นการปรับมุมเลี้ยวของหุ่นยนต์ สมการที่ใช้ดังสมการที่ (5.2)

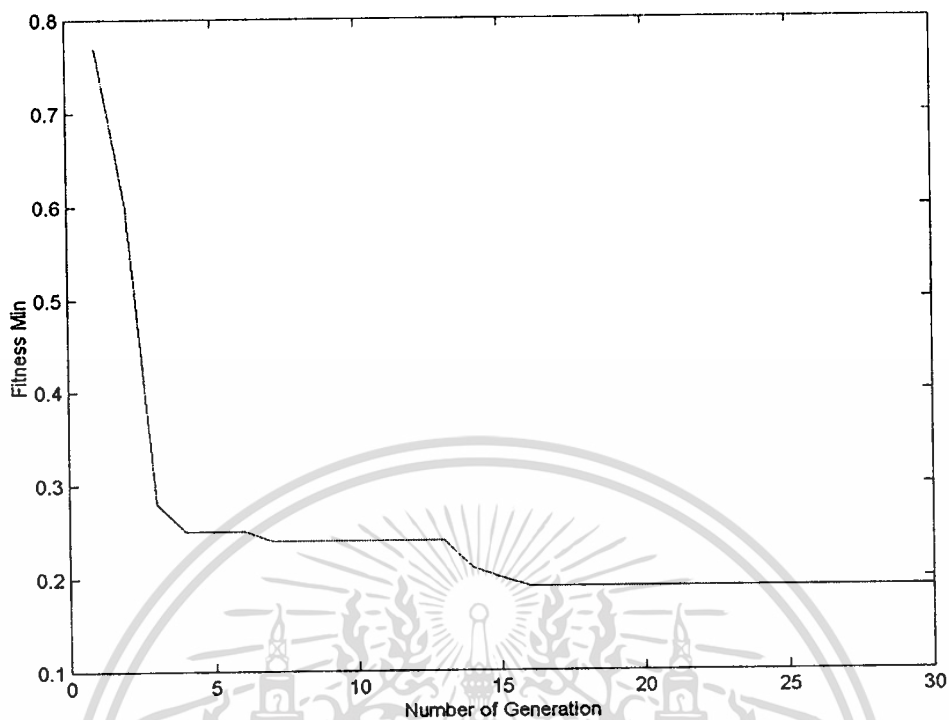
$$\theta(k+1) = \theta(k) + \frac{v_R(k) - v_L(k)}{D} \times Ts \quad (5.2)$$

และกำหนดค่า set-point = 90 องศา D=0.067 เมตร และ $T_s=0.01$ วินาที สำหรับค่าพารามิเตอร์ของตัวควบคุมแสดงดังตารางที่ 5.5

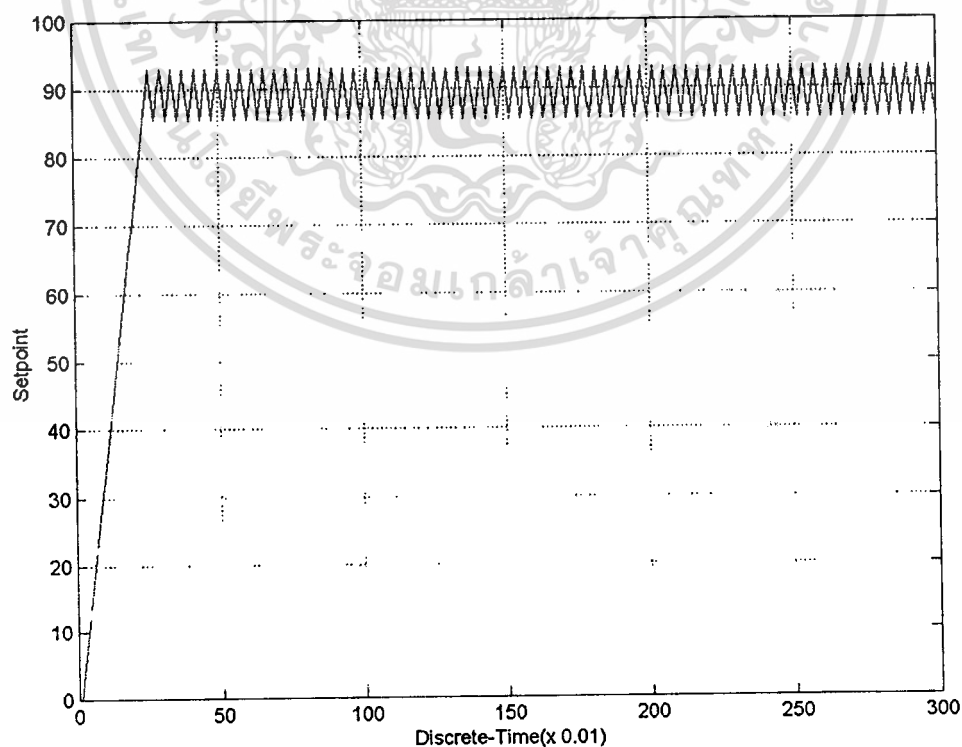
ตารางที่ 5.5 ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีดีที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

จำนวนประชากรเริ่มต้น	Kp	Kd	Ld	Ku_pd	Rise time (วินาที)	รอบ gen
15	2.1496	2.3893	4.9970	4.9844	0.19	17
30	2.4378	1.8337	4.9789	4.9963	0.19	13
60	2.8845	1.1201	4.9775	4.9885	0.19	11
90	1.5644	2.0621	4.9745	4.9774	0.19	9
120	1.2527	1.9868	4.9951	4.9961	0.19	10

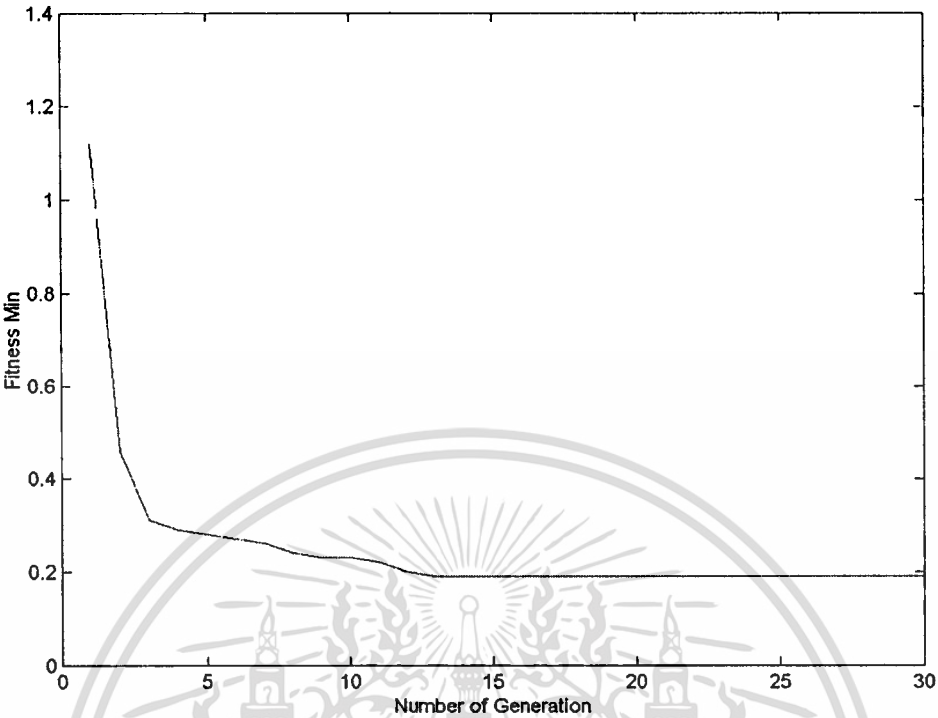
จากตารางที่ 5.5 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีดีโดยวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร พบว่า ที่จำนวนประชากรเริ่มต้นเป็น 15 30 60 90 และ 120 ตัวจะมีค่าเวลาไถ่ขึ้นเท่ากัน คือ 0.19 วินาที ซึ่งผลที่ได้จะแสดงดังรูปต่อไปนี้



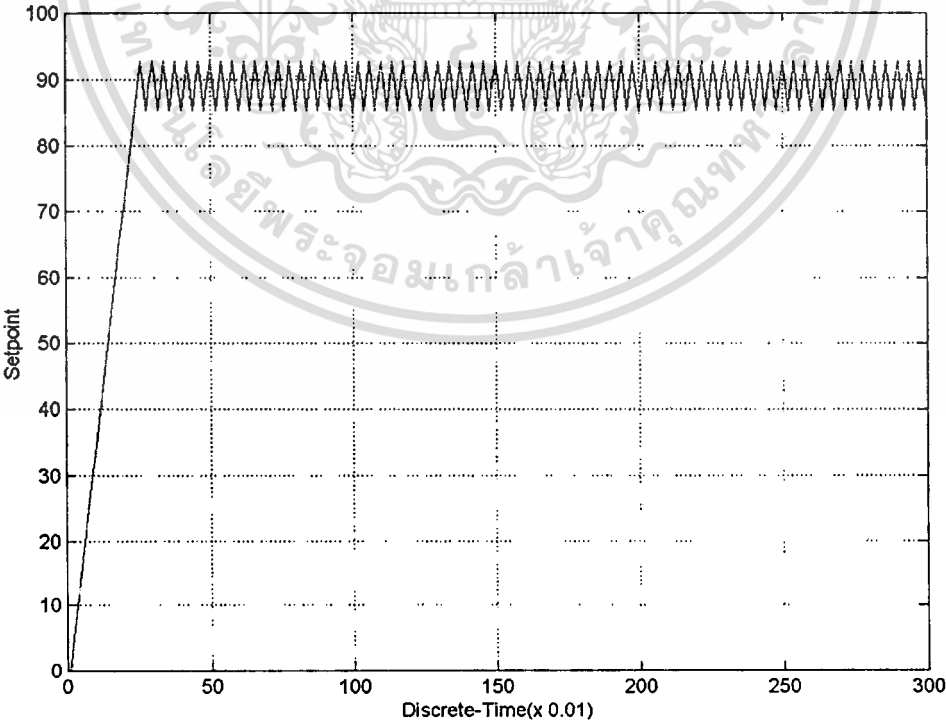
รูปที่ 5.26 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



รูปที่ 5.27 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีโดยใช้ประชากรเริ่มต้น 15 ตัว ของแบบ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
จำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

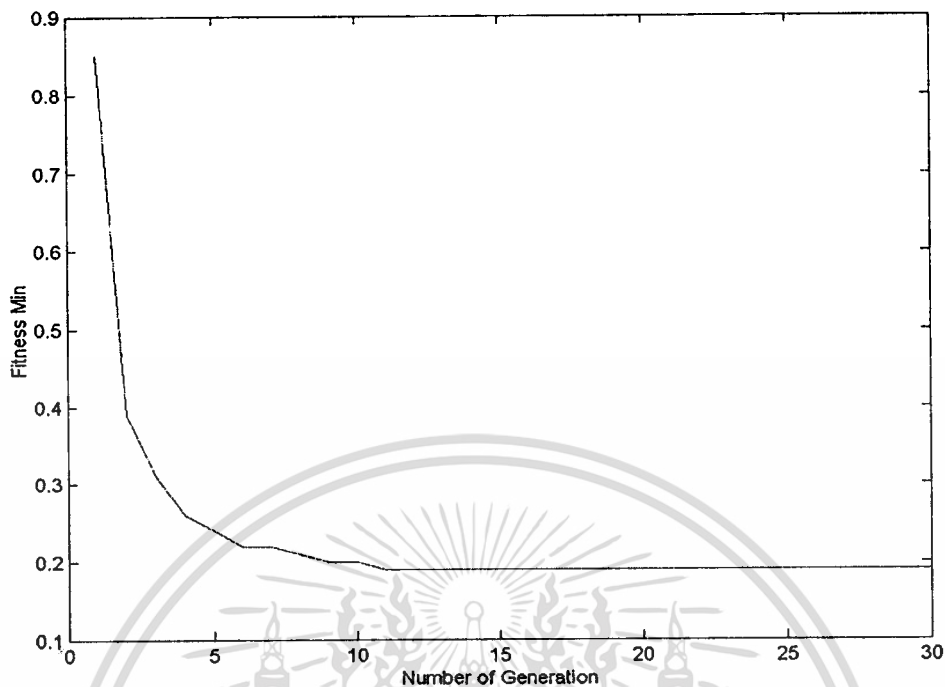


รูปที่ 5.28 แนวโน้มของการสุ่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

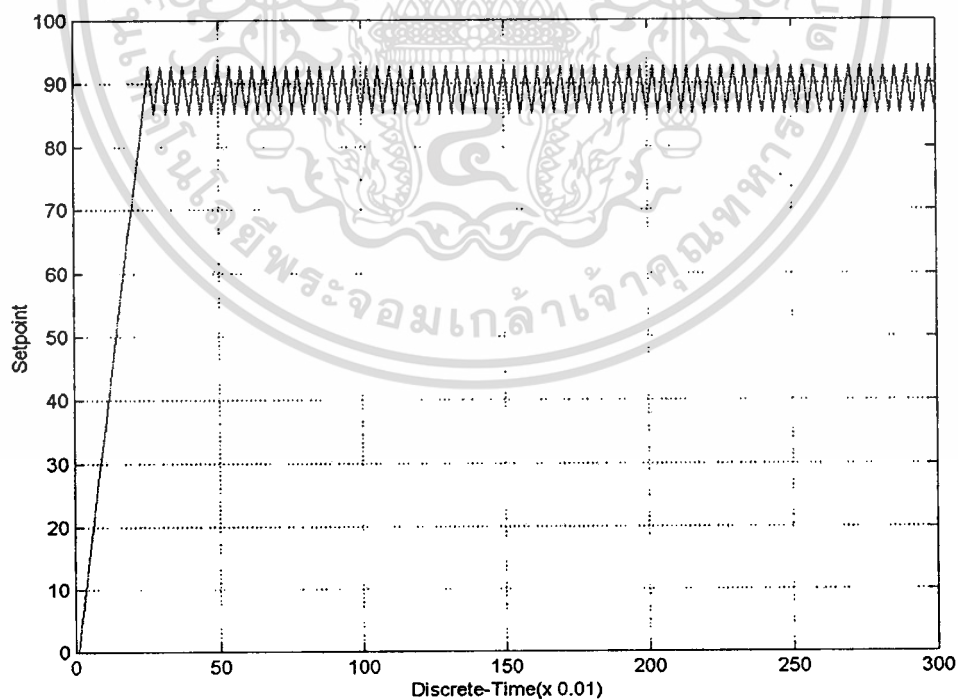


รูปที่ 5.29 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีโดยใช้ประชากรเริ่มต้น 30 ตัว ของแบบ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ห้ามลอกเลียนแบบหรือเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

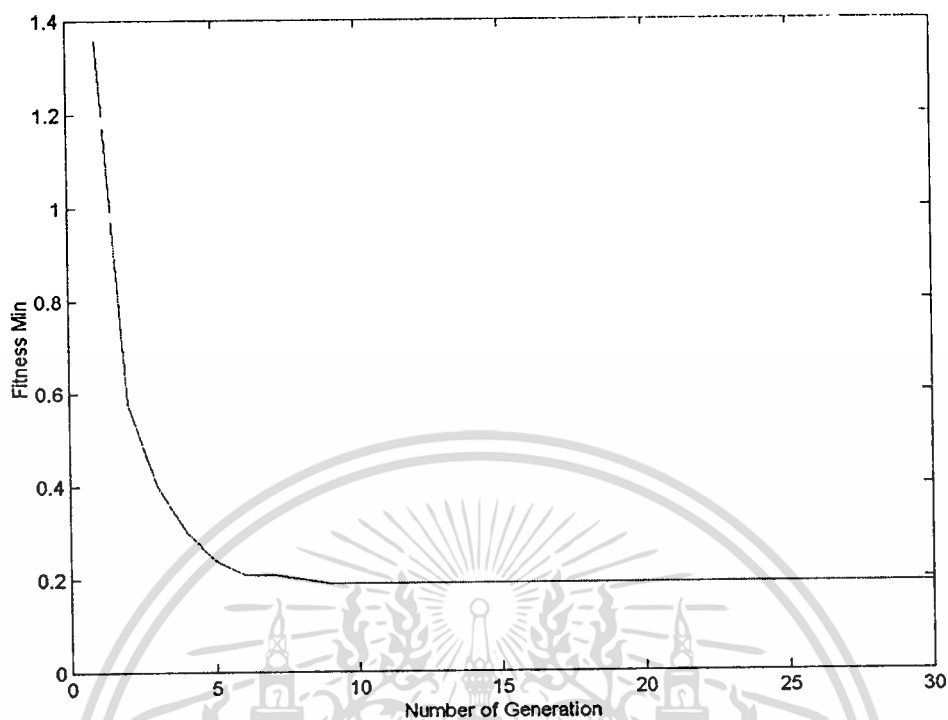


รูปที่ 5.30 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

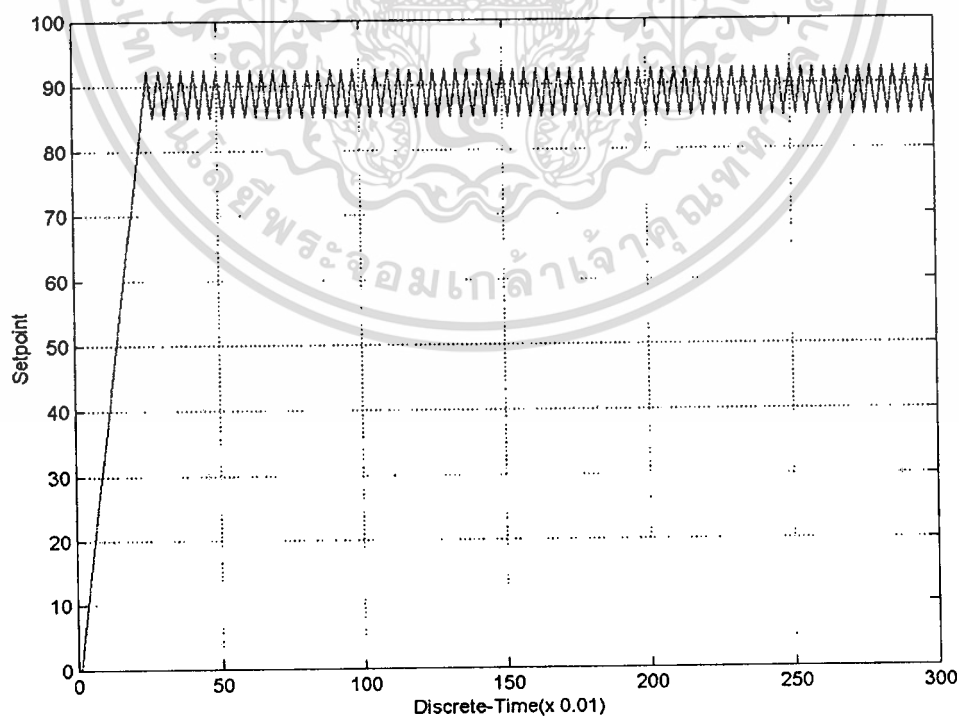


รูปที่ 5.31 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีโดยใช้ประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

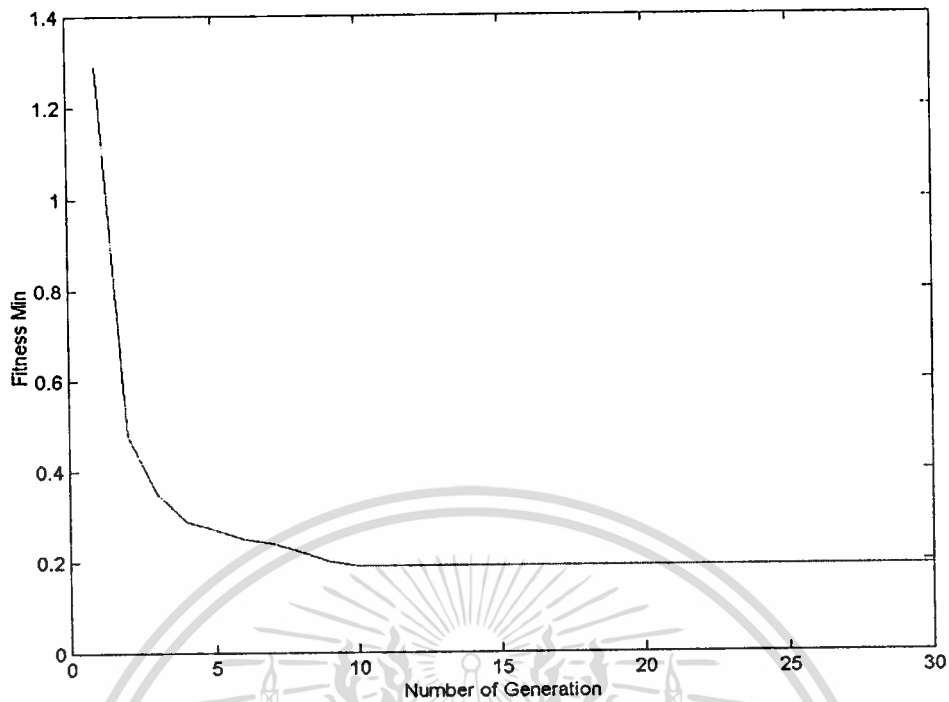


รูปที่ 5.32 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

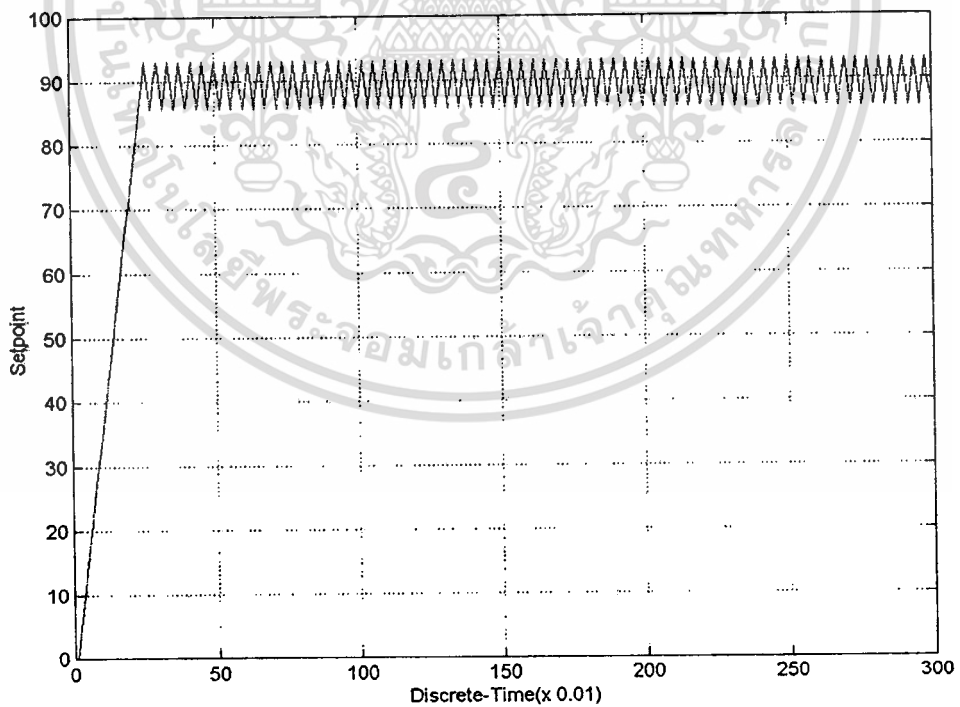


รูปที่ 5.33 ผลตอบสนองของตัวควบคุมแบบพีซีพีดีโดยใช้ประชากรเริ่มต้น 90 ตัว ของแบบ

เอกสารนี้เป็นเอกสารจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปรเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.34 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



รูปที่ 5.35 ผลตอบสนองของตัวควบคุมแบบฟัซซีฟิตีโดยใช้ประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

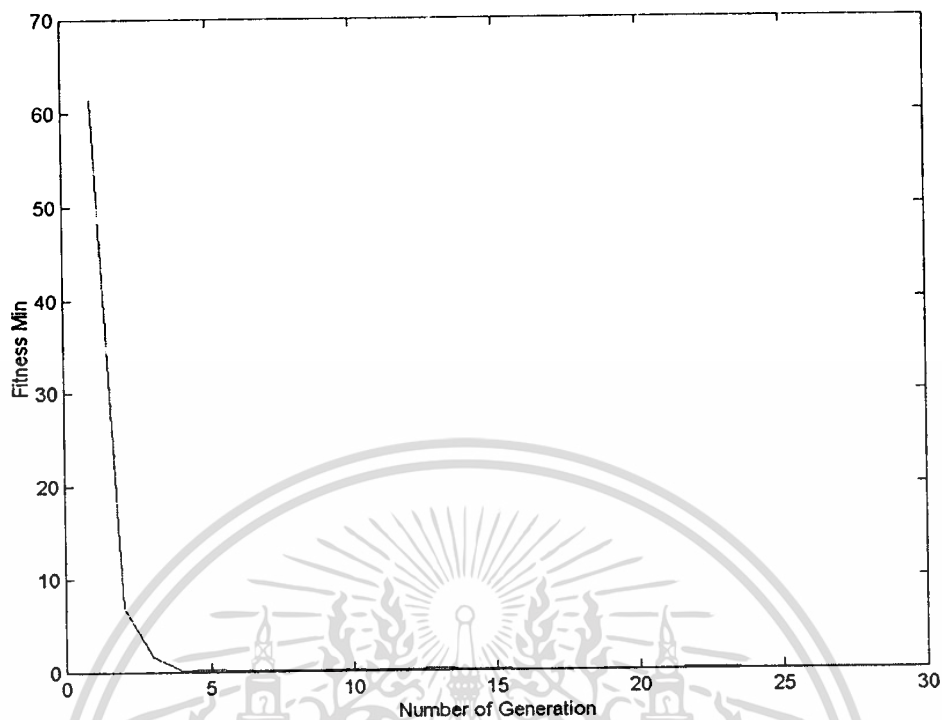
จากรูปที่ 5.26 5.28 5.30 5.32 และ 5.34 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีดีของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัว จะใช้จำนวนรอบในการเกิด (generation) มากถึง 17 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 90 และ 120 ตัว จะพบว่าจำนวนรอบของการเกิดจะลดลง คือ มีค่า 13 11 9 และ 10 รอบตามลำดับ จะเห็นว่าจะมีค่าที่ไม่แตกต่างกันมาก แต่การเลือกจำนวนประชากรเริ่มต้นไปใช้นั้น จะต้องคำนึงถึงความเร็วในการประมวลผลเป็นสำคัญ ดังนั้นที่จำนวนประชากรเริ่มต้นที่น้อยกว่าจะทำให้การประมวลผลเร็วขึ้น ซึ่งจำนวนประชากรเริ่มต้นที่ 30 ตัวจึงมีความเหมาะสมที่สุด

จากรูปที่ 5.27 5.29 5.31 5.33 และ 5.35 เป็นผลตอบสนองของตัวควบคุมแบบพีซีพีดีที่ได้จากการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ พบว่าที่จำนวนประชากรเริ่มต้น 15 30 60 90 และ 120 ตัวจะมีค่าของเวลาไต่ขึ้น (rise time) ที่เท่ากันคือ 0.19 วินาที

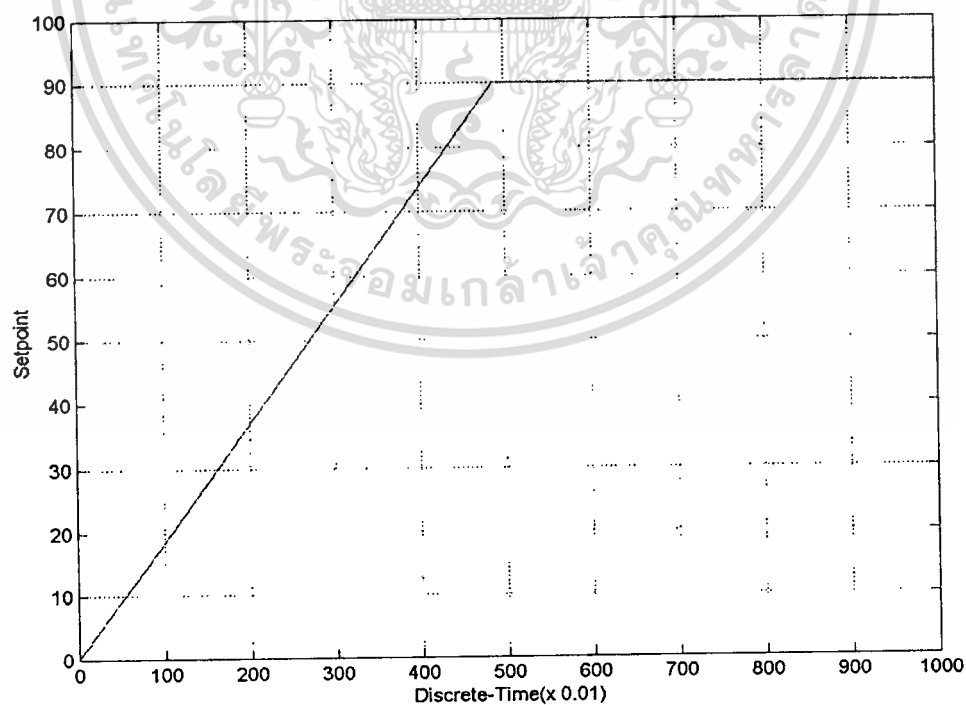
ตารางที่ 5.6 ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีไอที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

จำนวนประชากรเริ่มต้น	Kp	Ki	Li	Ku_pi	Max Overshoot	รอบ gen
15	4.0846	3.4263	0.8776	0.7048	0	24
30	1.6269	6.2038	0.9756	0.6181	0	20
60	4.7183	5.8345	0.6332	0.9523	0	20
90	4.7199	6.5202	0.6331	0.9525	0	22
120	3.4005	3.5617	0.7331	0.8225	0	26

จากตารางที่ 5.6 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีไอโดยวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร พบว่า จำนวนประชากรเริ่มต้นเป็น 15 30 60 90 และ 120 ตัวมีค่าพุ่งเกินเท่ากัน คือ 0



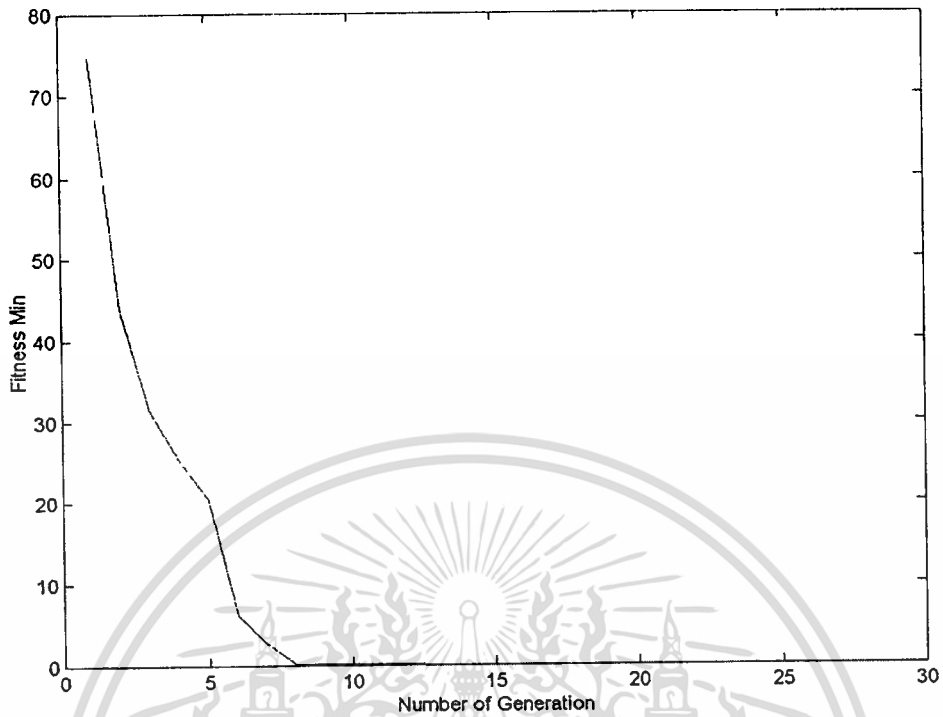
รูปที่ 5.36 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



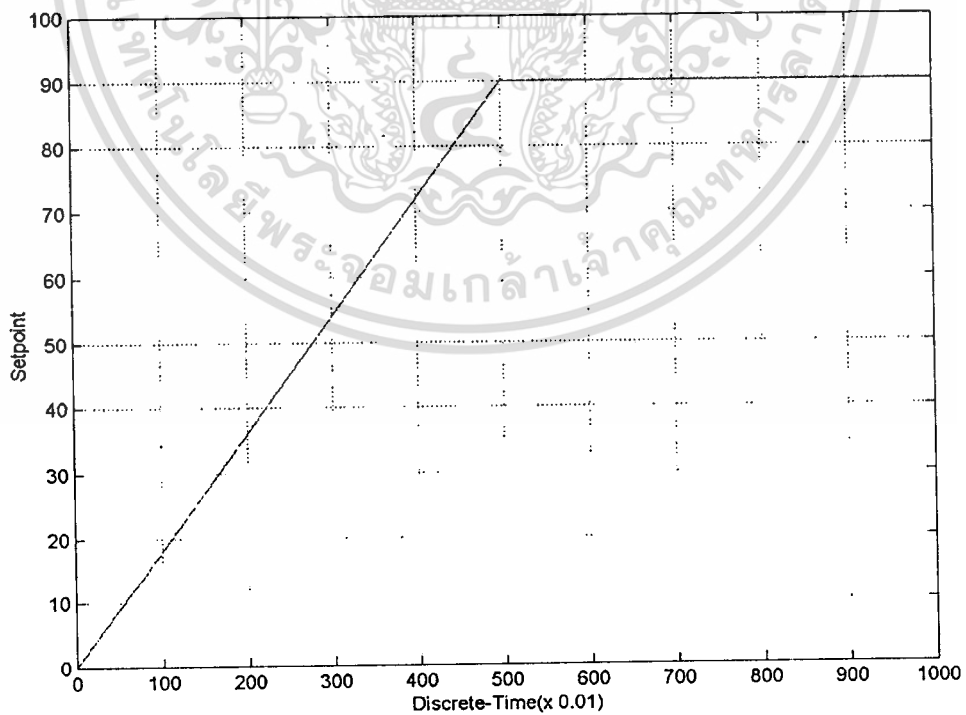
รูปที่ 5.37 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 15 ตัว ของแบบ

จำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



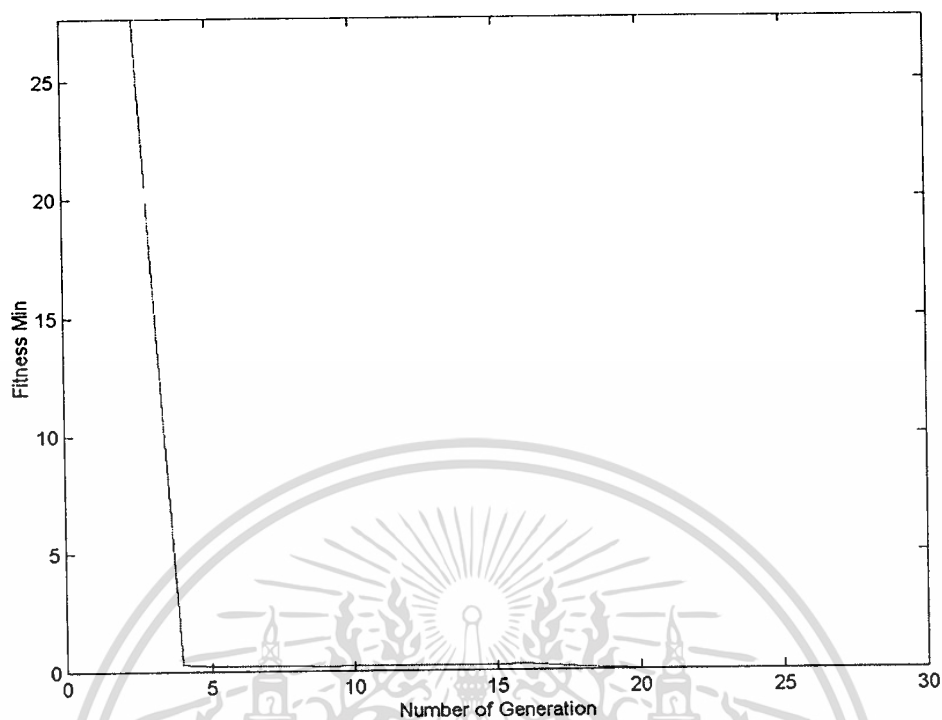
รูปที่ 5.38 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



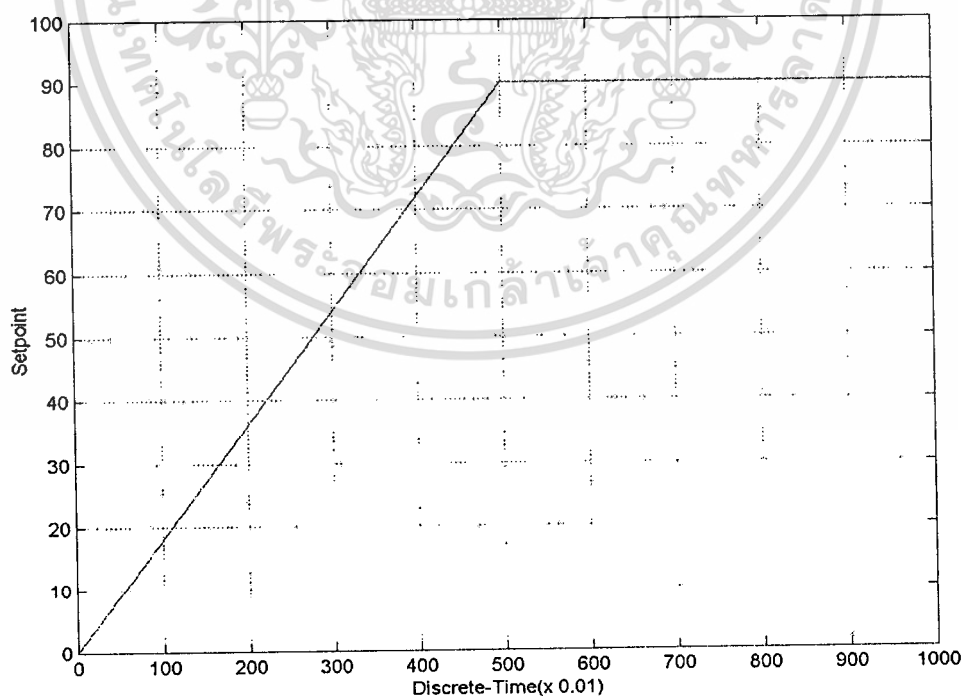
รูปที่ 5.39 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 30 ตัว ของแบบ

จำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

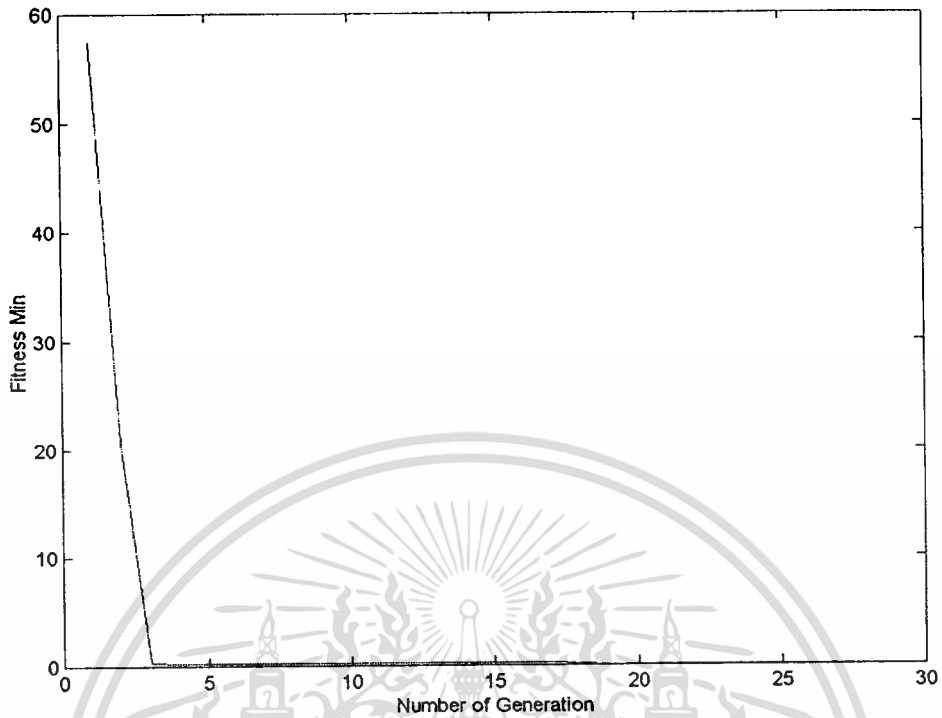


รูปที่ 5.40 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

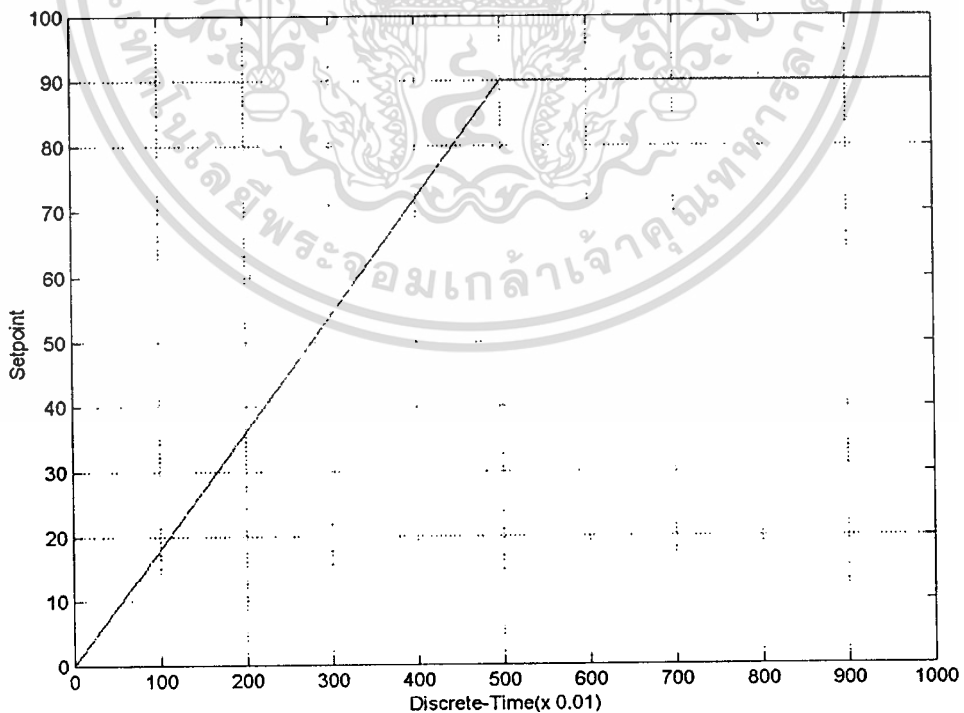


รูปที่ 5.41 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



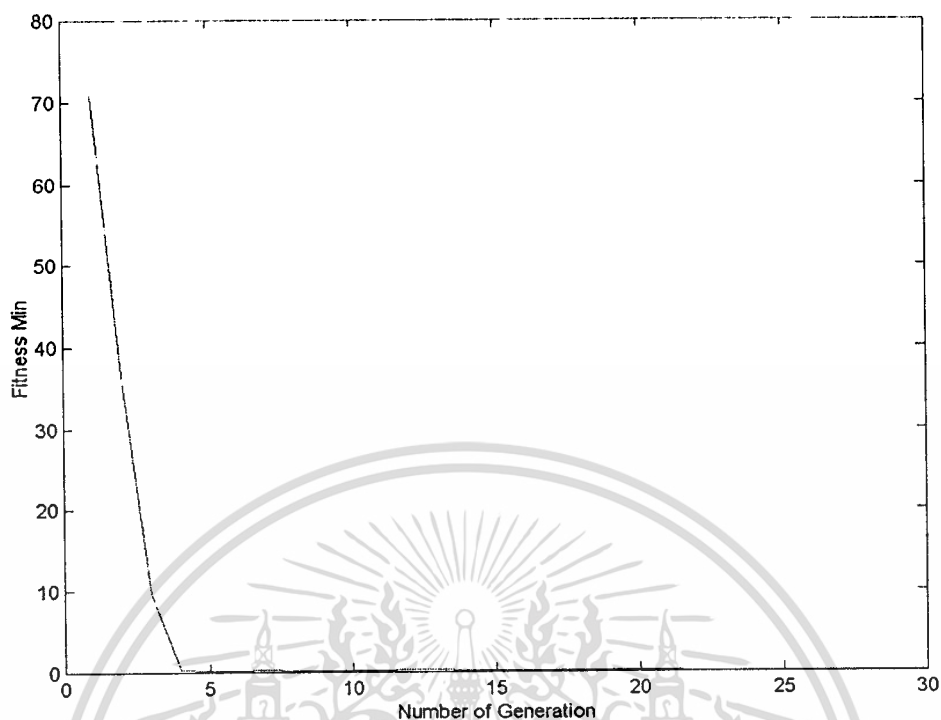
รูปที่ 5.42 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



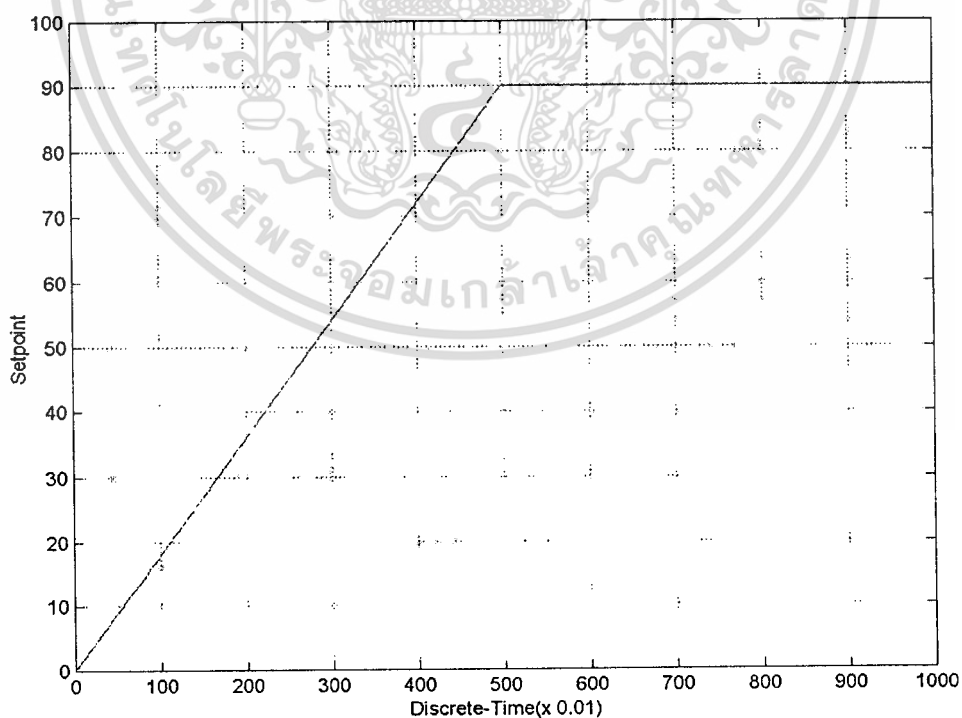
รูปที่ 5.43 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 90 ตัว ของแบบ

จำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.44 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลอง
หุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



รูปที่ 5.45 ผลตอบสนองของตัวควบคุมแบบพีซีพีไอโดยใช้ประชากรเริ่มต้น 120 ตัว ของแบบ

จำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

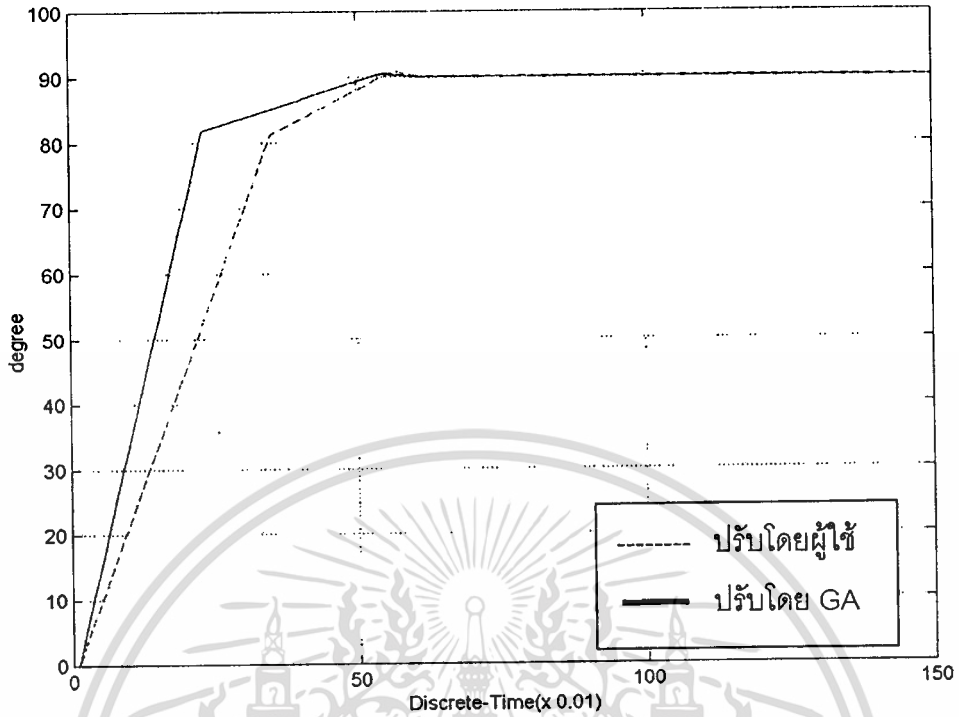
รูปที่ 5.36 5.38 5.40 5.42 และ 5.44 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีพีไอ ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัว จะใช้จำนวนรอบในการเกิด (generation) มากถึง 24 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 90 และ 120 ตัวจะพบว่าจำนวนรอบของการเกิดจะลดลง ซึ่งจำนวนรอบของการเกิดไม่ได้ขึ้นอยู่กับจำนวนของประชากรเริ่มต้น จะเห็นได้ว่าจากแนวโน้มการลู่เข้าที่จำนวนประชากรเริ่มต้นที่ 120 ตัว ไม่ได้ดีกว่าประชากรเริ่มต้นที่ 30 ตัว ดังนั้นจำนวนประชากรเริ่มต้นที่เหมาะสมควร จึงต้องคำนึงถึงเวลาในการประมวลผลด้วย ซึ่งจำนวนประชากรเริ่มต้นที่ 30 ตัว จะมีความเหมาะสมที่สุด

จากรูปที่ 5.37 5.39 5.41 5.43 และ 5.45 เป็นผลตอบสนองของตัวควบคุมแบบฟัซซีพีไอ ที่ได้จากการปรับแต่งโดยใช้วิธีทางพันธุศาสตร์ พบว่าที่จำนวนประชากรเริ่มต้น 15 30 60 90 และ 120 ตัวจะมีค่าพุ่งเกิน ที่เท่ากันคือ 0 ซึ่งเป็นค่าที่ดีที่สุด

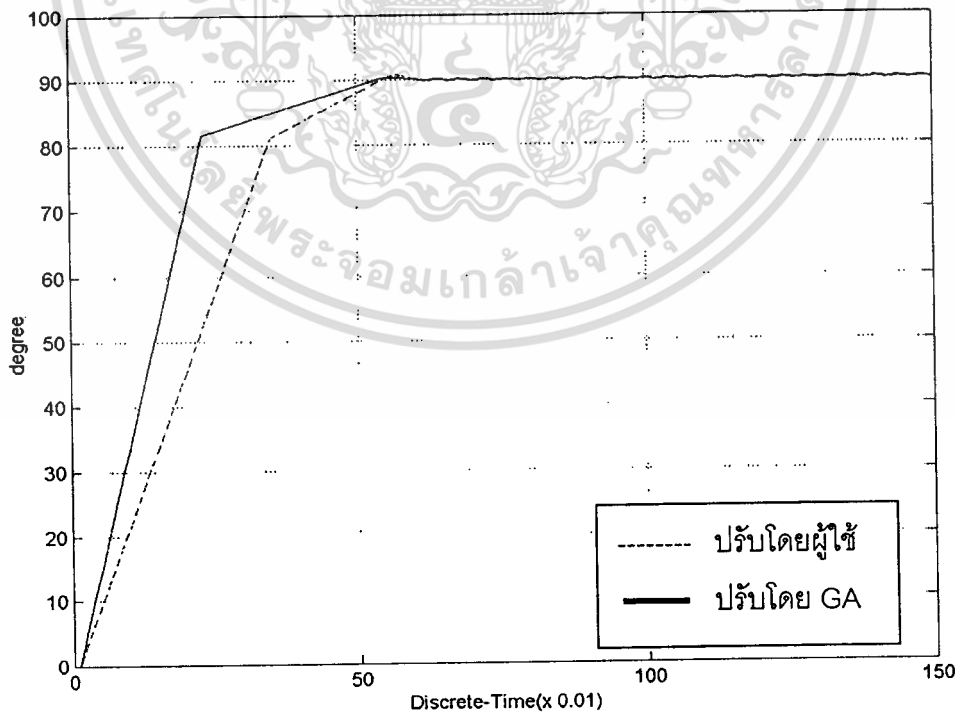
ตารางที่ 5.7 ค่าพารามิเตอร์ที่ปรับโดยผู้ใช้ กับค่าพารามิเตอร์ที่ปรับโดยวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร

ค่าเกณฑ์	Fuzzy PD				Fuzzy PI			
	Kp	Kd	Ku_pd	Ld	Kp	Ki	Ku_pi	Li
โดยผู้ใช้	2	2	4	4	3	3	1	1
GA-15	2.1496	2.3893	4.9970	4.9844	4.0846	3.4263	0.8776	0.7048
GA-30	2.4378	1.8337	4.9789	4.9963	1.6269	6.2038	0.9756	0.6181
GA-60	2.8845	1.1201	4.9775	4.9885	4.7183	5.8345	0.6332	0.9523
GA-90	1.5644	2.0621	4.9745	4.9774	4.7199	6.5202	0.6331	0.9525
GA-120	1.2527	1.9868	4.9951	4.9961	3.4005	3.5617	0.7331	0.8225

จากตารางที่ 5.7 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีพีสแควร์ไอดี เปรียบเทียบกับการปรับโดยผู้ใช้ ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร ด้วยวิธีทางพันธุศาสตร์ พบว่าการปรับแต่งด้วยวิธีทางพันธุศาสตร์จะมีค่าพารามิเตอร์ที่ละเอียดมากกว่า

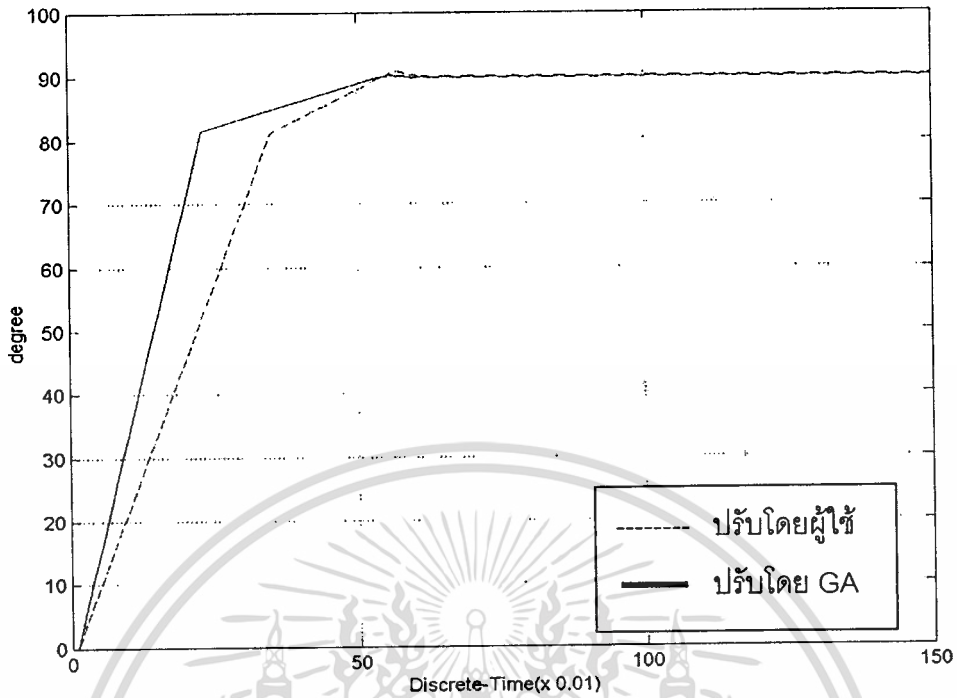


รูปที่ 5.46 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-15 ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

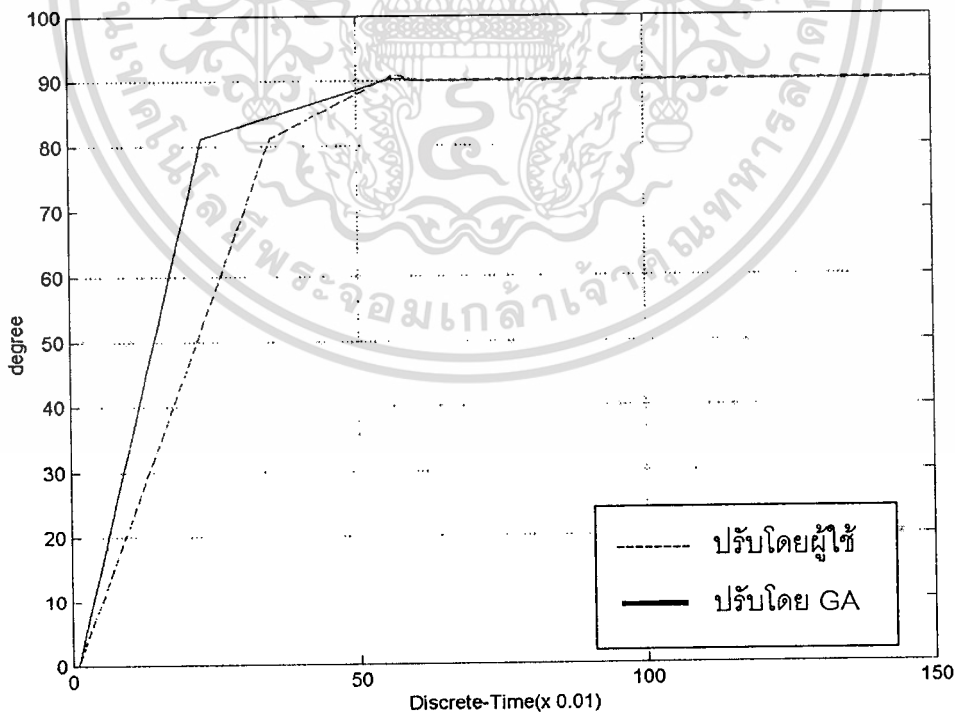


รูปที่ 5.47 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-30

เอกสารนี้เป็นเอกสารของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

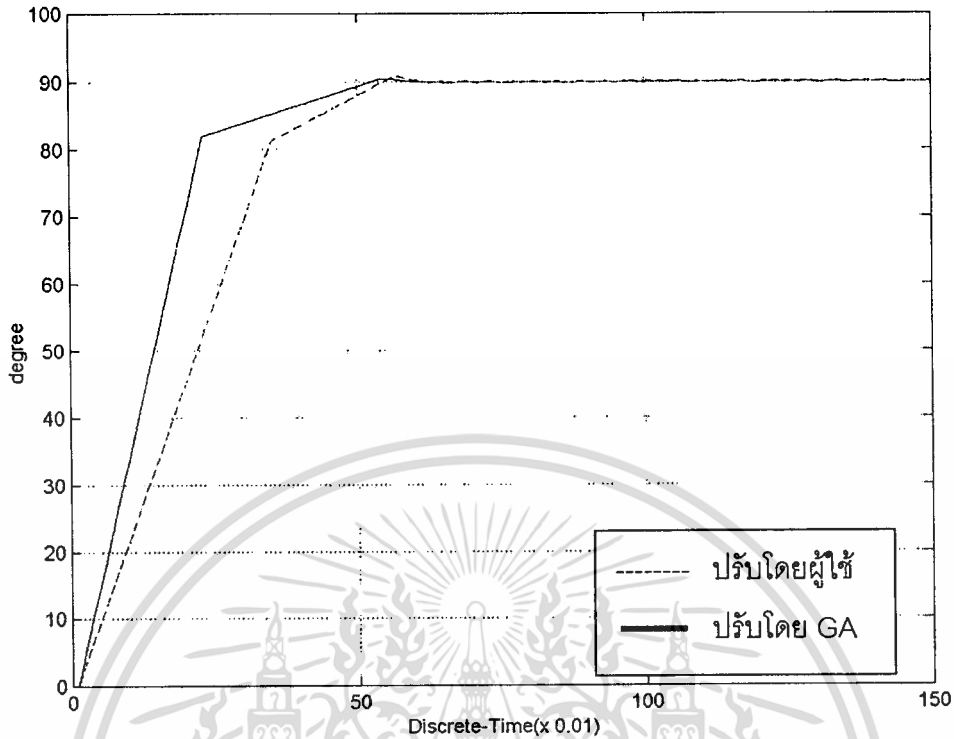


รูปที่ 5.48 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-60 ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร



รูปที่ 5.49 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้กับการปรับแต่งโดย GA-90 ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.50 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้อยู่กับการปรับแต่งโดย GA-120 ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

จากรูปที่ 5.46 5.47 5.48 5.49 และ 5.50 เป็นการเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้อยู่กับการปรับแต่งโดยวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปรพบว่า ผลตอบสนองของการแต่งโดยวิธีทางพันธุศาสตร์จะดีกว่าการปรับแต่งโดยผู้ใช้อยู่ และประสิทธิภาพที่ได้จะแสดงดังตารางที่ 5.8

ตารางที่ 5.8 ประสิทธิภาพของการปรับแต่งของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 8 ตัวแปร

ปรับโดย	Rise-time(sec)	Setting-time(sec)	Overshoot(%)
ผู้ใช้	0.3014	0.5065	0.008
GA-15	0.19	0.4616	0.0051
GA-30	0.19	0.4714	0.0045
GA-60	0.19	0.4770	0.0028
GA-90	0.19	0.4855	0.0033
GA-120	0.19	0.4616	0.0044

จากตารางที่ 5.8 พบว่าการปรับแต่งโดยวิธีทางพันธุศาสตร์ ของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 8 ตัวแปร จะมีค่า Rise-Time , Setting-Time และ Overshoot ที่น้อยกว่าการปรับโดยผู้ใช้ แสดงว่าการใช้วิธีทางพันธุศาสตร์จะดีกว่า

5.2 การจำลองหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีแบบ 10 ตัวแปร

การจำลองจะแบ่งเป็นสองการจำลอง โดยการจำลองแรกจะเป็นการจำลองกับแบบจำลองที่ไม่เป็นเชิงเส้น ส่วนการจำลองที่ 2 จะเป็นการจำลองกับแบบจำลองหุ่นยนต์เคลื่อนที่

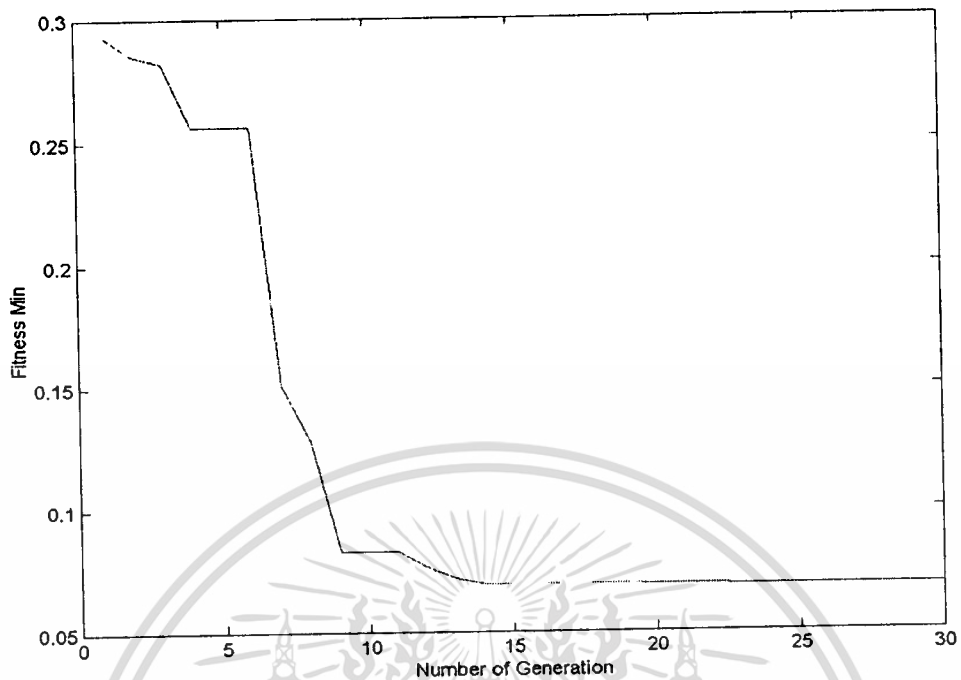
5.2.1 แบบจำลองที่ไม่เป็นเชิงเส้น (Nonlinear Plant)

การจำลองนี้จะเป็นการทดสอบหาค่าแนวโน้มของการลู่เข้าของโปรแกรม โดยใช้จำนวนของประชากรเริ่มต้นที่แตกต่างกัน และอีกส่วนจะเป็นการเปรียบเทียบกันระหว่างการปรับแต่งโดยวิธีหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีแบบ 8 ตัวแปรกับวิธีหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีแบบ 10 ตัวแปร สมการที่ใช้ดังสมการที่ 5.1 และกำหนดค่า $\text{set-point} = 5$ โดยผลที่ได้จะแสดงดังตารางที่ 5.9

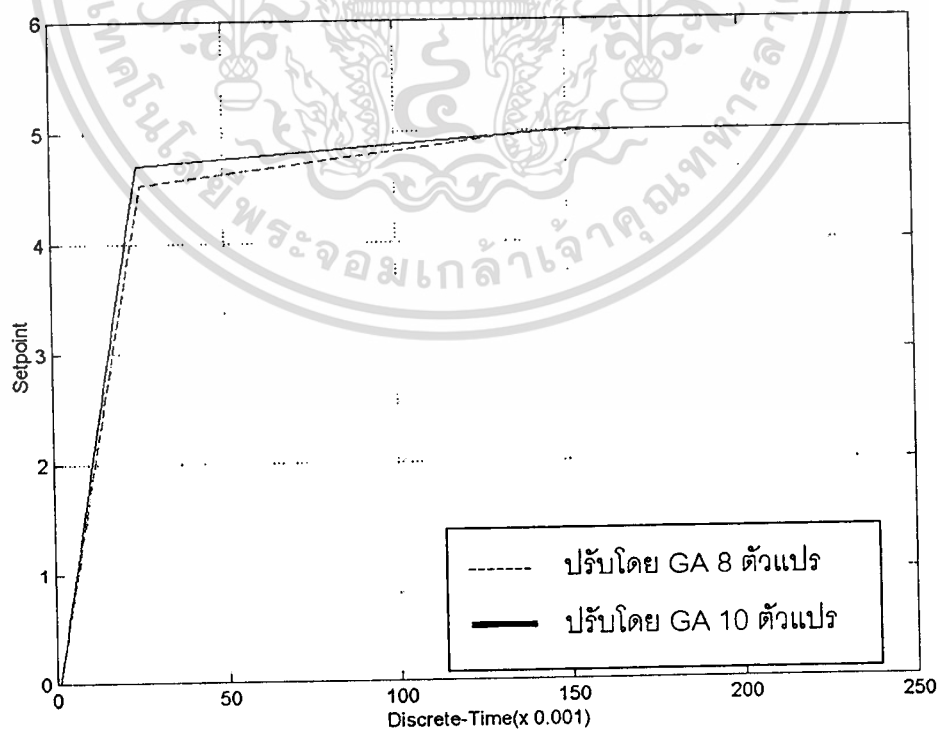
ตารางที่ 5.9 ค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 10 ตัวแปร

จำนวนประชากรเริ่มต้น	K_{p_d}	K_d	L_d	$K_{u_{pd}}$	K_{p_i}	K_i	L_i	$K_{u_{pi}}$	sel	T_s
15	1.7769	4.4946	19.9896	19.5932	1.3704	0.1119	1.2724	1.2610	4.6149	0.001
30	2.6801	1.3691	20	19.9028	1.8735	0.9193	0.8670	1.9135	4.9108	0.001
60	5	0.6217	20	19.9986	1.4128	0.2151	2.8412	0.9907	4.8502	0.001
90	5	0.5191	20	19.9986	1.3858	0.2482	2.7984	0.9838	4.8474	0.001
120	5	0.7406	19.9998	19.9989	1.6253	0.1793	2.5814	0.7957	4.8488	0.001

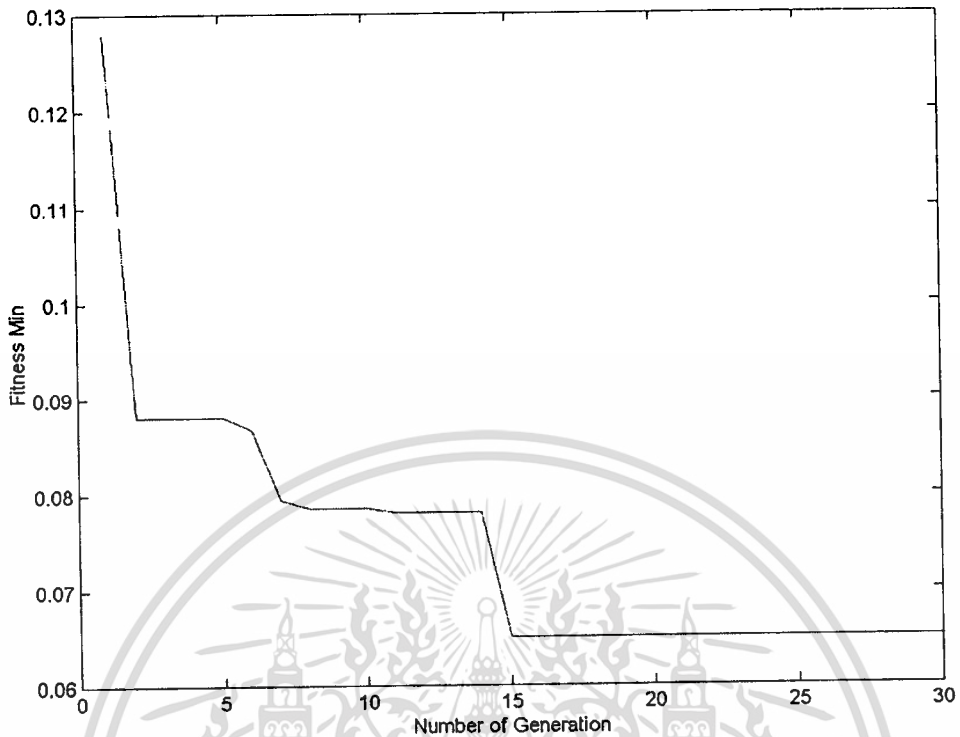
จากตารางที่ 5.9 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีเอสแควร์ไอดีแบบ 10 ตัวแปร โดยวิธีทางพันธุศาสตร์ของแบบจำลองที่ไม่เป็นเชิงเส้น



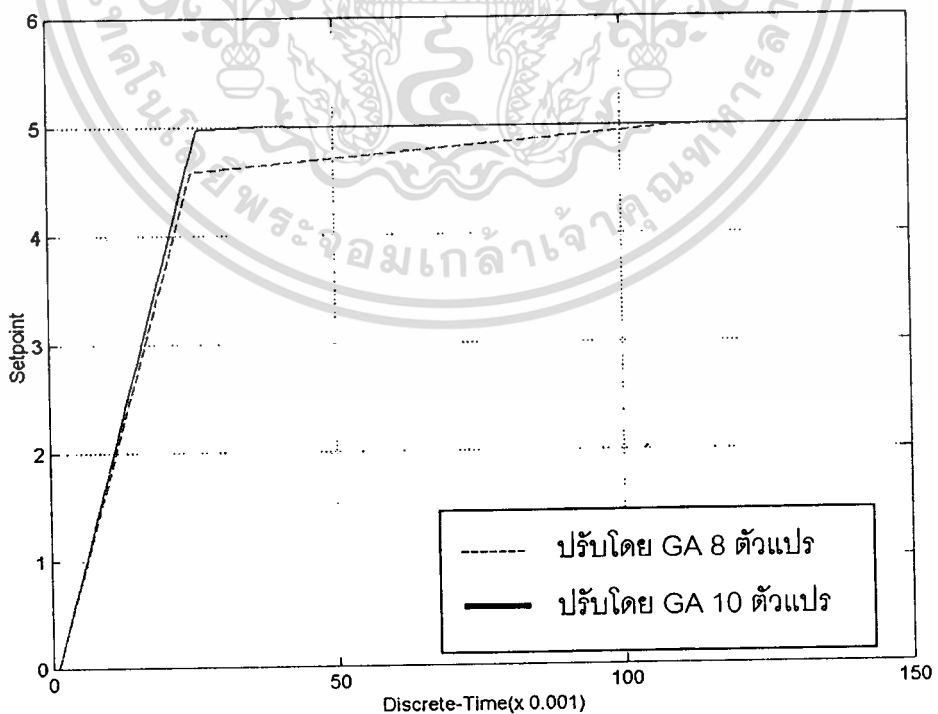
รูปที่ 5.51 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น แบบ 10-ตัวแปร



รูปที่ 5.52 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่งเอกสารที่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยเท่านั้น ไม่อนุญาตให้ใช้ในการปฏิบัติงานด้านการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 15 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



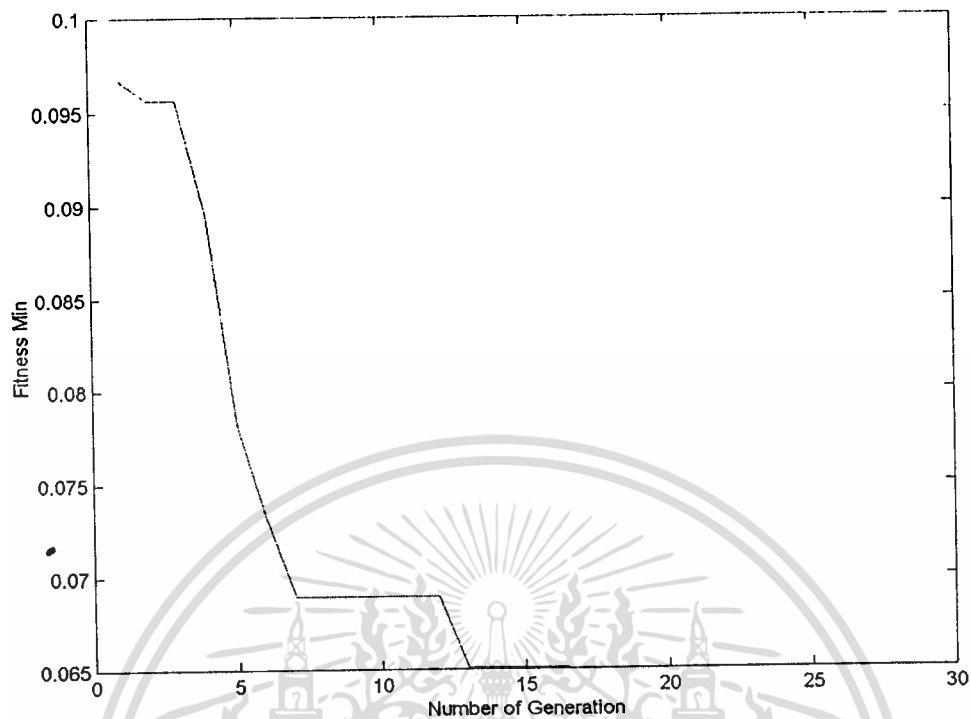
รูปที่ 5.53 แนวโน้มของการรู้เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร



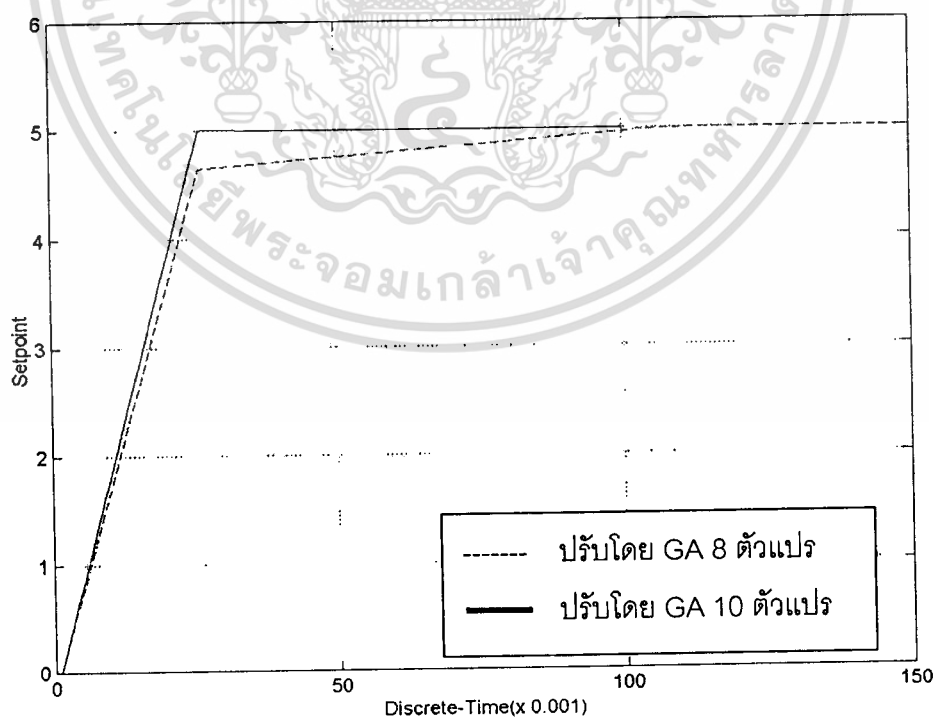
รูปที่ 5.54 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์เท่านั้น ไม่ควรแจกจ่ายหรือใช้เพื่อวัตถุประสงค์ทางการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 30 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

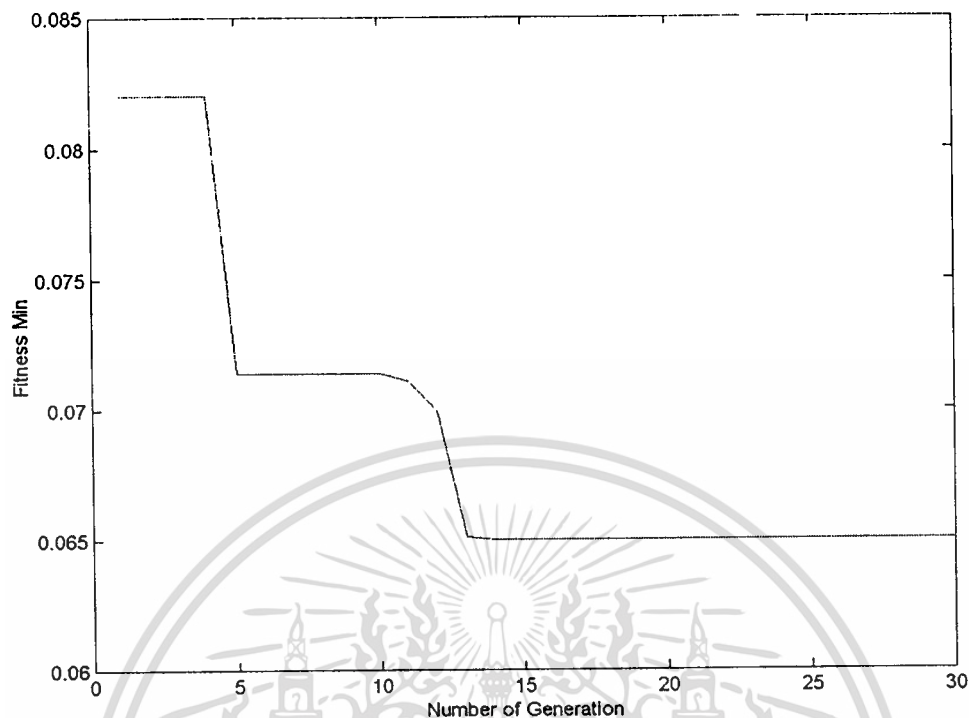


รูปที่ 5.55 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร

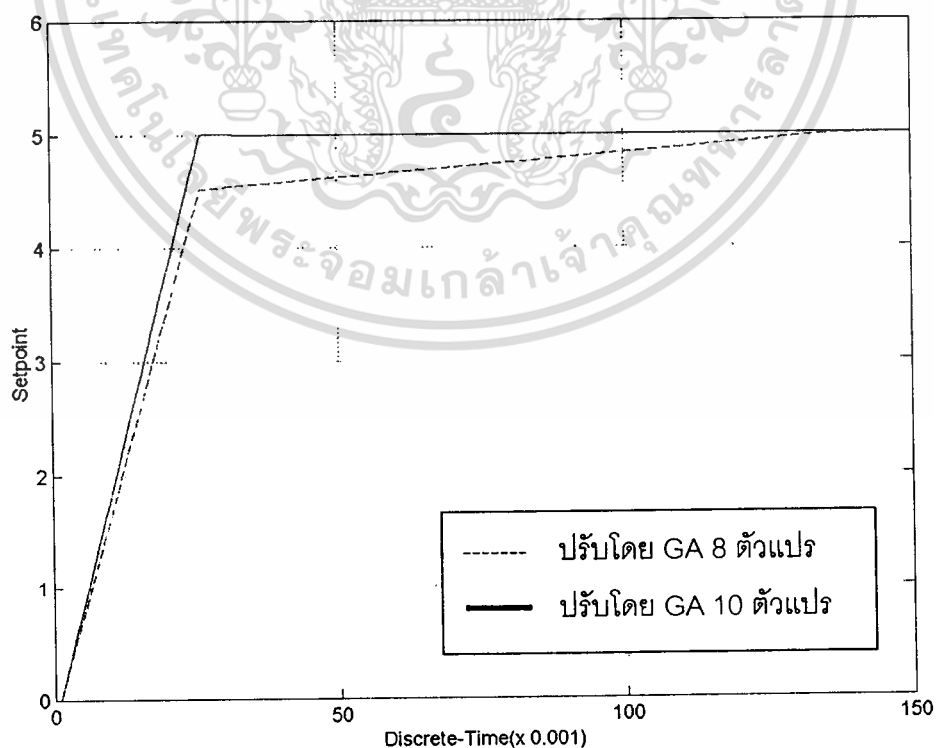


รูปที่ 5.56 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่ง

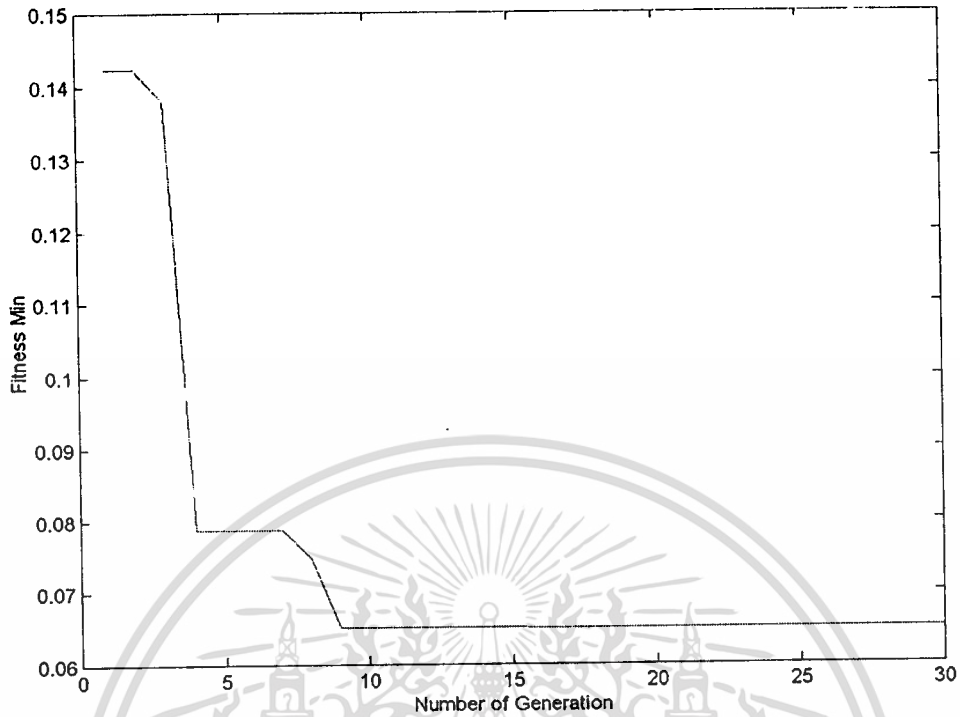
เอกสารนี้เป็นเอกสาร โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 60 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



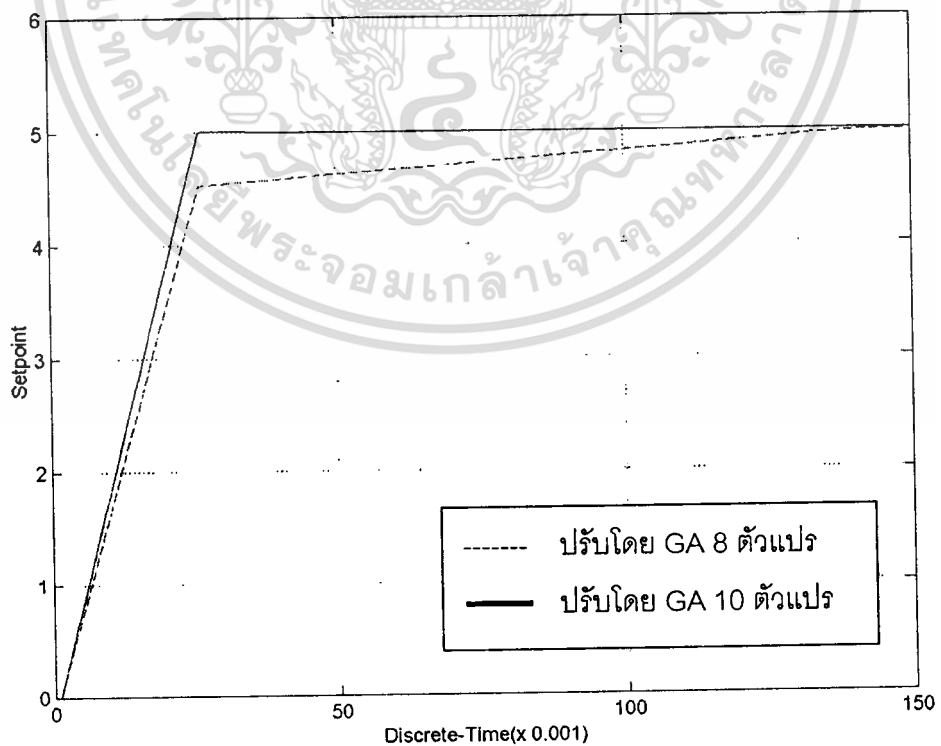
รูปที่ 5.57 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร



รูปที่ 5.58 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขไปใช้ประโยชน์ด้านการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 90 ตัว ของ แบบจำลองที่ไม่เป็นเชิงเส้น ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.59 แนวโน้มของการลูเข้าค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร



รูปที่ 5.60 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่งเอกสารที่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 120 ตัว ของแบบจำลองที่ไม่เป็นเชิงเส้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.51 5.53 5.55 5.57 และ 5.59 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีสแควร์ไอดี แบบ 10 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัวจะใช้จำนวนรอบในการเกิด (generation) มากถึง 23 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 90 และ 120 ตัวจะพบว่าจำนวนรอบของการเกิดจะลดลง และมีค่าที่ใกล้เคียงกัน ดังนั้นจำนวนประชากรเริ่มต้นที่เหมาะสมควรจะอยู่ที่ 30 ตัว เนื่องจากจะลดเวลาในการประมวลผล

จากรูปที่ 5.52 5.54 5.56 5.58 และ 5.60 เป็นการเปรียบเทียบผลตอบสนองของการปรับแต่งโดยผู้ใช้งานกับการปรับแต่งโดยวิธีทางพันธุศาสตร์ ของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร พบว่าผลตอบสนองของการแต่งโดยวิธีทางพันธุศาสตร์ จะดีกว่าการปรับแต่งโดยผู้ใช้งาน และประสิทธิภาพที่ได้จะแสดงดังตารางที่ 5.10

ตารางที่ 5.10 ประสิทธิภาพของการปรับแต่งของแบบจำลองที่ไม่เป็นเชิงเส้นแบบ 10 ตัวแปร
เปรียบเทียบกับ 8 ตัวแปร

ประชากรเริ่มต้น	Rise-time(sec)	Setting-time(sec)	Overshoot(%)	Max gen
GA8 30	0.02	0.0568	0	5
GA10 15	0.02	0.108	0	23
GA10 30	0.02	0.0256	0	15
GA10 60	0.02	0.0255	0	13
GA10 90	0.02	0.0255	0	14
GA10 120	0.02	0.0255	0	15

จากตารางที่ 5.10 พบว่าการปรับแต่งโดย GA 10 ตัวแปรของแบบจำลองที่ไม่เป็นเชิงเส้นจะมีค่า Setting-Time และ Overshoot ที่น้อยกว่าการปรับโดย GA 8 ตัวแปร ส่วนค่าของ Rise-Time มีค่าเท่ากัน แสดงว่าการใช้ GA 10 ตัวแปรดีกว่า

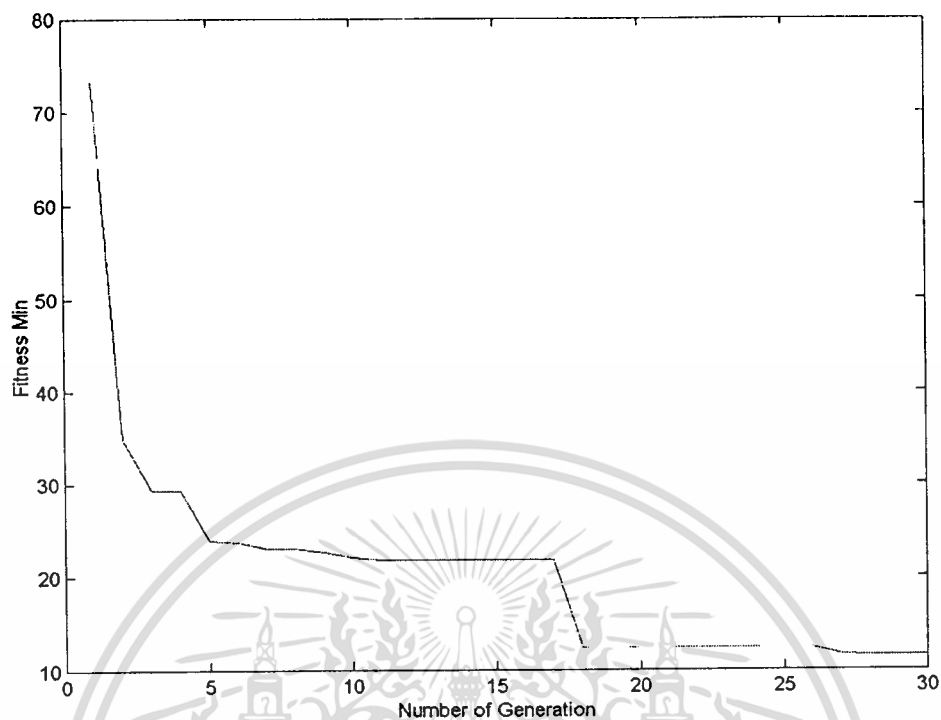
5.2.2 แบบจำลองหุ่นยนต์เคลื่อนที่ (Mobile Robot Model)

การจำลองนี้จะเป็นการทดสอบหาค่าแนวโน้มของการลู่เข้าของโปรแกรม โดยใช้ประชากรเริ่มต้นที่แตกต่างกัน และอีกส่วนหนึ่งจะเป็นการเปรียบเทียบกันระหว่าง การปรับแต่งโดยวิธีหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟีดแบ็คไอดีแบบ 8 ตัวแปรกับวิธีหาค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟีดแบ็คไอดีแบบ 10 ตัวแปร สมการที่ใช้ดังสมการที่ 5.2 และกำหนดค่า set-point = 90 องศา $D = 0.067$ เมตร โดยผลที่ได้จะแสดงดังตารางที่ 5.11

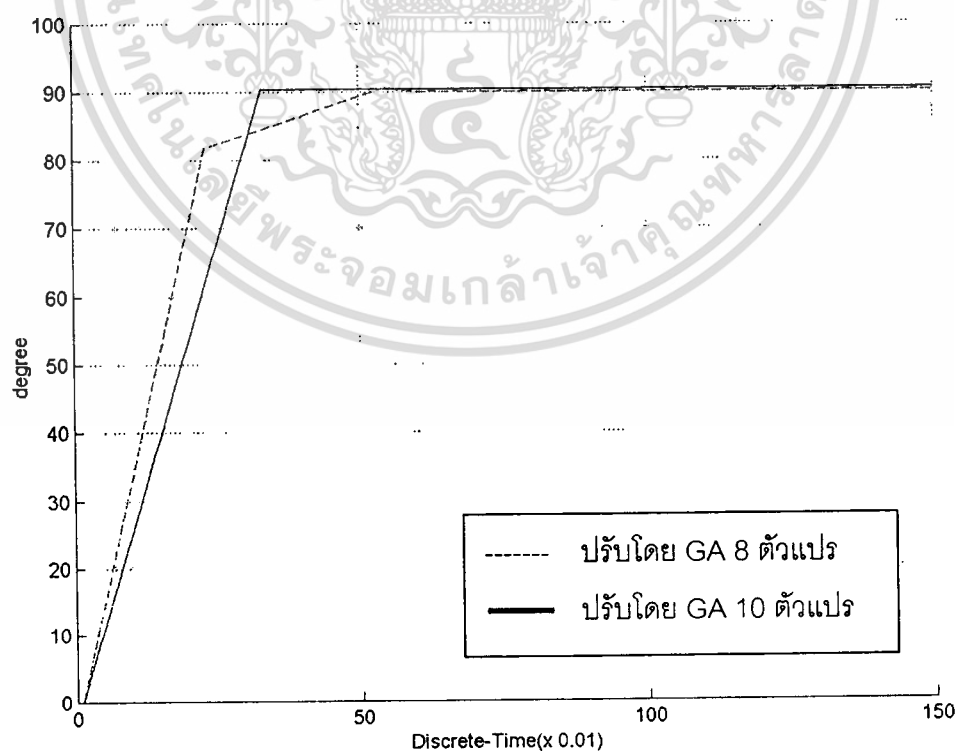
ตารางที่ 5.11 ค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟีดแบ็คไอดีที่ได้จากวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร

จำนวนประชากรเริ่มต้น	Kp_d	Kd	Ld	Ku_{pd}	Kp_i	Ki	Li	Ku_{pi}	sel	Ts
15	0.7391	1.4530	5	3.7861	3.9784	9.9638	0.3708	0.0001	88.2132	0.01
30	2.9949	0.3328	4.8237	5	1.1677	9.9570	0.0002	0.1758	89.5246	0.01
60	0.6753	0.6067	4.8305	4.9940	4.0550	9.9382	0.9967	0.0001	87.8392	0.01
90	3	0.3288	4.8237	5	1.1679	9.9571	0.0003	0.0557	89.5247	0.01
120	2.9620	0.3685	4.8237	5	1.1677	9.9564	0.0001	0.4538	89.5247	0.01

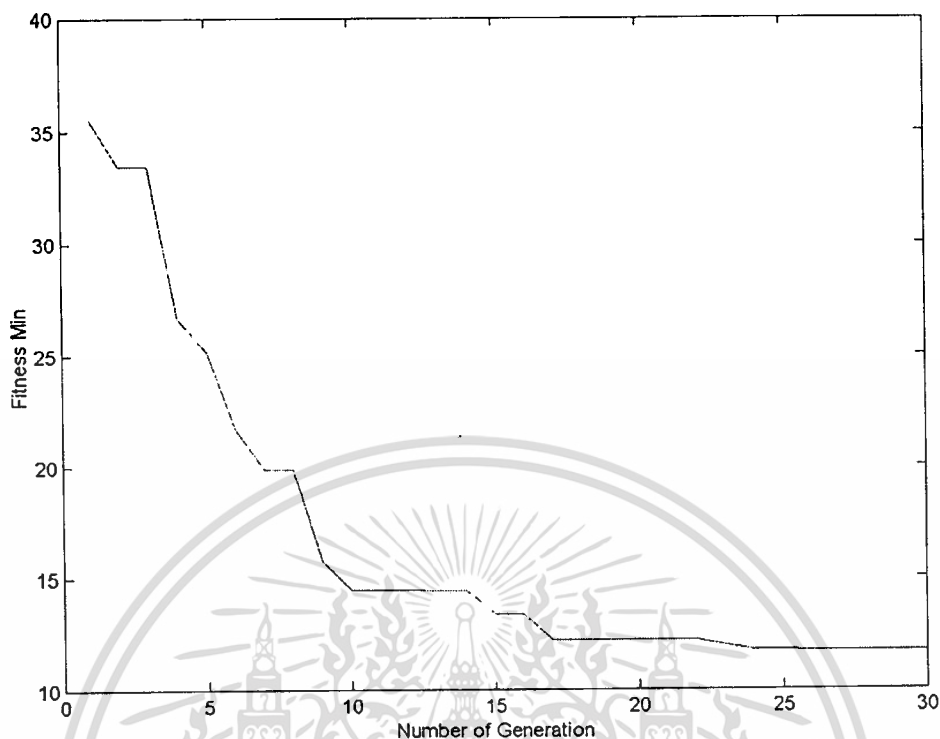
จากตารางที่ 5.11 เป็นค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟีดแบ็คไอดี โดยวิธีทางพันธุศาสตร์ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



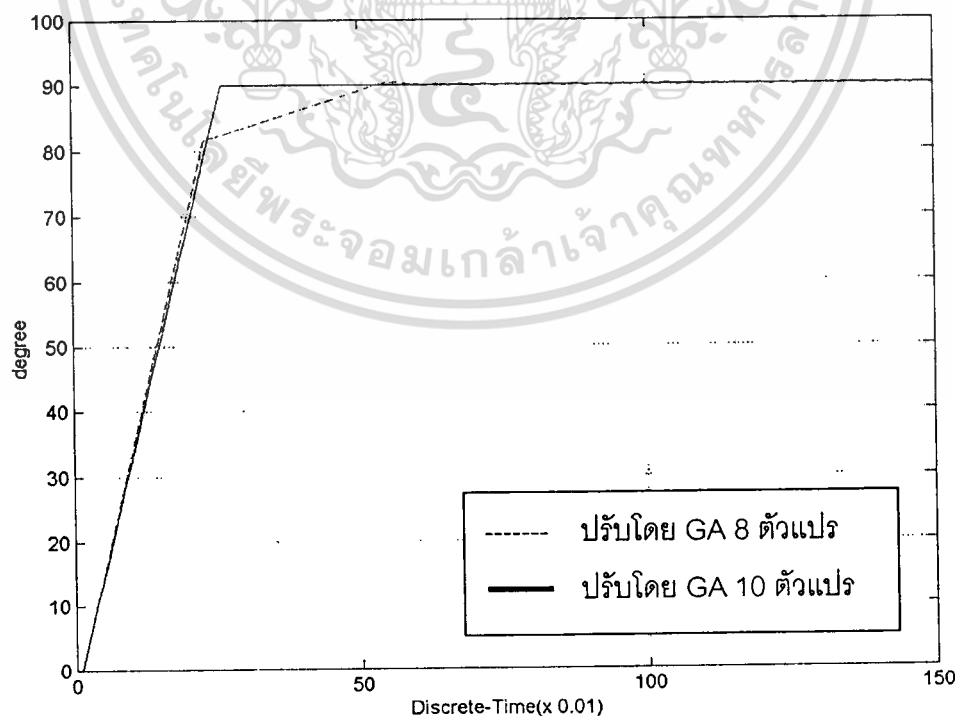
รูปที่ 5.61 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 15 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



รูปที่ 5.62 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 15 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



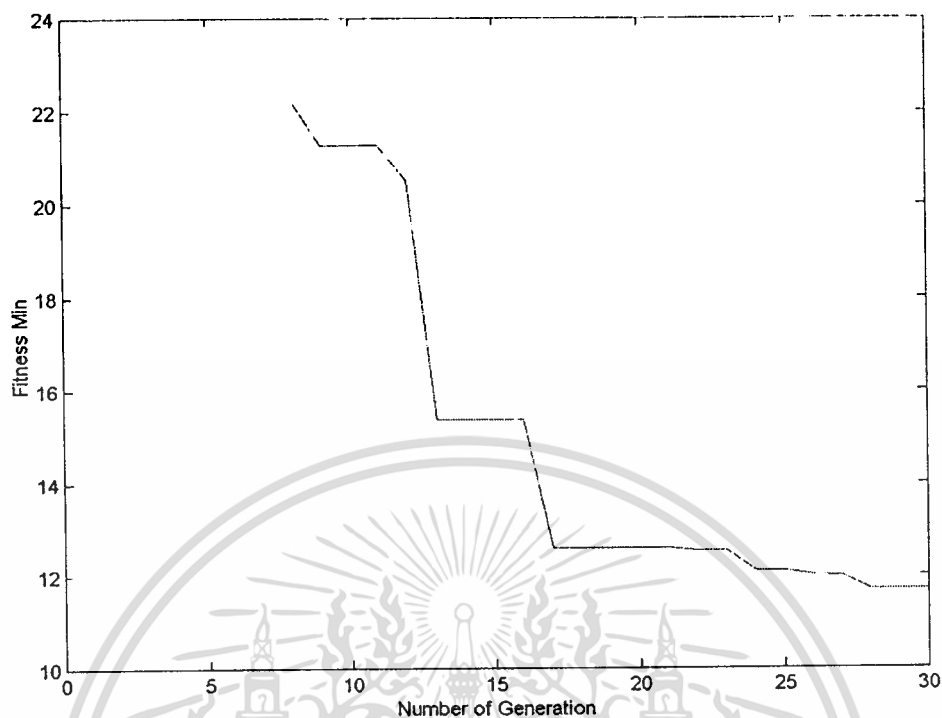
รูปที่ 5.63 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 30 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



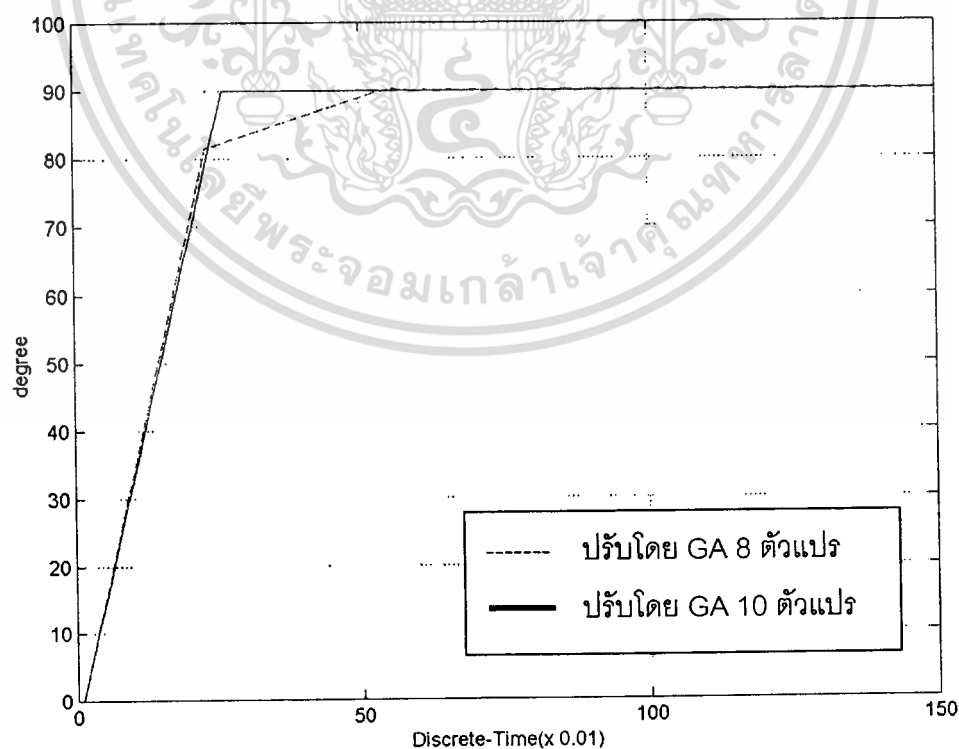
รูปที่ 5.64 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานใช้ภายในเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 30 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่

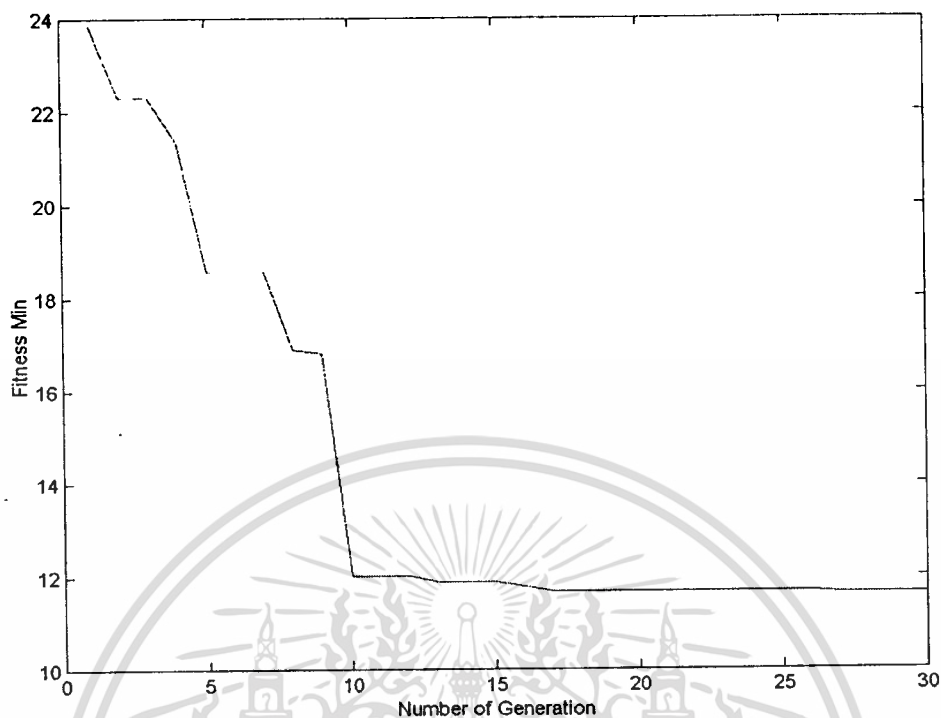
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



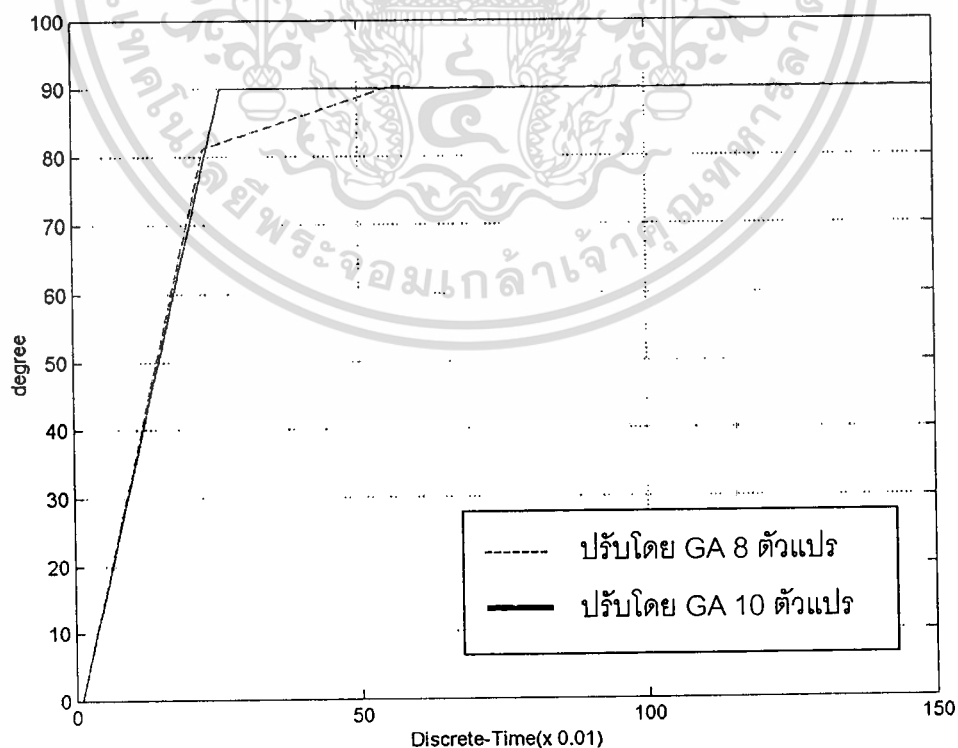
รูปที่ 5.65 แนวโน้มของการลู่อู่ค่าเป้าหมายโดยประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



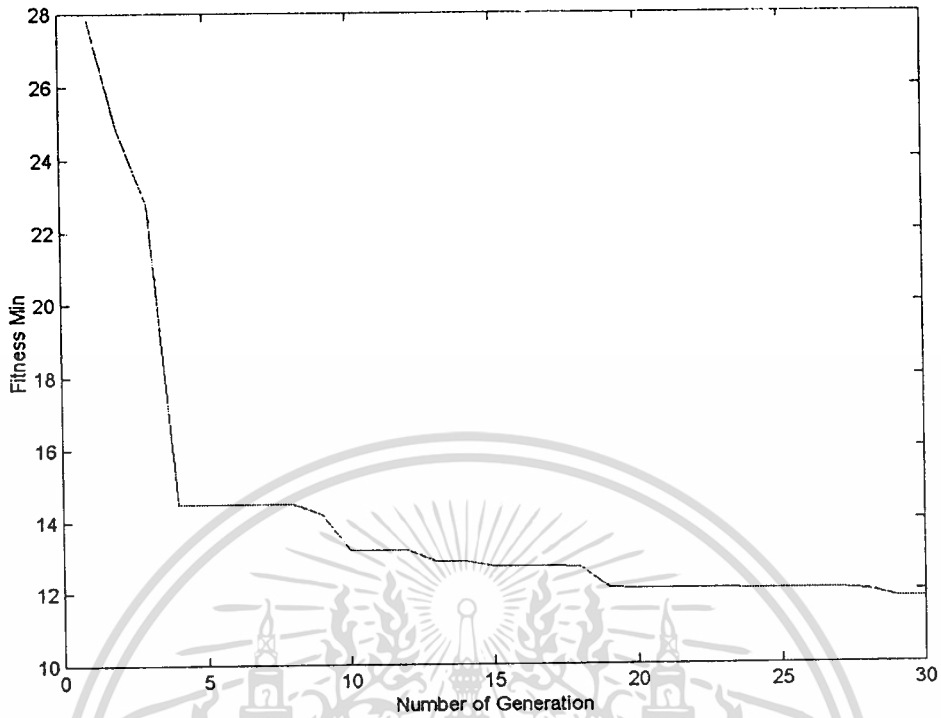
รูปที่ 5.66 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 60 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



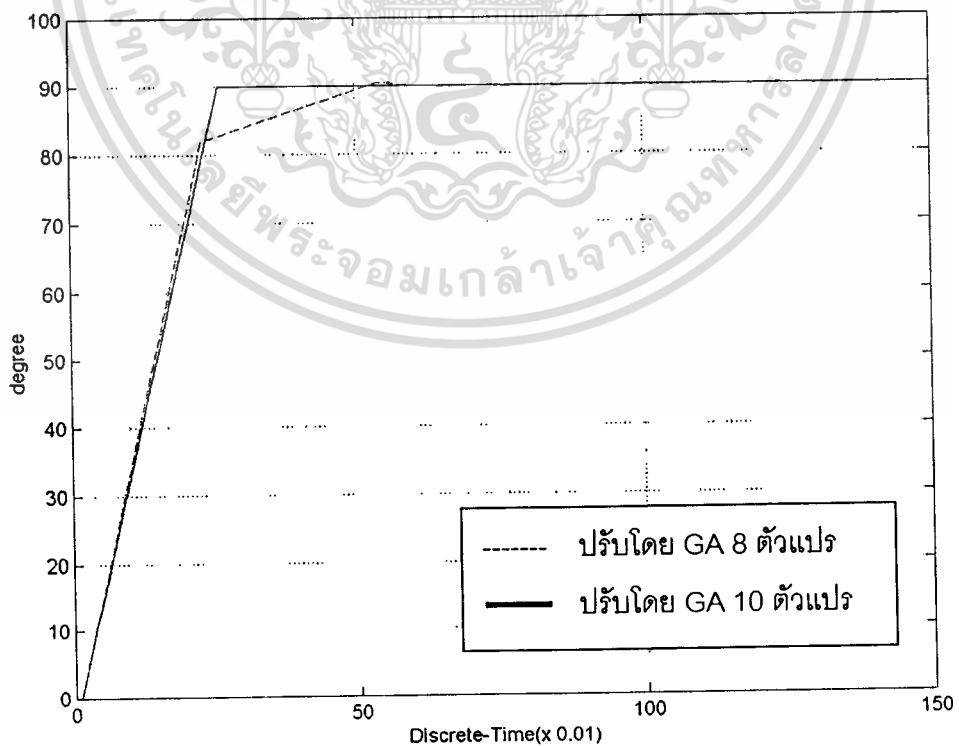
รูปที่ 5.67 แนวโน้มของการลู่เข้าค่าเป้าหมายโดยประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



รูปที่ 5.68 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 90 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ ไม่ว่ากรณีใดๆ ทั้งสิ้น ออกกฎหมายให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.69 แนวโน้มของการหาค่าเป้าหมายโดยประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ แบบ 10 ตัวแปร



รูปที่ 5.70 การเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปรกับการปรับแต่งเอกสารเป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า โดย GA 10 ตัวแปร ที่ประชากรเริ่มต้น 120 ตัว ของแบบจำลองหุ่นยนต์เคลื่อนที่ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.61 5.63 5.65 5.67 และ 5.69 เป็นแนวโน้มการลู่เข้าเป้าหมายของการใช้วิธีทางพันธุศาสตร์ ในการหาค่าพารามิเตอร์ของตัวควบคุมแบบพีซีพีสแควร์ไอดีของแบบจำลองหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร จะเห็นได้ว่าที่จำนวนประชากรเริ่มต้นเท่ากับ 15 ตัวจะใช้จำนวนรอบในการเกิด (generation) มากถึง 27 รอบ เมื่อเพิ่มจำนวนประชากรเป็น 30 60 และ 90 ตัว จะพบว่าจำนวนรอบของการเกิดจะลดลงไม่มาก และมีค่าที่ใกล้เคียงกัน จะเห็นว่าจากแนวโน้มการลู่เข้าที่จำนวนประชากรเริ่มต้นที่ 120 ตัว ไม่ได้ดีกว่าประชากรเริ่มต้นที่ 30 ตัว ดังนั้นจำนวนประชากรเริ่มต้นที่เหมาะสมควรจึงต้องคำนึงถึงเวลาในการประมวลผลด้วย ซึ่งจำนวนประชากรเริ่มต้นที่ 30 ตัวจึงมีความเหมาะสมที่สุด

จากรูปที่ 5.62 5.64 5.66 5.68 และ 5.70 เป็นการเปรียบเทียบผลตอบสนองของการปรับแต่งโดย GA 8 ตัวแปร กับการปรับแต่ง โดย GA 10 ตัวแปร ของแบบจำลองหุ่นยนต์เคลื่อนที่พบว่า ผลตอบสนองของการแต่งโดย GA 10 ตัวแปร จะดีกว่าการปรับแต่งโดย GA 8 ตัวแปร และประสิทธิภาพที่ได้จะแสดงดังตารางที่ 5.12

ตารางที่ 5.12 ประสิทธิภาพของการปรับแต่งของหุ่นยนต์เคลื่อนที่แบบ 10 ตัวแปร
เปรียบเทียบกับ 8 ตัวแปร

ประชากรเริ่มต้น	Rise-time(sec)	Setting-time(sec)(2%)	Overshoot(%)	Max gen
GA8 120	0.19	0.4646	0.0044	26
GA10 15	0.2549	0.3221	0	27
GA10 30	0.2	0.255	0	24
GA10 60	0.2	0.255	0	27
GA10 90	0.2	0.255	0	26
GA10 120	0.2	0.255	0	29

จากตารางที่ 5.12 พบว่า การปรับแต่งโดย GA 10 ตัวแปร ของแบบจำลองหุ่นยนต์เคลื่อนที่ จะมีค่า Setting-time และ Overshoot ที่น้อยกว่าการปรับโดย GA 8 ตัวแปร ส่วนค่าของ Rise-time มีความแตกต่างกันเล็กน้อย แสดงว่าการใช้ GA 10 ตัวแปรดีกว่า

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบฟัซซีฟัสแควรีไอดีนั้น เป็นการจำลองโดยใช้วิธีทางพันธุศาสตร์ในการหาค่าพารามิเตอร์ที่เหมาะสม โดยทดสอบใช้กับแบบจำลองที่ไม่เป็นเชิงเส้น และแบบจำลองของหุ่นยนต์เคลื่อนที่ ซึ่งใช้โปรแกรม Matlab ในการทดสอบ และเป็นการจำลองการทำงาน โดยเริ่มแรกจะเป็นการออกแบบสร้างตัวควบคุมแบบฟัซซีฟัสแควรีไอดีเมื่อทดสอบจนได้ค่าพารามิเตอร์ที่ปรับโดยผู้ให้แล้ว ต่อจากนั้นจึงออกแบบส่วนของวิธีทางพันธุศาสตร์ (GA) แล้วทดสอบโปรแกรม หลังจากนั้นจึงนำโปรแกรม GA มาช่วยปรับแต่งค่าพารามิเตอร์ ซึ่งการจำลองจะแบ่งออกเป็น 2 แบบ คือ ปรับแต่งค่าพารามิเตอร์แบบ 8 ตัวแปร และ 10 ตัวแปร เพื่อหาค่าพารามิเตอร์ที่ทำให้ผลตอบสนองทางเอาต์พุตดีที่สุด

6.1 สรุปผลการจำลอง

จากการจำลองปรับแต่งค่าพารามิเตอร์แบบ 8 ตัวแปร โดยใช้กับแบบจำลองที่ไม่เป็นเชิงเส้น และแบบจำลองของหุ่นยนต์เคลื่อนที่ พบว่าจำนวนรอบของการเกิด (max generation) มีแนวโน้มของการลู่เข้าที่แตกต่างกันเพียงเล็กน้อยระหว่างจำนวนประชากรเริ่มต้นที่ 30 60 90 และ 120 ตัว เนื่องจากมีจำนวนประชากรมากพอในการนำเข้าไปในกระบวนการทางพันธุศาสตร์ แต่ประชากรเริ่มต้นที่ 15 ตัวนั้นมีจำนวนประชากรที่น้อยเกินไป จึงทำให้แนวโน้มของการลู่เข้าช้ากว่าจำนวนประชากรเริ่มต้นที่ 30 60 90 และ 120 ตัว เมื่อประชากรเริ่มต้นที่กล่าวมานั้นมีแนวโน้มของการลู่เข้าที่ใกล้เคียงกันจึงไม่มีความจำเป็นที่จะเลือกใช้ประชากรเริ่มต้นที่มาก เพราะจะทำให้เวลาการทำงานในกระบวนการของวิธีทางพันธุศาสตร์นั้นมากขึ้น และบางครั้งจำนวนประชากรเริ่มต้นที่มากเกินไปก็ไม่สามารถลู่เข้าเป้าหมายได้เร็วกว่าจำนวนประชากรเริ่มต้นที่น้อยกว่า ดังนั้นการกำหนดจำนวนของประชากรเริ่มต้นจึงต้องคำนึงถึงความเหมาะสมเป็นหลัก คือ ไม่มากและน้อยเกินไป ส่วนผลตอบสนองทางเอาต์พุตที่ได้จะดีกว่าการปรับแต่งโดยผู้ให้

ส่วนการจำลองปรับแต่งค่าพารามิเตอร์แบบ 10 ตัวแปร โดยเพิ่มการปรับแต่งจุดสลับ (Switching Point) และเวลาสุ่ม (Sampling Time) อีก 2 ตัวแปร ซึ่งใช้กับแบบจำลองที่ไม่เป็นเชิงเส้น จากผลการจำลองเมื่อเปรียบเทียบกันระหว่างการปรับแต่งค่าพารามิเตอร์แบบ 8 ตัวแปรกับแบบ 10 ตัวแปร จะเห็นว่าผลตอบสนองทางเอาต์พุตของค่าพารามิเตอร์แบบ 10 ตัวแปรดีกว่าแบบ 8 ตัวแปร ส่วนแนวโน้มของการลู่เข้าที่ประชากรเริ่ม 30 60 และ 90 ตัว แตกต่างกันเล็กน้อย แต่

ประชากรเริ่มต้นน้อยเกินไป และการลู่เข้าที่ประชากรเริ่มต้น 120 ตัว ใช้จำนวนรอบของการเกิดที่น้อยแต่ใช้เวลาในการประมวลผลที่นานมาก ดังนั้นประชากรเริ่มต้นที่ 30 ตัว จึงเป็นจำนวนที่เหมาะสมที่สุด สำหรับแบบจำลองแบบไม่เป็นเชิงเส้นที่ใช้ทดสอบ ที่จะใช้ในกระบวนการทางพันธุศาสตร์

ส่วนการจำลองปรับแต่งค่าพารามิเตอร์แบบ 10 ตัวแปรของแบบจำลองหุ่นยนต์เคลื่อนที่นั้น พบว่า มีค่าของเวลาไต่ขึ้น (Rise Time) ที่มากกว่าการปรับแต่งค่าพารามิเตอร์แบบ 8 ตัวแปรเล็กน้อย แต่ค่าเวลาคงตัว (Setting Time) มีค่าที่ดีกว่าเนื่องจากได้มีการปรับจุดตัดของตัวควบคุมแบบพีซีพีเอสแควร์ไอดี ส่วนแนวโน้มการลู่เข้าของเกิดที่ประชากรเริ่มต้นที่ 60 90 และ 120 ตัวมีค่าใกล้เคียงกัน แต่แนวโน้มการลู่เข้าของประชากรเริ่มต้นที่ 30 ตัว นั้นดีกว่าประชากรเริ่มต้นที่ 60 90 และ 120 ตัว จึงสรุปได้ว่าไม่จำเป็นเสมอไปที่ประชากรเริ่มต้นที่มากจะได้ผลลัพธ์ที่ดี ส่วนประชากรเริ่มต้นที่ 15 ตัวนั้นมีจำนวนที่น้อยเกินไป

สรุปได้ว่าจำนวนประชากรเริ่มต้นที่ 15 ตัว นั้นมีจำนวนที่น้อยเกินไปในกระบวนการทางพันธุศาสตร์สำหรับกรณีศึกษาในวิทยานิพนธ์นี้ ส่วนประชากรเริ่มต้นที่ 30 60 90 และ 120 ตัว มีแนวโน้มที่ใกล้เคียงกัน ส่วนการเลือกใช้นั้นควรจะเลือกใช้ประชากรเริ่มต้นที่ 30 ตัว เนื่องจากเป็น การลดการคำนวณของฟังก์ชันเป้าหมายในกระบวนการทางพันธุศาสตร์ จากการจำลองผลสุดท้ายแสดงให้เห็นว่าไม่จำเป็นเสมอไปที่ประชากรเริ่มต้นที่มากจะได้ผลลัพธ์ที่ดี ส่วนการเปรียบเทียบกันระหว่างการปรับ แต่งค่าพารามิเตอร์แบบ 8 ตัวแปรกับแบบ 10 ตัวแปรนั้น การปรับแต่งค่าพารามิเตอร์แบบ 10 ตัวแปรจะดีกว่าเพราะมีการปรับจุดสลับของตัวควบคุมแบบพีซีพีเอสแควร์ไอดี การจำลองผลดังกล่าวข้างต้นให้ผลสัมฤทธิ์ที่ดีบ่งชี้ถึงศักยภาพในการนำการออกแบบระบบควบคุมที่นำเสนอthisไปสร้างจริงต่อไป

6.2 ปัญหาที่พบในงานวิจัย

เนื่องจากวิธีทางพันธุศาสตร์สามารถออกแบบได้หลายวิธี จึงต้องมีการออกแบบและทดสอบประสิทธิภาพของแต่ละวิธีเพื่อให้ได้วิธีที่เหมาะสม ซึ่งทำให้เสียเวลาในการทำขั้นตอนนี้

6.3 แนวทางการพัฒนาในอนาคต

การปรับแต่งค่าพารามิเตอร์ โดยใช้วิธีทางพันธุศาสตร์นั้นเป็นแบบ Off-line คือต้องปรับแต่งค่าพารามิเตอร์ก่อน ถึงจะนำไปใช้งานจริง การปรับปรุงในอนาคตจะพัฒนาให้เป็นแบบ On-line เพื่อให้สามารถใช้งานแบบเวลาจริง (Real-Time) และงานวิจัยนี้เป็นการจำลองการทำงาน จึงต้องมีการพัฒนาต่อไปเพื่อให้สามารถนำไปใช้งานจริง

เอกสารอ้างอิง

- [1] H.A .Malki, H. Li, and G.Chen, "New design and stability analysis of fuzzy proportional – derivative control systems," IEEE Trans. Fuzzy Syst., vol. 2, pp. 245-254, nov. 1994.
- [2] G. Chen, "Conventional and fuzzy PID Controllers: An overview," int. J. Intell. Contr. Syst.,vol. 1, no. 2. pp. 235-246, 1996
- [3] P. Sooraksa, T. Pattaradej, and G. Chen "Design and implementation of fuzzy P²ID controller for handlebar control of a bicycle robot." Integrated Computer-Aides Engineering, vol.9, no. 4, 2002, pp. 319-331.
- [4] K.F. man, K.S. Tang and S. Kwong, Genetic Algorithms Concepts and Designs London, U.K. Springer-verlay, 1989.
- [5] D.E. Coldberg, Genetic Algorithms in Search, Optimization and Machin Learning. New York : Addison-wesley, 1989.
- [6] Z. michalewicz, Genetic Algorithms+ Data Structure = Evolution program, 3rd . New York : Springer-Verlay, 1996.
- [7] T. Pattaradej, P. Chandang, M. Lertwatechahakul, P, Sooraksa, and G. Chen, " A Speed Evaluation for Conventional PID and Fuzzy Controllers." The Second Int. Conf., on Intelligent Technologies, Bangkok Thailand, 2001. pp. 91-96.
- [8] G. Chen and H. Ying, "BIBO stability of nonlinear fuzzy PI Control systems," J. of Intel. Fuzzy Syst., vol. 5, 1997. pp. 3-21.
- [9] Halland, J.H., Adaption in natural and artificial systems. MIT. Press, 1975
- [10] C. L. Karr, "Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm", Proc.ICGA 4,pp. 450-457, 1991.
- [11] A. H. Wright, "Genetic Algorithms for Real Parameter Optimization", In Foundation of Genetic Algorithms, J. E. Rawlins (Ed.), Morgan Kaufmann, pp. 205-218, 1991
- [12] Baker, J. E." Reducing Bias and Inefficiency in the Selection Algorithm", in ICGA2, Grefenstette, Proceedings of the Second International Conference on Genetic Algorithms and their Application, Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates, 1987, pp. 14-21,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [13] Mühlenbein, H. and Schlierkamp-Voosen, D." Predictive Models for the Breeder Genetic Algorithm" I. Continuous Parameter Optimization. *Evolutionary Computation*, 1 (1), pp. 25-49, 1993.
- [14] Syswerda, G." Uniform crossover in genetic algorithms" in *ICGA3, Proceedings of the Third International Conference on Genetic Algorithms*, San Mateo, California, USA: Morgan Kaufmann Publishers, pp. 2-9, 1989.
- [15] T. Onboonare, S. Charumartphan, and P. Sooraksa, "A Geno-Fuzzy P^2 ID Controller For Mobile Robots", *ACIAR, Proceedings of The Asian Conference on Industrial Automation and Robotics*, Bangkok, Thailand, pp. 189-192 2003
- [16] A. Visioli, "Fuzzy Logic Based Set-Point Weight Tuning of PID Controllers", *IEEE Trans. Systems and Humans*, vol. 29, no. 6, pp. 587-592, nov. 1999.
- [17] คมสันต์ หงษ์สมบัติ. 2542. "เจเนติกอัลกอริทึมสำหรับช่วยออกแบบตัวควบคุมพีซีซีของตัวชดเชยกำลังไฟฟ้าแบบสถิตย์เพื่อปรับปรุงเสถียรภาพระบบไฟฟ้ากำลัง." *วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.*
- [18] T. Onboonare and P. Sooraksa, "Off-line Tuning Fuzzy P^2 ID Controller Using Genetic Algorithms", *วารสารพระจอมเกล้าลาดกระบัง*, vol. 11, no. 1, April, 2003, pp. 35-42



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

บทความวิจัยที่ได้รับการตีพิมพ์ลงในวารสาร ในวิทยานิพนธ์ คือ

- [1] T. Onboonare, S. Charumartphan, and P. Sooraksa, "A Geno-Fuzzy P²ID Controller For Mobile Robots", ACIAR, Process dings of The Asian Conference on Industrial Automation and Robotics, Bangkok, Thailand, pp. 189-192, 2003.
- [2] T. Onboonare and P. Sooraksa, "Off-line Tuning Fuzzy P²ID Controller Using Genetic Algorithms",วารสารพระจอมเกล้าลาดกระบัง, vol. 11, no. 1, April, 2003, pp. 35-42



ACIAR

2003

PROCEEDINGS OF THE ASIAN CONFERENCE

ON INDUSTRIAL AUTOMATION AND ROBOTICS



8 - 9 May 2003

**Bangkok International Trade
and Exhibition Center
Bangkok, Thailand**

Co-organized by

**School of Advanced Technologies
Asian Institute of Technology**

and

CMP Media (Thailand) Co., Ltd.



A GENO-FUZZY P²ID CONTROLLER FOR MOBILE ROBOTS

Tossapon Onboonare, Suthas Charumartphan, and Pitikhate Sooraksa

Department of Information Engineering

Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Chalongkrung Rd. Ladkrabang Bangkok Thailand 10520

Abstract: This paper deals with tuning parameters for a new fuzzy P²ID controller using genetic algorithms. This type of controller composes of two parts, which are a fuzzy PD controller and a fuzzy PI controller working in parallel processing. A sub-controller is selected one at a time to control the system using switching mechanism performed under given sufficient conditions for stability. Functionally, the fuzzy PD controller is used to launch the output for good rise-time, then, at a designed time, it will switch to the fuzzy PI controller for robustness and good steady state correction. The term "geno-fuzzy P²ID" stems from the combination of all components for producing the controller, which are genetic algorithm, fuzzy PI and fuzzy PD controllers. Given a nonlinear plant, the simulation results show effectiveness of the designed system.

Key words: Fuzzy P²ID, Genetic algorithm

1. INTRODUCTION

P²ID came from the idea of combining Fuzzy PD controller and Fuzzy PI controller[1]. The system has a switch used for changing between two systems. When a whole system starts, Fuzzy PD controller start working first, after then when we reach the assigned point, the system switches to Fuzzy PI mode. The objective of this method is to decrease the output error to be as less as possible. Anyway, in order to achieve this goal, the parameters of Fuzzy PD controller and Fuzzy PI controller has to be adjusted properly. The conventional controller parameter tuning method is known as the Trial-and-Error method. However, the amount of Fuzzy PD and Fuzzy PI parameters are up to eight, so it would take long time to find the appropriate values of those eight parameters by Trial-and-Error technique.

Due to the problem stated above, the technique of Genetic Algorithm [2][3] is proposed in order to find the appropriate values of the system parameters. The results of this technique achieves good system output as the user requires.

This paper applies GA technique into the P²ID controller to find the appropriate parameter by using MATLAB program as a tool for the simulation.

2. FUZZY PI CONTROL

In this section is describe the mathematical principle for the fuzzy PI controller. To get more efficient, if we use the fuzzy logic system to design the DC motor speed control in mobile robot, as shown on Fig. 1. sometimes we don't know any parameters. In this paper, we choose the fuzzy PD system to control mobile robot working with the fuzzy PI to improve the rapid response in the transient interval. We will be using genetic algorithm with tuning parameters of the fuzzy PI and fuzzy PD controller.

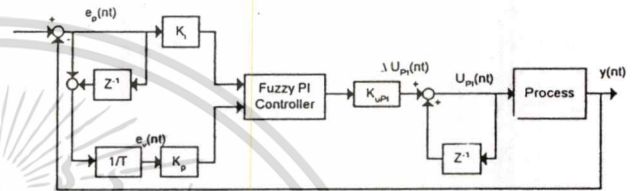


Fig. 1. The Fuzzy PI control system

In the first step, we conclude the concept of the fuzzy PI system structure [4] to understand easily according to the equation below:

$$\Delta u_{PI}(nT) = K_p e_p(nT) + K_i e_p(nT) \quad (1)$$

and

$$u_{PI}(nT) = u_{PI}(nT - T) + K_{uPI} \Delta u_{PI}(nT) \quad (2)$$

which $\Delta u_{PI}(nT)$ is the fuzzy control action of the PI controller, $e_p(nT)$ is the error signal, $e_v(nT)$ is the rate of change of the error signal and, K_p , K_i , K_{uPI} and L are the system gains.

The design of the fuzzy PI system can be possible following to the control rules that can be concluded as integration of nine regions by defining the position error and the speed error, as shown on Fig.2.

$$\begin{aligned} \Delta u_{PI}(nT) &= \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i |e_p(nT)|]} & \text{Region A} \\ &= \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i |e_p(nT)|]} & \text{Region B} \\ &= \frac{1}{2}[K_p e_v(nT) + L] & \text{Region C} \\ &= \frac{1}{2}[K_i e_p(nT) + L] & \text{Region D} \\ &= \frac{1}{2}[K_p e_v(nT) - L] & \text{Region E} \\ &= \frac{1}{2}[K_i e_p(nT) - L] & \text{Region F} \end{aligned} \quad (3) \quad (4) \quad (5) \quad (6) \quad (7) \quad (8)$$

$$= 0 \quad \text{Region G} \quad (9)$$

$$= L \quad \text{Region H}$$

$$= -L \quad \text{Region I} \quad (11) \quad (19)$$

Fig. 2. Regions of the fuzzy PI controller input-combination values

FUZZY PD CONTROL

Similar to the derivation of the fuzzy PI controller discussed in the previous section, the fuzzy PD controller, shown in Fig.3 , is described by

$$\Delta u_{PD}(nT) = K_p e_p(nT) + K_d e_v(nT) \quad (12)$$

and

$$u_{PD}(nT) = -u_{PD}(nT-T) + K_{uPD} \Delta u_{PD}(nT) \quad (13)$$

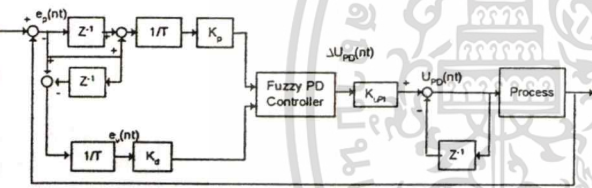


Fig.3 . The fuzzy PD control system.

which $\Delta u_{PD}(nT)$ is the fuzzy control action of the PD controller, $e_p(nT)$ is the error signal, $e_v(nT)$ is the rate of change of the error signal and , K_p , K_d , K_{uPD} , and L are the system gains.

The design of the fuzzy PD system can be possible following to the control rules that can be concluded as integration of nine regions by defining the position error and the speed error, as shown on Fig. 4.

$$u_{PD}(nT) = \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_p |e_p(nT)|]} \quad \text{Region 1} \quad (14)$$

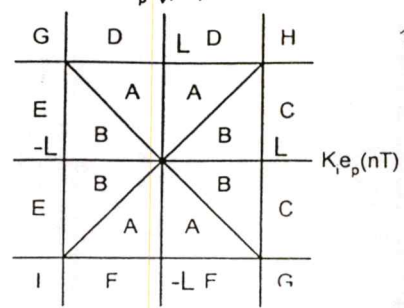
$$= \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_d |e_v(nT)|]} \quad \text{Region 2} \quad (15)$$

$$= \frac{1}{2}[-K_d e_v(nT) + L] \quad \text{Region 3} \quad (16)$$

$$= \frac{1}{2}[K_p e_p(nT) - L] \quad \text{Region 4} \quad (17)$$

$$= \frac{1}{2}[-K_d e_v(nT) - L] \quad \text{Region 5}$$

$$= \frac{1}{2}[K_p e_p(nT) + L] \quad \text{Region 6}$$



$$= 0 \quad \text{Region 7}$$

$$(20)$$

$$= -L \quad \text{Region 8} \quad (21)$$

$$= L \quad \text{Region 9} \quad (22)$$

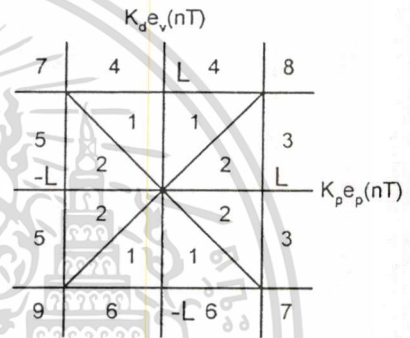


Fig. 4. Regions of the fuzzy PD controller input-combination values

4. FUZZY P²ID CONTROL

Conceptually, we assign the boot-up task to the fuzzy PD controller, which is also responsible for bring the output toward the set-point when the output is beyond the pre-set overshoot limit. The fuzzy PI control, on the other hand, is selected as a second pilot to drive the process output to be within the given tolerance limit, near a certain percentage of the desired steady-state value. The transition point depends on the designer and the process to be controlled.

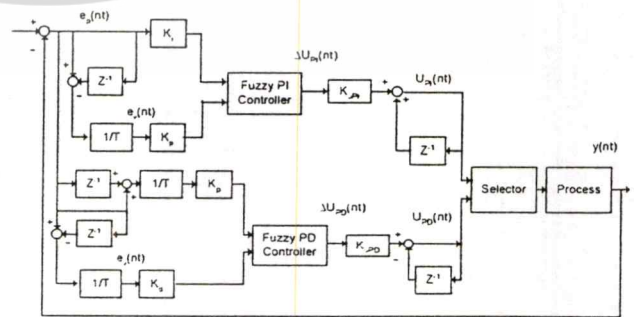


Fig. 5. The fuzzy P²ID control system.

Since these two individual controllers guarantee their local stability, by the small gain theorem as studied in [5][6], the operations are stable after choosing the gains that satisfy the stability criteria locally but for all times expect at the switching points. With this in mind, we consider only the stability at the switching points. Utilizing the error-phase plane show in Fig. 2..4.. we can visualize a principle for designing the switching

rule in the following way: if we consider the areas inside the two phase planes as the attraction regions in the sense of bounded input and bounded output(BIBO) stability[5][6], then at the switching point (see [7]) the change of the control energy at the transition between the two situations must be bounded even though it is inherently discontinuous. Based upon this fact, we must enforce a bounded control transition between the two controllers. In other words, at a switching point, the control energy using by the fuzzy PI controller must be less than that by the fuzzy PD controller[7].

5. SIMPLE GENETIC ALGORITHM

GA presumes that the potential solution of any problem is an individual and can be represented by set of parameter. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form

Genetic operation is consist of crossover, mutation and inversion. Genetic Algorithm have is employed for finding the P²ID controller's parameters. A whole process is divided into three parts. The first part is the method of applying GA to work with fuzzy PD, the second part is the same as the first part but apply to fuzzy PI, and the third part is done by taking the obtained parameters to use in fuzzy P²ID process. The system flowchart can be shown in fig .

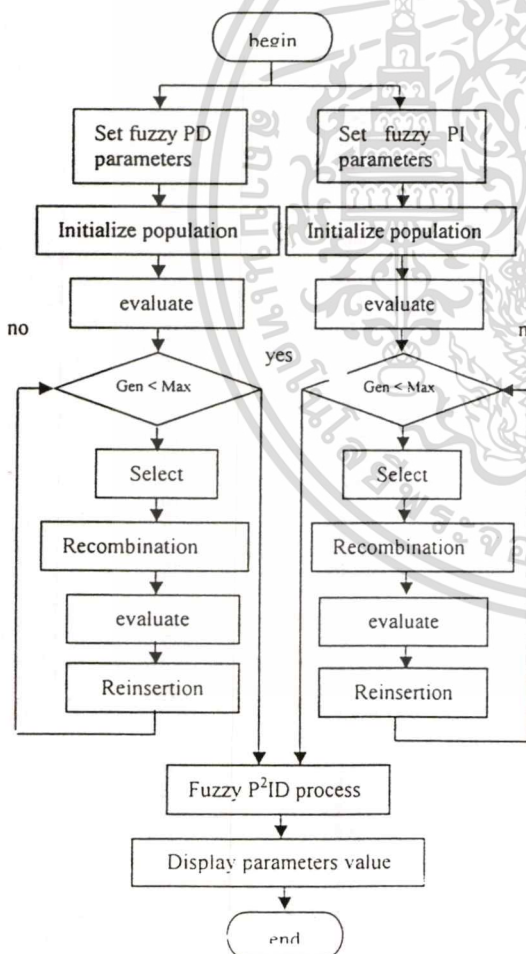


Fig. 6. The method of applying GA to find the P²ID controller's parameter.

From Fig. 6, genetic algorithm is assigned to work in two parts of a whole process. the first part is the tuning of fuzzy PD

parameters and the second part is the tuning of fuzzy PI parameters.

The system design procedure is as follow,

5.1 Chromosome Representation

Fuzzy PD contains four parameters {Kp, Kd, Ku_pd, Ld}. Chromosome I is defined as,

$$I = \{ Kp, Kd, Ku_pd, Ld \}$$

where these parameters, Kp, Kd, Ku_pd, Ld are all real numbers and the amount of I is fixed at 30.

For fuzzy PI, there are four parameters as well, and chromosome I is,

$$I = \{ Kp, Ki, Ku_pi, Li \}$$

where these parameters, Kp, Ki, Ku_pi, Li are all real numbers and the amount of I is fix at 30.

5.2 Objective Function

The objective function consists of two different objectives due to the different output conditions. Fuzzy PD controller needs the shortest time to reach the set point or the smallest rise-time. Fuzzy PI needs the most smooth output signal or the smallest average error signal by using equation IAE.

5.3 Genetic Operation

Genetic Algorithm procedure is applied similarly to both type of controllers as follow,

5.3.1 Choose one pair of "I" from group "I" and combine them.

5.3.2 Take the combined chosen pairs "I" into the crossover and mutation step.

5.3.3 Calculate value "I" of the offspring.

5.3.4 Take 2-3 smallest value "I" and replace in the old "I" group to be a new "I" group again.

6. SIMULATION AND RESULTS

The simulation of non-linear plant, is to compared between the tuning results done by user and by GA. The equation is,

$$\dot{y}(t) = 0.0001|y(t)| + u(t) \quad (23)$$

The set point value is set at 5 and Ts = 0.01. The result parameter is shown in Table1 and can be plotted as a graph in Fig. 7.

Table1. The tuning parameters done by user compared to GA

	Fuzzy PD			
Parameters	Kp	Kd	Ku_pd	Ld
By user	0.2	9	10	3
By GA	0.3034	3.8729	8.4631	9.0183

(a)

	Fuzzy PI			
Parameters	Kp	Ki	Ku_pi	Li
By user	0.1	0.1	8	2
By GA	0.0179	0.0982	9	2

(b)

From Table1. the parameters obtained from GA are compared to the parameter obtained from user tuning. Furthermore, using GA can decrease time amount to find the appropriate values of all eight parameters. Graph is shown in Fig. 7.

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

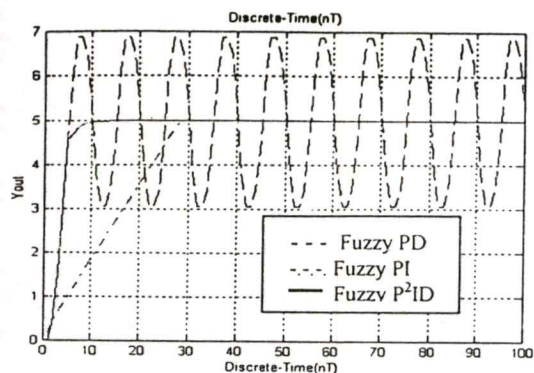


Fig. 7. The output of each controller by using GA.

Fig. 7. shows the result of combining fuzzy PD and fuzzy PI to be fuzzy P²ID by GA. The system starts working by Fuzzy PD until it reaches 4.5, then switches to fuzzy PI.

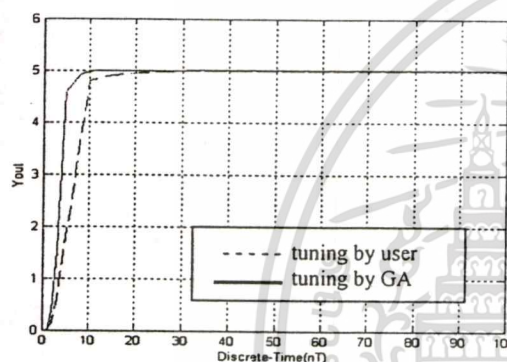


Fig. 8. The output tuning between by user and GA

Figure 8. shows the compared results obtained by user tuning and GA, the results using GA give better performance, as shown in Table2

Table2. The performance of parameter tuning in Nonlinear Model.

	Rise-time(sec)	Setting-time(sec)	Overshoot(%)
By user	0.0672	0.15	0
By GA	0.0285	0.08	0

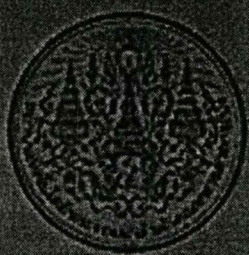
From Table 2, the rise-time and setting-time values obtained from GA tuning is less than from user.

7. CONCLUSIONS

The technique of genetic algorithms has been proposed as a means of Fuzzy P²ID controller. The tuning parameters by GA has been performance more than by user. For another advantage of GA show that output error decrease from optimize.

8. REFERENCES

- [1] G. Chen, "Conventional and fuzzy PID Controllers: An overview," Int. J. Intell. Contr. Syst., vol. 1, no. 2. pp. 235-246, 1996
- [2] K.F. Man, K.S. Tang and S. Kwong, Genetic Algorithms: Concepts and Designs. Springer Verlag, London, 1999
- [3] David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley Publishing Company, January 1989
- [4] Pitikhate Sooraksa, "Fuzzy PI+D Control for Certain Single Flexible-Link Robot Arms with Dampings", 20th Electrical Engineering Conference, Bangkok, 1997, pp.308-312.
- [5] G.Chen and H. Ying, BIBO stability of nonlinear fuzzy PI control systems, J.Intel. Fuzzy Syst. 5(1997), 3-21.
- [6] H.A. Malki, H. Li and G. Chen, New design and stability analysis of fuzzy proportional-derivative control system, IEEE Trans. Fuzzy Systems 2(4) (1994), 345-354
- [7] Pitikhate Sooraksa, Thana Pattaradej and Guanrong Chen, Design and implementation of fuzzy P²ID controller for handlebar control of a bicycle robot, Integrated Computer-Aided Engineering, Vol 9, Number 4, 2002

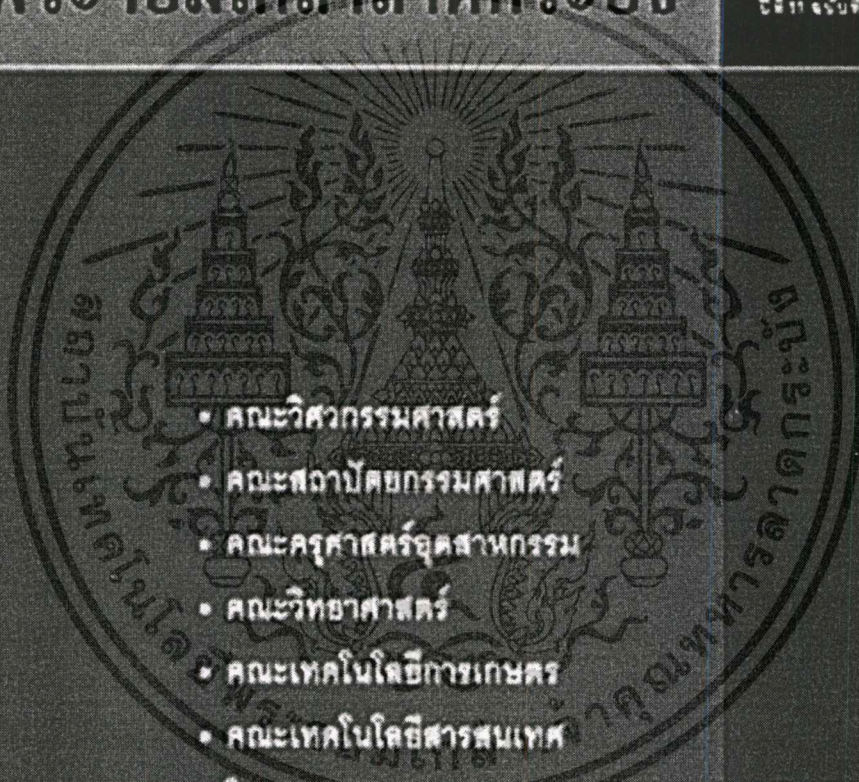


ISSN 0858-8430

วารสารพระจอมเกล้าลาดกระบัง

Vol.11 No.1 April 2003

ปีที่ 11 ฉบับที่ 1 เมษายน 2546



- คณะวิศวกรรมศาสตร์
- คณะสถาปัตยกรรมศาสตร์
- คณะครุศาสตร์อุตสาหกรรม
- คณะวิทยาศาสตร์
- คณะเทคโนโลยีการเกษตร
- คณะเทคโนโลยีสารสนเทศ
- โครงการคณะอุตสาหกรรมเกษตร
- บัณฑิตวิทยาลัย
- วิทยาเขตชุมพร
- สำนักวิจัยและบริการคอมพิวเตอร์
- สำนักหอสมุดกลาง
- โครงการสำนักวิจัยการสื่อสารและเทคโนโลยีสารสนเทศ

Off-line Tuning Fuzzy P²ID Controller Using Genetic Algorithms

Tossapon Onboonare
Lecturer

Pitikhate Sooraksa
Associate Professor, Ph. D.

Department of Information Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Rd. Ladkrabang Bangkok Thailand 10520

Abstract

Off-line tuning parameters for a new fuzzy P²ID controller using genetic algorithms is presented in this paper. This type of controller composes of two parts, which are a fuzzy PD controller and a fuzzy PI controller working in parallel processing. A sub-controller is selected one at a time to control the system using switching mechanism performed under given sufficient conditions for stability. Functionally, the fuzzy PD controller is used to launch the output for good rise-time, then, at a designed time, it will switch to the fuzzy PI controller for robustness and good steady state correction. Given a nonlinear plant and a mobile robot model, the simulation results show effectiveness of the designed system.

Keywords : Fuzzy P²ID, Genetic algorithms

1. INTRODUCTION

Up to present, PID controller has been widely employed in many industrial applications. The disadvantage of the PID controller is that they may not work precisely with the non-linear system or higher order. Later there was a method proposed by CHEN [1] to enhance the system performance by applying Fuzzy Logic into the controller system, which is then called Fuzzy PD and Fuzzy PI. Each controller has their own unique characteristic depends on the type of application used. The development of them is later done by proposing a system that came from the combination of the strong points of each controllers, which is called "P²ID" developed in [7].

P²ID came from the idea of combining Fuzzy PD controller and Fuzzy PI controller. The system has a switch used for changing between two systems. When a whole system starts, Fuzzy PD controller start

working first, after then when we reach the assigned point, the system switches to Fuzzy PI mode. The objective of this method is to decrease the output error to be as less as possible. Anyway, in order to achieve this goal, the parameters of Fuzzy PD controller and Fuzzy PI controller has to be adjusted properly. The conventional controller parameter tuning method is known as the Trial-and-Error method. However, the amount of Fuzzy PD and Fuzzy PI parameters are up to eight, so it would take long time to find the appropriate values of those eight parameters by Trial-and-Error technique.

Due to the problem stated above, the technique of Genetic Algorithm [2][3] is proposed in order to find the appropriate values of the system parameters. The results of this technique achieves good system output as the user requires.

This paper applies GA technique for tuning the

P²ID controller.[7] MATLAB program is used as a tool for the simulation.

2. FUZZY P²ID CONTROL

The proposed Fuzzy P²ID controller is an integrated scheme that combines the fuzzy PD and fuzzy PI control action, where that advantages of control actions are utilized to compensate and complement each individual shortcoming. We first briefly describe architecture of fuzzy PI control and we then review that of fuzzy PD control.

2.1 Fuzzy PI Control

In this section is describe the mathematical principle for the fuzzy PI controller. To get more efficient, if we use the fuzzy logic system to design the DC motor speed control in mobile robot, as shown on Fig.1, sometimes we don't know any parameters. In this paper, we choose the fuzzy PD system to control mobile robot working with the fuzzy PI to improve the rapid response in the transient interval. We will be using genetic algorithm with tuning parameters of the fuzzy PI and fuzzy PD controller.

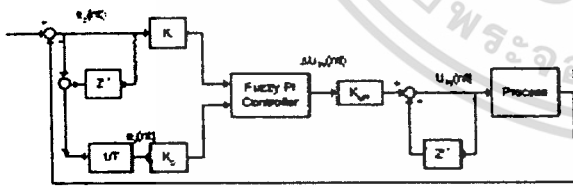


Fig. 1. The Fuzzy PI control system

In the first step, we conclude the concept of the fuzzy PI system structure [4] to understand easily according to the equation below:

$$\Delta u_{PI}(nT) = K_p e_v(nT) + K_i e_p(nT) \quad (1)$$

and

$$u_{PI}(nT) = u_{PI}(nT-T) + K_{uPI} \Delta u_{PI}(nT) \quad (2)$$

which $\Delta u_{PI}(nT)$ is the fuzzy control action of the PI controller, $e_p(nT)$ is the error signal, $e_v(nT)$ is the rate of change of the error signal and, K_p , K_i , K_{uPI} and L are the system gains.

The design of the fuzzy PI system can be possible following to the control rules that can be concluded as integration of nine regions by defining the position error and the speed error, as shown on Fig.2.

$$\Delta u_{PI}(nT)$$

$$= \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i |e_p(nT)|]} \quad \text{Region A} \quad (3)$$

$$= \frac{L[K_i e_p(nT) + K_p e_v(nT)]}{2[2L - K_i |e_v(nT)|]} \quad \text{Region B} \quad (4)$$

$$= \frac{1}{2}[K_p e_v(nT) + L] \quad \text{Region C} \quad (5)$$

$$= \frac{1}{2}[K_i e_p(nT) + L] \quad \text{Region D} \quad (6)$$

$$= \frac{1}{2}[K_p e_v(nT) - L] \quad \text{Region E} \quad (7)$$

$$= \frac{1}{2}[K_i e_p(nT) - L] \quad \text{Region F} \quad (8)$$

$$= 0 \quad \text{Region G} \quad (9)$$

$$= L \quad \text{Region H} \quad (10)$$

$$= -L \quad \text{Region I} \quad (11)$$

		$K_p e_v(nT)$		$K_i e_p(nT)$			
G	D	L	D				
E	-L	B	A				
E	-L	B	A				
I	F	-L	F				
G	D	L	D				
E	-L	B	A				
E	-L	B	A				
I	F	-L	F				

Fig. 2. Regions of the fuzzy PI controller input-combination values

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Fuzzy PD Control

Similar to the derivation of the fuzzy PI controller discussed in the previous section, the fuzzy PD controller, shown in Fig.3, is described by

$$\Delta u_{PD}(nT) = K_p e_p(nT) + K_d e_v(nT) \quad (12)$$

and

$$u_{PD}(nT) = -u_{PD}(nT - T) + K_{uPD} \Delta u_{PD}(nT) \quad (13)$$

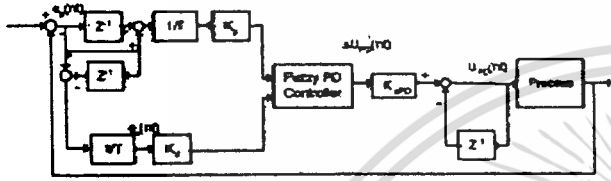


Fig.3. The fuzzy PD control system.

which $\Delta u_{PD}(nT)$ is the fuzzy control action of the PD controller, $e_p(nT)$ is the error signal, $e_v(nT)$ is the rate of change of the error signal and, K_p , K_d , K_{uPD} , and L are the system gains.

The design of the fuzzy PD system can be possible following to the control rules that can be concluded as integration of nine regions by defining the position error and the speed error, as shown on Fig. 4.

$$\Delta u_{PD}(nT)$$

$$= \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_p |e_p(nT)|]} \quad \text{Region 1} \quad (14)$$

$$= \frac{L[K_p e_p(nT) - K_d e_v(nT)]}{2[2L - K_d |e_v(nT)|]} \quad \text{Region 2} \quad (15)$$

$$= \frac{1}{2}[-K_d e_v(nT) + L] \quad \text{Region 3} \quad (16)$$

$$= \frac{1}{2}[K_p e_p(nT) - L] \quad \text{Region 4} \quad (17)$$

$$= \frac{1}{2}[-K_d e_v(nT) - L] \quad \text{Region 5} \quad (18)$$

$$= \frac{1}{2}[K_p e_p(nT) + L] \quad \text{Region 6} \quad (19)$$

$$= 0 \quad \text{Region 7} \quad (20)$$

$$= -L \quad \text{Region 8} \quad (21)$$

$$= L \quad \text{Region 9} \quad (22)$$

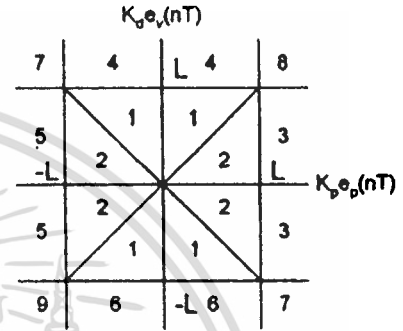


Fig. 4. Regions of the fuzzy PD controller input-combination values

Based on our long-term experience with these fuzzy PD and fuzzy PI controllers, we found that the fuzzy PD controller has faster rise time than the fuzzy PI controller, while the fuzzy PI controller possesses better steady-state accuracy than the other. In fact, this is clear by inspecting the design strategies employed in their control rule base. Hence, it is expected that if these two controllers are combined together in a suitable way, to form a hybrid structure where an appropriate selecting rule is imposed by switching, then a superior performance over each individual controller may be achieved. With this concept, the architecture of the new control algorithm is developed, as depicted in Fig. 5.

Since these two individual controllers guarantee their local stability, by the small gain theorem as studied in [5][6], the operations are stable after choosing the gains that satisfy the stability criteria locally but for all times expect at the switching points. With this in mind, we consider only the stability at the switching points. Utilizing the error-phase plane show in Fig. 2.4., we can visualize a principle for designing the switching rule in the following way: if we consider the areas inside the two phase planes as the attraction regions in the sense of bounded input and bounded output (BIBO) stability[5][6], then at the switching point (see [7]) the change of the control energy at the transition between the two situations must be bounded even though it is inherently discontinuous. Based upon this fact, we must enforce a bounded control transition between the two controllers. In other words, at a switching point, the control energy using by the fuzzy PI controller must be less than that by the fuzzy PD controller]. For more detail, the reader is referred to [7].

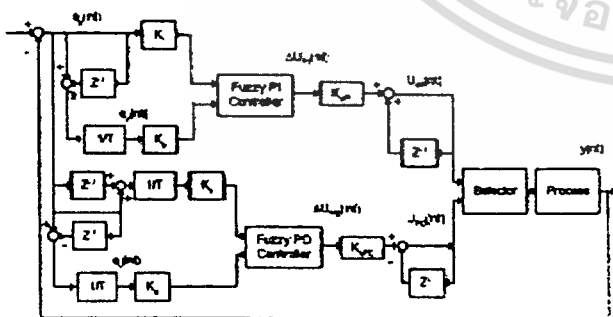


Fig. 5. The fuzzy P²ID control system.

3. SIMPLE GENETIC ALGORITHM

GA is one of optimization method inspired by the mechanism of natural selection where stronger individuals are likely the winners in a competing environment. GA presumes that the potential solution of any problem is an individual and can be represented by set of parameter. These parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form

In a practical GA application, a population pool of chromosome has to be installed and these can be randomly set initially. The size of this population varies from one problem to another. In each cycle of genetic operation, termed as an evolving process, a subsequent generation is created from the chromosomes in the current population. This can only succeed if a group of these chromosomes, generally called parents or a collection term mating pool is select via a specific selection routine. The genes of the parents are mixed and recombined for the production of offspring in the next generation. It is expected that from this process of evolution, the better chromosome will create a larger number of offspring, and thus has a higher chance of surviving in subsequent generation, emulating the survival-of-the-fittest mechanism in nature.

Genetic operation is consist of crossover, mutation and inversion. Genetic Algorithm here is employed for finding the P²ID controller's parameters. A whole process is divided into three parts. The first part is the method of applying GA to work with fuzzy PD, the second part is the same as the first part but apply to fuzzy PI, and the third part is done by taking the obtained parameters to use in fuzzy P²ID process. The system flowchart can be shown in Fig. 6.

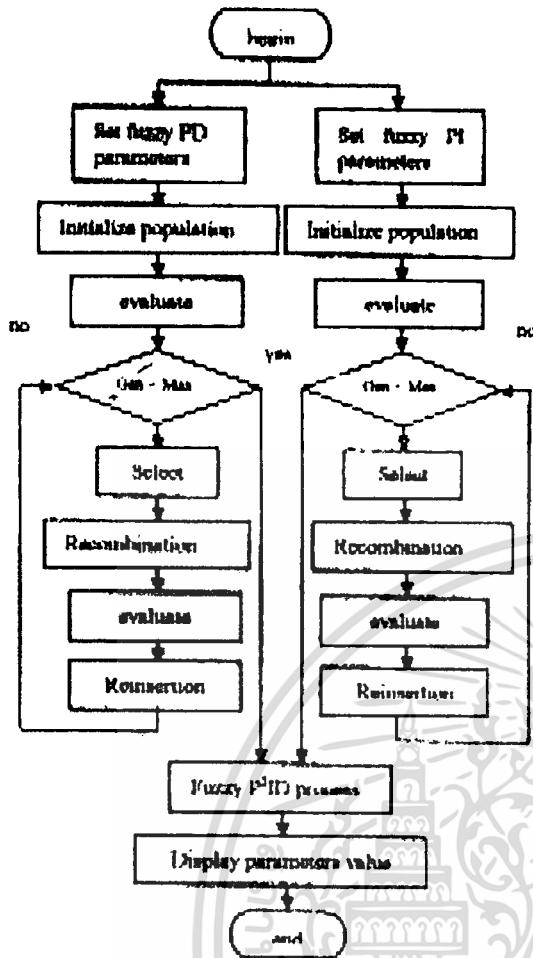


Fig. 6. The method of applying GA to find the P²ID controller's parameter.

From Fig. 6., genetic algorithm is assigned to work in two parts of a whole process, the first part is the tuning of fuzzy PD parameters and the second part is the tuning of fuzzy PI parameters.

The system design procedure is as follow,

3.1 Chromosome Representation

Fuzzy PD contains four parameters {K_p, K_d, K_u_{pd}, L_d}. Chromosome I is defined as,

$$I = \{ K_p, K_d, K_{u_pd}, L_d \}$$

{ K_p, K_d, K_u_{pd}, L_d } is all real numbers and the amount of I is fix at 30.

For fuzzy PI, there are four parameters as well, and chromosome I is,

$$I = \{ K_p, K_i, K_{u_pi}, L_i \}$$

Where these parameters, K_p, K_i, K_u_{pi} and L_i are all real numbers and the amount of I is fixed at 30.

3.2 Objective Function

The objective function consists of two different objectives due to the different output conditions. Fuzzy PD controller needs the shortest time to reach the set point or the smallest rise-time. Fuzzy PI needs the most smooth output signal or the smallest average error signal by using equation IAE.

3.3 Genetic Operation

Genetic Algorithm procedure is applied similarly to both type of controllers as follow,

3.3.1 Choose one pair of "T" from group "T" and combine them.

3.3.2 Take the combined chosen pairs "T" into the crossover and mutation step.

3.3.3 Calculate value "T" of the offspring.

3.3.4 Take 2-3 smallest value "T" and replace in the old "T" group to be a new "T" group again.

4. SIMULATION AND RESULTS

There are two simulations, first is the simulation of non-linear plant and another is the simulation of robot model.

4.1 Nonlinear Model

This simulation is to compared between the tuning results done by user and by GA. The equation is,

$$y(t) = 0.0001|y(t)| + u(t) \quad (23)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The set point value is set at 5 and $T_s = 0.01$ second. The result parameter is shown in Table 1 and can be plotted as a graph in Fig. 7.

Table 1. The gain values obtained from user tuning and from GA.

	Fuzzy PD			
Parameters	Kp	Kd	Ku_pd	Ld
By user	0.2	9	10	3
By GA	0.3034	3.8729	8.4631	9.0183

(a)

	Fuzzy PI			
Parameters	Kp	Ki	Ku_pi	Li
By user	0.1	0.1	8	2
By GA	0.0179	0.0982	9	2

(b)

From Table 1, the parameters obtained from GA are compared to the parameter obtained from a skillful user tuning. Furthermore, using GA can decrease time amount to find the appropriate values of all eight parameters. Graph is shown in Fig. 7.

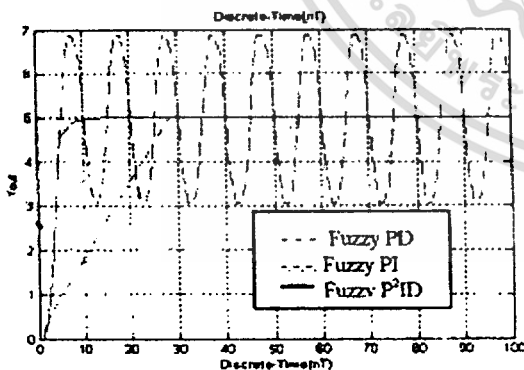


Fig. 7. The output of each controller by using GA.

Fig. 7. shows the result of combining fuzzy PD and fuzzy PI to be fuzzy P^2ID by GA. The system starts working by Fuzzy PD until it reaches 4.5, then switches to fuzzy PI.

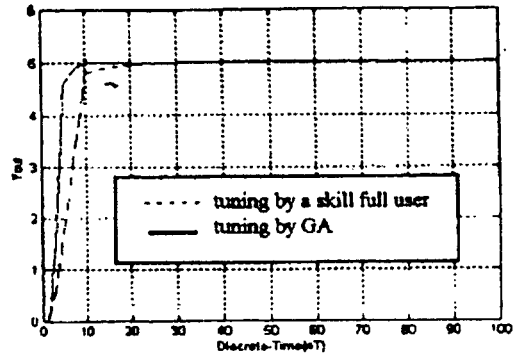


Fig. 8. Comparison of the tuning output by user and by GA.

Fig. 8. shows the compared results obtained by user tuning and GA, the results using GA give better performance, as shown in Table 2.

Table 2. The performance of parameter tuning in nonlinear Model.

	Rise-time(sec)	Setting-time(sec)	Overshoot (%)
By user	0.0672	0.15	0
By GA	0.0285	0.08	0

From Table 2, the rise-time and setting-time values obtained from GA tuning is less than from user.

4.2 Mobile Robot Model

This section compares the parameter tuning by a skillful user and genetic algorithm. The plant to be control is a mobile robot implemented in [8]. Robot's physical parameters are show in Table 3.

Table 3. Robot's physical parameters.

Physical parameters	Value
Wheelbase(D)	6.7 cm
Wheel radius(r)	3.15 cm
Sampling interval(T_s)	0.01second

The equation of motion for the robot is,

$$\theta(k+1) = \theta(k) + \frac{vR(k) - vL(k)}{D} \times Ts \quad (24)$$

The set-point values is set at 90 degree and T_s is 0.01second. The result parameters are shown in Table 4 and can be plotted as a graph in Fig. 9.

Table 4. The gain values obtained from user tuning and from GA.

	Fuzzy PD			
Parameters	Kp	Kd	Ku_pd	Ld
By user	0.7	5	10	20
By GA	0.77	4.45	7	40

(a)

	Fuzzy PI			
Parameters	Kp	Ki	Ku_pi	Li
By user	0.5	0.1	7	10
By GA	0.294	0.1	7	10

(b)

From Table 4, the values obtained from GA is more delicate compared to the values from user tuning. Furthermore, using GA can decrease time amount to find the appropriate values for eight controllers. The response can be plotted in Fig. 9.

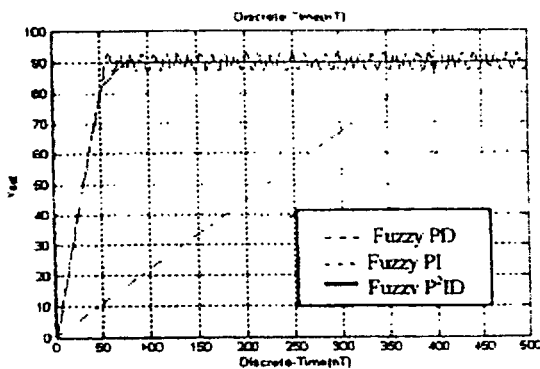


Fig. 9. The output of each controller by using GA. in this paper

Fig. 9. shows the result of combining fuzzy PD and fuzzy PI to be fuzzy P²ID by using GA tuning. The system starts working by Fuzzy PD until it reach 81, then switches to fuzzy PI.

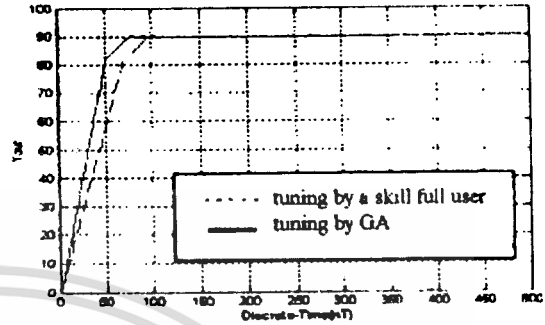


Fig. 10. Comparison of the tuning output by a skillful user and by GA.

Fig. 10. shows the compared results obtained by a skillful user tuning and GA, the results using GA gives better performance, as shown in Table 5.

Table 5. The performance of mobile robot model tuning

	Rise-time(sec)	Setting-time(sec)	Overshoot (%)
By user	0.6046	0.92	0
By GA	0.4260	0.71	0

From Table 5, the rise-time values and setting-time values using GA is smaller than the values adjusted by a skillful user.

5. CONCLUSIONS

We applied the technique of genetic algorithms for tuning Fuzzy P²ID controller developed in [7]. The tuning parameters by GA has been performed remarkably well, compared to tuning by a skillful user. Next step should be focused on a real-world implementation using the tuning method demonstrated

ACKNOWLEDGEMENT

This research is supported in part by The Thailand Research Fund.

REFERENCES

- [1] G. Chen, "Conventional and fuzzy PID Controllers: An overview," *Int. J. Intell. Contr. Syst.*, vol. 1, no. 2. pp. 235-246, 1996
- [2] K.F. Man, K.S. Tang and S. Kwong, *Genetic Algorithms: Concepts and Designs*, Springer Verlag, London, 1999
- [3] D E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Publishing Company, January 1989
- [4] P Sooraksa, "Fuzzy PI+D Control for Certain Single Flexible-Link Robot Arms with Dampings", 20th Electrical Engineering Conference, Bangkok, 1997, pp.308-312.
- [5] G.Chen and H. Ying, BIBO stability of nonlinear fuzzy PI control systems, *J.Intel. Fuzzy Syst.* 5(1997), 3-21.
- [6] H.A. Malki, H. Li and G. Chen, New design and stability analysis of fuzzy proportional-derivative control system, *IEEE Trans. Fuzzy Systems* 2(4) (1994), 345-354
- [7] P Sooraksa, T Pattaradej and G Chen, Design and implementation of fuzzy P²ID controller for handlebar control of a bicycle robot, *Integrated Computer-Aided Engineering*, Vol 9, Number 4, 2002
- [8] M.-J. Jung, H.-S. Kim, H.-S. Shim and J.-H. Kim, Fuzzy Rule Extraction for Shooting Action Controller for Soccer Robot, Dept. of Electrical Engineering Korea Advanced Institute of Science and Tech.

ประวัติผู้เขียน

นายทศพล อ่อนบุญเชื้อ เกิดเมื่อวันที่ 24 มกราคม 2520 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาในระดับปริญญาตรี หลักสูตรอุตสาหกรรมศาสตรบัณฑิต สาขาเทคโนโลยีอิเล็กทรอนิกส์ ปีการศึกษา 2542 และ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543 และในปีการศึกษา 2544 ได้เข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปัจจุบันเป็นอาจารย์ประจำอยู่ที่ สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอีสเทิร์นเอเซีย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้