



การปรับค่าตัวควบคุม PID ด้วยขั้นตอนวิธีหาค่าที่เหมาะสม

Ant Colony Optimization for PID Controller Adjustment



นายณัฐพงศ์	พงศ์ไพจิตรวงศ์	รหัส 50360913
นายวีระพล	บัวแก้ว	รหัส 50365284

คณะวิศวกรรมศาสตร์
สำหรับ 11 ส.ค. 2555
เลขทะเบียน..... 15734058
เลขเรียกหนังสือ..... ผ.ร.
มหาวิทยาลัยนเรศวร ๒๖33๙ ๑

2553

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร


ปีการศึกษา 2553

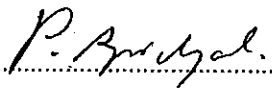


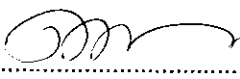
ใบรับรองโครงการ

หัวข้อโครงการ การปรับค่าตัวควบคุม PID ด้วยขั้นตอนวิธีอานานิคมมด
ผู้ดำเนินโครงการ นายณัฐพงศ์ พงศ์ไพจิตรวงศ์ รหัส 50360913
 นายวีระพล บัวแก้ว รหัส 50365284
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ธนิต มาลากร
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2553

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของ
การศึกษาดำเนินหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์


.....ที่ปรึกษาโครงการ
(ผู้ช่วยศาสตราจารย์ ดร.ธนิต มาลากร)


.....กรรมการ
(ดร.พิศุทธิ์ อภิษยกุล)


.....กรรมการ
(ดร.พรพิศุทธิ์ วรจิรันตน์)

หัวข้อโครงการ	การปรับปรุงตัวควบคุม PID ด้วยขั้นตอนวิธีอานานิคมมด		
ผู้ดำเนินโครงการ	นายณัฐพงศ์ พงศ์ไพจิตรวงศ์	รหัส 50360913	
	นายวีระพล บัวแก้ว	รหัส 50365284	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ธนิต มาลากร		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2553		

บทคัดย่อ

ตัวควบคุมแบบ PID ถูกนำมาใช้ในอุตสาหกรรมรวมไปถึงการควบคุมกระบวนการต่างๆ ซึ่งตัวควบคุมแบบ PID ดังกล่าวนี้อาศัยพารามิเตอร์ 3 ชนิด นั่นคือ K_p , K_i และ K_d การปรับค่าพารามิเตอร์ทั้งสามดังกล่าว ส่งผลให้ตัวควบคุมแบบ PID สามารถส่งสัญญาณควบคุมเพื่อให้ผลตอบสนองต่อระบบเป็นไปตามความต้องการ ทั้งเรื่องของ เวลาขึ้น ค่าพุ่งสูงสุด และค่าผิดพลาดที่สภาวะคงตัว อย่างไรก็ตามการปรับพารามิเตอร์ของ PID ทั้งสามนี้ต้องอาศัยการลองผิดลองถูก ซึ่งเสียเวลาค่อนข้างมาก เพื่อแก้ปัญหาเรื่องนี้จึงได้มีการคิดค้นวิธีในการปรับพารามิเตอร์ ซึ่งบางวิธีได้รับความนิยมน้อยแพร่หลาย เช่น ซิกเมตอร์ - นิโคส (ZN), เซน - ฮรอน - เรสวิก (CHR) และ หวาง - จวง - ฉาน (WJC) แม้ว่าวิธีการดังกล่าวข้างต้นสามารถแก้ปัญหาการลองผิดลองถูกลงไปได้ แต่ปัญหาที่เกิดขึ้นใหม่ คือ ผลตอบสนองมักขึ้นอยู่กับเกณฑ์สมรรถนะของแต่ละวิธี แนวทางการพัฒนาในการปรับพารามิเตอร์ K_p , K_i และ K_d แนวทางหนึ่งคือ การใช้เทคนิคการหาค่าเหมาะที่สุด ได้แก่ การค้นหาแบบกลไกคัดสรรคัพันธ์ที่ดีตามธรรมชาติ (GA), การค้นหาแบบการจับกลุ่มของอนุภาค (PSO), การค้นหาแบบการสลับของภพ (SFL) และการค้นหาแบบอานานิคมมด (ACO)

โครงการนี้เป็น การนำขั้นตอนวิธีระบบอานานิคมมด (ACS) มาใช้ในการปรับตัวควบคุมแบบ PID ซึ่งวิธีการดังกล่าวได้ลอกเลียนแบบพฤติกรรมของมดในการค้นหาอาหารตามธรรมชาติ ผลจากการจำลองโดยคอมพิวเตอร์ แสดงให้เห็นว่า ตัวควบคุมแบบ PID ที่ถูกออกแบบด้วย ACS ให้ผลตอบสนองที่ดีกว่าการออกแบบด้วยวิธีแบบดั้งเดิม

Project Title Ant Colony Optimization for PID Controller Adjustment
Name Mr. Nuttaphong Phongpijitwong ID 50360913
Mr. Weerapon Buakaew ID 50365284
Project Advisor Asst. Prof. Tanit Malakorn, Ph.D.
Major Computer Engineering
Department Electrical and Computer Engineering
Academic year 2010

ABSTRACT

A proportional – integral – derivative controller (PID controller) is commonly used in industry and applicable to numerous control processes. The PID controller involves three constant parameters: K_p , K_i and K_d . By tuning the three parameters, the PID controller can provide control action designed for specific process requirements, such as rise time, overshoot, steady-state error. However, these PID parameters are often turned manually by trial and error which is time-consuming. To conquer this problem, several methods have been proposed in the literature; some of which have been known and applied widely, such as Ziegler-Nichols (ZN), Chien-Hrones-Reswick (CHR) and Wang-Juang-Chan (WJC). The main problem is that the time response obtained via these methods only satisfies the performance criteria for that particular method. An improvement in tuning can be achieved using optimization techniques--namely Genetic Algorithm (GA), Particle Swam Optimization (PSO), Shuffled-Frog Leaping Algorithm (SFL), and Ant Colony Optimization (ACO).

This project considers the use of Ant Colony System (ACS) for tuning a PID controller. This method imitates the ant behavior in the nature to search for food. The simulation results illustrate that the PID controller designed by ACS yields the time response characteristics better than that designed by the classical ones.

กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์เรื่องการปรับค่าตัวควบคุม PID ด้วยขั้นตอนวิธีอานานิคมมดฉบับนี้ได้สำเร็จลุล่วงได้ เนื่องจากคณะผู้จัดทำได้รับความอนุเคราะห์จากผู้ช่วยศาสตราจารย์ ดร. ธนิต มาลากร ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้ ที่กรุณาให้คำปรึกษา แนะนำวิธีการทำงาน พร้อมทั้งแนะแนวทางในการแก้ปัญหาต่าง ๆ และคอยกระตุ้นให้คณะผู้จัดทำทำงานอย่างต่อเนื่อง ตลอดจนสละเวลาอันมีค่าเพื่อตรวจสอบและแก้ไขข้อบกพร่องต่างๆ คณะผู้จัดทำรู้สึกเป็นเกียรติอย่างมากที่ได้รับความอนุเคราะห์จากอาจารย์

ในโอกาสนี้ทางคณะผู้จัดทำโครงการจึงขอขอบพระคุณอาจารย์ทุกท่าน บิดา มารดา ที่คอยสั่งสอนให้ความรู้จนคณะผู้จัดทำสำเร็จการศึกษา ทั้งนี้ขอขอบคุณเพื่อน ๆ ที่ได้ช่วยในการทดสอบโปรแกรม และคอยให้กำลังใจ ช่วยให้คำปรึกษาทั้งในเรื่องเรียนและในเรื่องส่วนตัวจนสำเร็จลุล่วงมาด้วยดี



นายณัฐพงศ์

นายวีระพล

พงศัไพจิตรวงศ์

บัวแก้ว

สารบัญ

	หน้า
ใบรับรองโครงการวิจัย.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ญ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบข่ายของโครงการ.....	2
1.4 ขั้นตอนและแผนการดำเนินงาน.....	3
1.5 ผลที่คาดว่าจะได้รับ.....	4
1.6 งบประมาณ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ระบบควบคุมเบื้องต้น.....	5
2.1.1 ฟังก์ชันถ่ายโอน (Transfer Function).....	5
2.1.2 ปริภูมิสถานะ (State Space).....	6
2.1.3 ความสัมพันธ์ระหว่างปริภูมิสถานะกับฟังก์ชันถ่ายโอน.....	6
2.1.3.1 การแปลงระบบจากปริภูมิสถานะให้อยู่ในรูปฟังก์ชันถ่ายโอน.....	7
2.1.3.2 การแปลงระบบจากฟังก์ชันถ่ายโอนให้อยู่ในรูปของปริภูมิสถานะ.....	8
2.2 ผลตอบสนองเชิงเวลา.....	9
2.2.1 ผลตอบสนองชั่วขณะ.....	9
2.2.2 ผลตอบสนองที่สภาวะคงตัว.....	11

สารบัญ (ต่อ)

	หน้า
2.3 การออกแบบตัวควบคุม PID	15
2.3.1 วิธีการของซีกเลอร์ – นิโกล์ (Ziegler – Nichols: ZN)	15
2.3.1.1 วิธีการที่หนึ่ง (First Method)	15
2.3.1.2 วิธีการที่สอง (Second Method)	16
2.3.2 วิธีการของเชน-ฮรอน-เรสวิก (Chien–Hrones–Reswick: CHR)	17
2.3.3 วิธีการของหวาง – จวง – ฉาน (Wang – Juang – Chan: WJC)	18
2.4 การค้นหาผลเฉลยแบบศึกษาสำนึก (Heuristic Search)	25
2.4.1 การค้นหาแบบทาบู (Taboo Search: TS)	25
2.4.2 การค้นหาแบบการซิมูเลเทดแอนนีลลิ่ง (Simulated Annealing: SA)	26
2.4.3 การค้นหาแบบฮาร์โมนี (Harmony Search)	27
2.4.4 การค้นหาแบบการจับกลุ่มของอนุภาค (Particle Swarm Optimization: PSO)	27
2.4.5 การค้นหาแบบกลไกคัดสรรพันธุ์ที่ดีตามธรรมชาติ (Genetic Algorithms: GA)	29
2.4.6 การค้นหาแบบการสลัดของกบ (Shuffled Frog Leaping Algorithm: SFL)	30
2.4.7 การค้นหาแบบอาณานิคมมด (Ant Colony Optimization: ACO)	30
บทที่ 3 ขั้นตอนการดำเนินงาน	33
3.1 การประยุกต์ระบบของอาณานิคมมด	33
3.2 ตัวอย่างการประยุกต์ระบบของอาณานิคมมด	44
3.3 ขั้นตอนการออกแบบโปรแกรม GUI	59
บทที่ 4 ผลการทดลอง	63
4.1 ผลการทดลองกับระบบที่ 1	63
4.2 ผลการทดลองกับระบบที่ 2	66
4.3 ผลการทดลองกับระบบที่ 3	68
4.4 ผลการทดลองกับระบบที่ 4	70

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปและข้อเสนอแนะ.....	72
5.1 สรุปผลการดำเนินงาน.....	72
5.2 ข้อเสนอแนะและแนวทางการพัฒนาต่อ.....	72
เอกสารอ้างอิง	73
ภาคผนวก การใช้งาน โปรแกรม.....	74
ประวัติผู้ดำเนินโครงการ	80



สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน.....	3
2.1 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่หนึ่ง).....	16
2.2 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่สอง).....	17
2.3 การหาค่า R.....	17
2.4 การหาค่า K_p , K_i และ K_d ด้วยวิธี CHR.....	18
2.5 การหาค่า K_p , T_i และ T_d ด้วยวิธี WJC.....	18
2.6 การคำนวณค่าพารามิเตอร์แบบ PID ด้วยขั้นตอนวิธี ZN.....	20
2.7 พารามิเตอร์ที่ใช้ในการคำนวณด้วยวิธี CHR.....	21
2.8 การคำนวณค่าพารามิเตอร์แบบ PID ด้วยขั้นตอนวิธี CHR.....	21
2.9 พารามิเตอร์ที่ใช้ในการคำนวณด้วยวิธี WJC.....	23
2.10 การหาค่า K_p , T_i และ T_d ด้วยขั้นตอนวิธี WJC.....	23
2.11 ACO และผู้คิดค้น.....	32
3.1 การคำนวณค่าความน่าจะเป็น.....	38
3.2 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=1$ ของรอบที่ 1.....	47
3.3 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=2$ ของรอบที่ 1.....	48
3.4 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=3$ ของรอบที่ 1.....	49
3.5 การบันทึกเส้นทางของมดแต่ละตัวในรอบที่ 1.....	50
3.6 ค่า K_p , K_i และ K_d ที่ได้จากมดแต่ละตัวในรอบที่ 1.....	51
3.7 การคำนวณค่าความคลาดเคลื่อนในรอบที่ 1.....	51
3.8 ตำแหน่งที่ทำการปรับพีโร โมนในรอบที่ 1.....	53
3.9 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=1$ ของรอบที่ 2.....	53
3.10 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=2$ ของรอบที่ 2.....	54
3.11 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=3$ ของรอบที่ 2.....	55
3.12 การบันทึกเส้นทางของมดแต่ละตัวในรอบที่ 2.....	56
3.13 ค่า K_p , K_i และ K_d ที่ได้จากมดแต่ละตัวในรอบที่ 2.....	57
3.14 การคำนวณค่าความคลาดเคลื่อนในรอบที่ 2.....	57
3.15 ตำแหน่งที่ทำการปรับพีโร โมนในรอบที่ 2.....	58
3.16 ตำแหน่งที่ทำการปรับพีโร โมนในรอบที่ 2 (ต่อ).....	59

สารบัญตาราง(ต่อ)

ตารางที่	หน้า
3.17 ค่าพารามิเตอร์เริ่มต้นของโปรแกรมสำเร็จ.....	60
4.1 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 1 ด้วยวิธี ZN	64
4.2 ผลการทดลองของระบบที่ 1.....	65
4.3 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 2.....	66
4.4 ผลการทดลองของระบบที่ 2.....	67
4.5 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 3.....	68
4.6 ผลการทดลองของระบบที่ 3.....	69
4.7 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 4.....	70
4.8 ผลการทดลองของระบบที่ 4.....	71



สารบัญรูป

รูปที่	หน้า
2.1 ผลการตอบสนองชั่วขณะของระบบอันดับสอง	9
2.2 ผลตอบสนองของระบบ, สัญญาณอ้างอิง และสัญญาณความผิดพลาด	12
2.3 แผนภาพกล่องของระบบปิด	13
2.4 ผลตอบสนองรูปตัวเอส	15
2.5 แผนภาพกล่องของระบบวงปิดที่มีตัวควบคุม	16
2.6 ผลตอบสนองที่เกิดจากปรับค่า K จนเกิดการแกว่งกวัด	17
2.7 แผนภาพกล่องแสดงการป้อนกลับของฟังก์ชันถ่ายโอน	19
2.8 ผลตอบสนองของระบบตัวอย่าง $K^* = 11.25$	19
2.9 ผลตอบสนองของระบบควบคุมแบบ ZN	20
2.10 การหาค่า K , L และ T จากผลตอบสนองของระบบ	21
2.11 ผลตอบสนองของระบบควบคุมแบบ CHR ด้วยวิธี Overshoot ร้อยละ 20	22
2.12 ผลตอบสนองของระบบควบคุมแบบ CHR ด้วยวิธี Least Overshoot	23
2.13 ผลตอบสนองที่มีระบบควบคุมแบบ WJC	24
2.14 การเปรียบเทียบผลตอบสนองของระบบที่มีตัวควบคุมแบบ PID	25
2.15 แผนภาพแสดงการเลือกทางเดินที่สั้นที่สุดของมด	31
3.1 เส้นทางเดินของมด	35
3.2 ตัวอย่างทำการปล่อยมดในเส้นทางเดิน	36
3.3 แสดงรหัสเทียมของฟังก์ชันหลัก	40
3.4 แสดงรหัสเทียมของฟังก์ชันหาเส้นทาง	41
3.5 แสดงรหัสเทียมของฟังก์ชันหาความคลาดเคลื่อน	41
3.6 แสดงรหัสเทียมของฟังก์ชันปรับฟีโรโมน (Update pheromone)	41
3.7 แสดงรหัสเทียมของฟังก์ชันปรับเส้นทางอ้างอิง	41
3.8 แผนภาพการปรับพารามิเตอร์ PID ด้วยขั้นตอนวิธีของ ACS	42
3.9 ตารางเส้นทางเดินของมด	45
3.10 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 1	46
3.11 จุดเริ่มต้นของเส้นทาง	46
3.12 ความน่าจะเป็นที่มดจะเลือก ณ จุด $x = 1$	47

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.13 เส้นทางการเดินของมดช่วงที่ 1 ของมดตัวที่ 1 ในรอบที่ 1.....	48
3.14 เส้นทางการเดินของมดช่วงที่ 2 ของมดตัวที่ 1 ในรอบที่ 1.....	49
3.15 เส้นทางการเดินของมดช่วงที่ 3 ของมดตัวที่ 1 ในรอบที่ 1.....	50
3.16 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 2.....	52
3.17 เส้นทางการเดินของมดช่วงที่ 1 ของมดตัวที่ 1 ในรอบที่ 2.....	54
3.18 เส้นทางการเดินของมดช่วงที่ 2 ของมดตัวที่ 1 ในรอบที่ 2.....	55
3.19 เส้นทางการเดินของมดช่วงที่ 3 ของมดตัวที่ 1 ในรอบที่ 2.....	56
3.20 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 3.....	58
3.21 ผลตอบสนองของระบบที่ใช้ ACS เมื่อเทียบกับการออกแบบด้วยวิธี ZN.....	59
3.22 แผนภาพกล่องของระบบป้อนกลับแบบหนึ่งที่ไม่มีตัวควบคุม.....	61
3.23 แผนภาพกล่องของระบบป้อนกลับแบบหนึ่งที่มีตัวควบคุมแบบ PID.....	61
3.24 หน้าจอของโปรแกรม.....	61
4.1 หน้าต่างโปรแกรมปรับพารามิเตอร์ด้วยขั้นตอนวิธีของ ACS.....	63
4.2 แสดงการป้อนข้อมูลในโปรแกรมของระบบที่ 1.....	64
4.3 แสดงผลลัพธ์โปรแกรมของระบบที่ 1.....	65
4.4 ผลตอบสนองของระบบที่ 2.....	66
4.5 แสดงผลลัพธ์โปรแกรมของระบบที่ 2.....	67
4.6 ผลตอบสนองของระบบที่ 3.....	68
4.7 แสดงผลลัพธ์โปรแกรมของระบบที่ 3.....	69
4.8 ผลตอบสนองของระบบที่ 4.....	70
4.9 แสดงผลลัพธ์โปรแกรมของระบบที่ 4.....	71
ก.1 ไฟล์ Setup.....	74
ก.2 หน้าแรกของโปรแกรม.....	74
ก.3 ทำการเลือกโฟลเดอร์ที่ต้องการติดตั้งโปรแกรม.....	75
ก.4 ทำการสร้างไอคอนไว้ที่หน้าจอ.....	75
ก.5 เริ่มการติดตั้งโปรแกรม.....	76
ก.6 การติดตั้งโปรแกรมเสร็จสิ้น.....	76

สารบัญรูป(ต่อ)

รูปที่	หน้า
ก.7 โฟลเดอร์ที่ทำการติดตั้งโปรแกรม.....	77
ก.8 หน้าต่างของโปรแกรม.....	77
ก.9 ทำการป้อนข้อมูล.....	78
ก.10 ผลการคำนวณของโปรแกรม.....	79



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

ปัญหาทางด้านวิศวกรรมหลายปัญหาจัดว่าเป็นปัญหาการหาค่าเหมาะที่สุด (Optimization problem) ยกตัวอย่างเช่น การออกแบบระบบขับเคลื่อนเพื่อใช้เชื้อเพลิงน้อยที่สุด การหาระยะทางที่ใช้เวลาเดินทางน้อยที่สุด หรือการจัดวางเครื่องจักรในโรงงานเพื่อให้สามารถผลิตชิ้นงานได้มากที่สุดโดยใช้ระยะเวลาน้อยที่สุด เป็นต้น การหาผลเฉลยสำหรับปัญหาประเภทนี้มีอยู่ 2 ลักษณะ กล่าวคือ การคำนวณเชิงวิเคราะห์ (Analytical method) และการค้นหาผลเฉลย (Searching method) สำหรับการคำนวณเชิงวิเคราะห์นั้น ต้องอาศัยการสร้างรูปแบบทางคณิตศาสตร์เพื่อใช้บรรยายพฤติกรรมของปัญหา จากนั้นจึงนำหลักการทางคณิตศาสตร์มาประยุกต์ใช้ในการหาผลเฉลย อย่างไรก็ตาม พบว่าการสร้างรูปแบบทางคณิตศาสตร์ดังกล่าวนี้ มีความยุ่งยากซับซ้อน นอกจากนี้ยังพบว่าหลายปัญหาไม่สามารถหาผลเฉลยได้ภายในระยะเวลาที่จำกัด เช่น ปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem: TSP) ปัญหาการจัดตารางเวลา (Timetabling Problem) รวมทั้งปัญหาการวางแผนการขนส่ง (Vehicle routing) เป็นต้น

ในการหาผลเฉลยด้วยการค้นหา เริ่มจากการกำหนดเซตของผลเฉลยที่เป็นไปได้ (Feasible solution) ทั้งหมด จากนั้นจึงใช้ขั้นตอนวิธี (Algorithm) ในการค้นหาผลเฉลยที่เหมาะสมที่สุด (Optimal solution) จากผลเฉลยในเซตดังกล่าว ขั้นตอนวิธีที่นิยมเลือกใช้ ได้แก่ การค้นหาแนวกว้างก่อน (Breadth first search: BFS) หรือการค้นหาแนวลึกก่อน (Depth first search: DFS) เป็นต้น อย่างไรก็ตาม หากปัญหาที่พิจารณามีผลเฉลยที่เป็นไปได้อยู่เป็นจำนวนมาก การค้นหาผลเฉลยดังกล่าวอาจใช้เวลานาน ด้วยเหตุนี้ นักวิจัยกลุ่มหนึ่งจึงนำเสนอแนวทางการค้นหาผลเฉลยแบบศึกษาสำนึก (Heuristic search) ซึ่งเป็นการค้นหาผลเฉลยบางตัว จากนั้น จึงนำข้อมูลที่ได้จากผลเฉลยเหล่านั้นมาปรับปรุงเพื่อใช้ค้นหาผลเฉลยที่เหลือ จนในที่สุดทำให้ได้ผลเฉลยที่ใกล้เคียงกับผลเฉลยที่เหมาะสมที่สุดนั่นเอง ขั้นตอนวิธีในการค้นหาผลเฉลยแบบศึกษาสำนึก ได้แก่ การค้นหาแบบไต่เขา (Hill climbing Search) การค้นหาดีที่สุดก่อน (Best first search) การค้นหาแบบ A* (A* search) ขั้นตอนวิธีแบบอาณานิคม (Ant colony algorithm) ขั้นตอนวิธีแบบการสลัดที่ของกบ (Shuffle frog algorithm) ขั้นตอนวิธีเชิงกรรมพันธุ์ (Genetic algorithm) เป็นต้น

โครงการนี้เป็นการศึกษาแนวทางการปรับปรุงพารามิเตอร์ K_p , K_i และ K_d ในตัวควบคุมแบบ PID โดยประยุกต์ใช้ขั้นตอนวิธีแบบอาณานิคม เนื่องจาก การคำนวณหาค่าพารามิเตอร์ K_p , K_i และ K_d ในระบบควบคุมที่ไม่มีกฏตายตัว ส่วนใหญ่ต้องอาศัย

ประสบการณ์ของผู้ออกแบบเป็นหลัก ซึ่งทำให้ระบบควบคุมแบบ PID ได้ผลตอบสนองยังไม่เป็นที่น่าพอใจ จึงจำเป็นต้องปรับแต่งค่าพารามิเตอร์ดังกล่าวเพื่อให้ได้ผลตอบสนองดีขึ้น

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อให้เกิดความเข้าใจในการออกแบบตัวควบคุมระบบ PID ของระบบเชิงเส้นไม่แปรตามเวลา
2. เพื่อศึกษาและประยุกต์ใช้การค้นหาผลเฉลยแบบศึกษาสำนึกในงานด้านวิศวกรรม
3. เพื่อศึกษาและพัฒนาการใช้งานในส่วนประสานกราฟิกกับผู้ใช้ (Graphic User Interface : GUI) ในโปรแกรม MATLAB

1.3 ขอบข่ายของโครงการ

1. ระบบที่ศึกษาต้องเป็นระบบเชิงเส้นไม่แปรตามเวลา
2. ใช้ขั้นตอนวิธีแบบอาณานิคมมดในการปรับปรุงพารามิเตอร์ K_p , K_i และ K_d ของตัวควบคุมแบบ PID
3. โปรแกรมที่ได้ต้องเป็นส่วนประสานกราฟิกกับผู้ใช้ (Graphic User Interface: GUI) ในโปรแกรม MATLAB ที่ง่ายต่อการใช้งานสำหรับผู้ใช้ทั่วไป

1.5 ผลที่คาดว่าจะได้รับจากโครงการ

1. ได้รับความรู้เกี่ยวกับหลักการค้นหาผลเฉลยแบบศึกษาสำนึก
2. สามารถประยุกต์ใช้ขั้นตอนวิธีแบบอาณานิคมมดกับปัญหาทางวิศวกรรมได้
3. ได้โปรแกรมสำเร็จสำหรับการปรับปรุงพารามิเตอร์ K_p , K_i และ K_d ของตัวควบคุมแบบ PID ในรูปแบบของ GUI

1.6 งบประมาณ

ค่าวัสดุสำนักงาน	เป็นเงิน	1,000	บาท
ค่าถ่ายเอกสารและเข้าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	1,000	บาท
...	รวมเป็นเงินทั้งสิ้น	2,000	บาท
			(สองพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ



บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

ในการศึกษาและประยุกต์ใช้ขั้นตอนวิธีแบบอาณานิคมมด (Ant Colony Optimization: ACO) เพื่อปรับปรุงพารามิเตอร์ในการควบคุมแบบ PID จำเป็นต้องมียุทธศาสตร์ความรู้พื้นฐานเกี่ยวกับระบบควบคุม การออกแบบตัวควบคุมแบบ PID รวมทั้งการค้นหาผลเฉลยแบบศึกษาสำนึก (Heuristic Search) โดยมีรายละเอียด ดังนี้

2.1 ระบบควบคุมเบื้องต้น

ในการศึกษาระบบควบคุมจำเป็นต้องอาศัยแบบจำลองทางคณิตศาสตร์เพื่อบรรยายระบบ โดยแบบจำลองที่ใช้บรรยายระบบเชิงเส้นไม่แปรตามเวลามีอยู่ 2 รูปแบบกล่าวคือ ฟังก์ชันถ่ายโอน (Transfer function) ซึ่งใช้ศึกษาระบบในโดเมนของลาปลาซ และสมการสถานะ (State equation) ซึ่งใช้ศึกษาระบบในโดเมนของเวลา

2.1.1 ฟังก์ชันถ่ายโอน (Transfer Function)

การสร้างแบบจำลองทางคณิตศาสตร์ในรูปแบบของฟังก์ชันถ่ายโอนนั้น เริ่มจากการหาสมการที่บรรยายระบบ แต่เนื่องจากข้อจำกัดของการแปลงลาปลาซ ซึ่งประยุกต์ได้กับสมการเชิงอนุพันธ์สามัญเชิงเส้นที่ตัวแปรไม่แปรตามเวลา ดังนั้นระบบที่สามารถนำมาสร้างแบบจำลองในรูปแบบของฟังก์ชันถ่ายโอนต้องเป็นระบบพลวัตเชิงเส้นไม่แปรตามเวลาเท่านั้น เมื่อได้สมการเชิงอนุพันธ์ที่บรรยายระบบดังกล่าวจึงทำการแปลงลาปลาซ โดยกำหนดเงื่อนไขเริ่มต้นเป็นศูนย์ จากนั้นให้หาอัตราส่วนระหว่างสัญญาณออกที่ต้องการทราบกับสัญญาณเข้าของระบบ โดยอัตราส่วนที่ได้เรียกว่า ฟังก์ชันถ่ายโอน (Transfer function) ซึ่งเป็นฟังก์ชันตรรกยะในตัวแปรลาปลาซโดยทั่วไปนิยมเขียนแทนฟังก์ชันถ่ายโอนของระบบด้วย $G(s)$ ซึ่งเขียนอยู่ในรูปทั่วไปได้ดังสมการที่ 2.1

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_{m-1} s + b_m}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad (2.1)$$

โดยที่ $m \leq n$

2.1.2 ปริภูมิสถานะ (State Space)

เนื่องจากการวิเคราะห์ระบบด้วยฟังก์ชันถ่ายโอนใช้ได้กับเฉพาะระบบเชิงเส้นไม่แปรตามเวลา ที่ซึ่งต้องกำหนดให้เงื่อนไขเริ่มต้นเป็นศูนย์เท่านั้น ดังนั้นหากพารามิเตอร์ของระบบมีเงื่อนไขเริ่มต้น ผลตอบสนองที่ได้จากการวิเคราะห์ด้วยฟังก์ชันถ่ายโอนย่อมเกิดค่าผิดพลาดขึ้น นอกจากนี้ระบบในปัจจุบันมีขนาดใหญ่ และบางระบบอาจเป็นระบบไม่เชิงเส้นซึ่งไม่สามารถหาฟังก์ชันถ่ายโอนได้ จากปัญหาที่กล่าวมาข้างต้น จึงมีผู้เสนอแนวคิดการวิเคราะห์ระบบในโดเมนของเวลา โดยการกำหนดตัวแปรต่างๆขึ้นมา เรียกว่า ตัวแปรสถานะ (State Variables) ซึ่งแทนด้วย $x \in \mathbb{R}^n$, แทนสัญญาณควบคุมหรือสัญญาณเข้าด้วย $u \in \mathbb{R}^m$ และแทนผลตอบสนองของระบบหรือสัญญาณออกด้วย $y \in \mathbb{R}^p$ ดังนั้นระบบที่ถูกบรรยายในปริภูมิสถานะสามารถเขียนอยู่ในรูปทั่วไปได้เป็น

$$\Sigma := \begin{cases} \dot{x}(t) = f(x; u; t) \\ y(t) = g(x; u; t) \end{cases} \quad (2.2)$$

ในกรณีระบบที่พิจารณาเป็นระบบเชิงเส้น ไม่แปรตามเวลาสมการที่ 2.2 สามารถเขียนได้ดังนี้

$$\Sigma := \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (2.3)$$

โดยที่

$A \in \mathbb{R}^{n \times n}$ เป็นเมทริกซ์ระบบ (System Matrix)

$B \in \mathbb{R}^{n \times m}$ เป็นเมทริกซ์สัญญาณเข้า (Input Matrix)

$C \in \mathbb{R}^{p \times n}$ เป็นเมทริกซ์สัญญาณออก (Output Matrix)

$D \in \mathbb{R}^{p \times m}$ เป็นเมทริกซ์ป้อนไปข้างหน้า (Feed forward Matrix)

2.1.3 ความสัมพันธ์ระหว่างปริภูมิสถานะกับฟังก์ชันถ่ายโอน

จากที่ได้กล่าวมาแล้วข้างต้นระบบพลวัตเชิงเส้นไม่แปรตามเวลาสามารถบรรยายได้ทั้งในรูปแบบของฟังก์ชันถ่ายโอน และในรูปแบบของปริภูมิสถานะ แม้ว่าการบรรยายทั้ง 2 แบบข้างต้นมีความแตกต่างกัน แต่ทั้งสองวิธีใช้บรรยายระบบเดียวกัน ดังนั้นรูปแบบทั้งสองจึงมีความสัมพันธ์กัน ดังนี้

2.1.3.1 การแปลงระบบจากปริภูมิสถานะให้อยู่ในรูปฟังก์ชันถ่ายโอน
จากสมการที่ 2.3 เมื่อทำการแปลงลาปลาซ จะได้

$$sX(s) - x(0) = AX(s) + BU(s) \quad (2.4)$$

นั่นคือ

$$X(s) = (sI - A)^{-1}BU(s) + (sI - A)^{-1}x_0 \quad (2.5)$$

และผลการแปลงลาปลาซของสมการออกในสมการที่ 2.3 จะได้เป็น

$$Y(s) = CX(s) + DU(s) \quad (2.6)$$

แทนค่า $X(s)$ ที่ได้จากสมการที่ 2.5 แทนลงในสมการข้างต้น จะได้

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) + C(sI - A)^{-1}x_0 \quad (2.7)$$

ถ้ากำหนดให้เงื่อนไขเริ่มต้น $x_0 = 0$ ดังนั้นจะได้ความสัมพันธ์ระหว่างสัญญาณเข้าและสัญญาณออกคือ

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) := G(s)U(s) \quad (2.8)$$

เพราะฉะนั้นฟังก์ชันถ่ายโอนของระบบคือ

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (2.9)$$

ตัวอย่างที่ 2.1 ให้เมทริกซ์ในปริภูมิสถานะของระบบคือ

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = [0 \quad 1], D = 0$$

จะได้ฟังก์ชันถ่ายโอนเป็น

$$G(s) = [0 \quad 1] \left(\begin{bmatrix} s & -1 \\ 1 & s+3 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{s^2 + 3s + 1}$$

ตัวอย่างที่ 2.2 ให้เมทริกซ์ในปริภูมิสถานะของระบบคือ

$$A = \begin{bmatrix} -14 & -56 & -64 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C = [0 \ 0 \ 64], D = 0$$

จะได้ฟังก์ชันถ่ายโอนเป็น

$$G(s) = [0 \ 0 \ 64] \begin{bmatrix} s+14 & 56 & 64 \\ -1 & s & 0 \\ 0 & -1 & s \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{64}{s^3 + 14s^2 + 56s + 64}$$

2.1.3.2 การแปลงระบบจากฟังก์ชันถ่ายโอนให้อยู่ในรูปของปริภูมิสถานะ

การแปลงระบบจากฟังก์ชันถ่ายโอนให้อยู่ในรูปปริภูมิสถานะนั้นมีหลายวิธีซึ่งในหัวข้อนี้ได้เสนอวิธีที่เป็นที่นิยมใช้ทั่วไปเรียกว่า รูปแบบบัญญัติความสามารถควบคุมได้ (Controllable Canonical Form: CCF) พิจารณาฟังก์ชันถ่ายโอนที่เขียนอยู่ในรูปของ

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0s^n + b_1s^{n-1} + \dots + b_{n-1}s + b_n}{s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} \quad (2.10)$$

แล้วรูปแบบบัญญัติความสามารถควบคุมได้คือ

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [b_n - a_n b_0 \quad b_{n-1} - a_{n-1} b_0 \quad \dots \quad b_1 - a_1 b_0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + [b_0] u$$

ตัวอย่างที่ 2.3 จากตัวอย่างที่ 2.2 ระบบตัวอย่างคือ $\frac{64}{s^3 + 14s^2 + 56s + 64}$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -64 & -56 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [64 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

เพราะฉะนั้นจะได้ $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -64 & -56 & -14 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $C = [64 \ 0 \ 0]$, $D = 0$

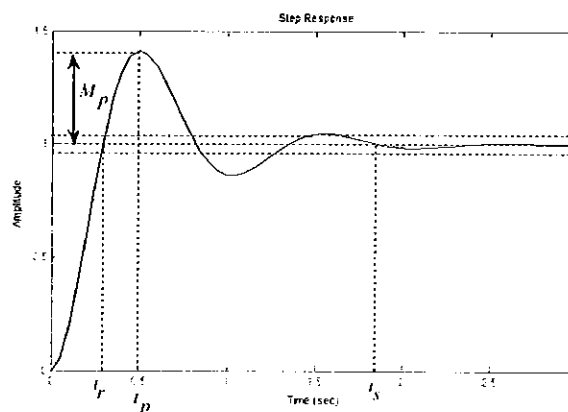
ผู้อ่านจะสังเกตได้ว่า ระบบในตัวอย่างที่ 2.2 และ 2.3 เป็นระบบเดียวกันแต่สามารถเขียนบรรยายในรูปแบบของปริภูมิสถานะแตกต่างกันได้

2.2 ผลตอบสนองเชิงเวลาของระบบอันดับสอง

ในหัวข้อนี้เป็นการศึกษาผลตอบสนองเชิงเวลาของระบบอันดับสองซึ่งประกอบไปด้วยสองส่วน คือผลตอบสนองชั่วขณะและผลตอบสนองที่สภาวะคงตัวโดยมีรายละเอียดดังต่อไปนี้

2.2.1 ผลตอบสนองชั่วขณะ

พิจารณาผลตอบสนองของระบบอันดับสอง เมื่อป้อนสัญญาณเข้าเป็นแบบฟังก์ชันขั้นบันไดหนึ่งหน่วย ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ผลการตอบสนองชั่วขณะของระบบอันดับสอง

จากผลตอบสนองที่ได้ในรูปที่ 2.1 พบว่ามีพารามิเตอร์ที่สำคัญอยู่ 4 ค่าที่ใช้ในการระบุคุณสมบัติเชิงเวลาของระบบอันดับสอง โดยพารามิเตอร์ทั้ง 4 มีดังนี้

1. เวลาขึ้น (Rise Time: t_r) คือ เวลาที่ผลตอบสนองมีค่าเท่ากับค่าสุดท้ายเป็นครั้งแรก โดยคำนวณได้จากสมการดังนี้

$$t_r = \frac{\pi - \theta}{\omega_d} = \frac{\pi - \theta}{\omega_n \sqrt{1 - \zeta^2}} \quad (2.11)$$

โดยที่ $\theta = \cos^{-1} \zeta$

2. เวลาสูงสุด (Peak Time: t_p) คือ เวลาที่ผลตอบสนองมีค่าสูงสุดเป็นครั้งแรก โดยคำนวณได้จากสูตร

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (2.12)$$

3. ค่าพุ่งสูงสุด (Maximum Overshoot: M_p) คือ อัตราส่วนระหว่างผลต่างของผลตอบสนองสูงสุดกับค่าที่สภาวะคงตัวเมื่อเทียบกับค่าที่สภาวะคงตัว โดยคำนวณได้จากสูตร

$$M_p = \frac{y(t_p) - y_{ss}}{y_{ss}} \quad (2.13)$$

ในกรณีที่ป้อนสัญญาณเข้าแบบฟังก์ชันขั้นบันไดหนึ่งหน่วยค่าพุ่งสูงสุดคำนวณได้จากสูตร

$$\%M_p = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\% \quad (2.14)$$

4. เวลาสู่สมดุล (Setting Time: t_s) คือ เวลาที่ผลตอบสนองลดลงจนเข้าสู่สภาวะสมดุลเป็นครั้งแรกซึ่งค่าเวลาสู่สมดุลนี้เป็นค่าคงตัวของเวลาที่นานที่สุด (Largest time constant) ของระบบควบคุม โดยคำนวณหาค่าเวลาสู่สมดุลได้จาก

$$t_s = \frac{\ln(x/100)}{\zeta\omega_n} \quad (2.15)$$

โดยที่ x คือค่าช่วงร้อยละที่ใช้ โดยช่วงร้อยละที่นิยมใช้ในการหาค่าเวลาสู่สมดุล คือ ร้อยละ 1, ร้อยละ 2 และร้อยละ 5

$$\text{สำหรับกรณี 1\%} \quad t_s = \frac{4.6}{\zeta\omega_n}$$

$$\text{สำหรับกรณี 2\%} \quad t_s = \frac{4}{\zeta\omega_n}$$

$$\text{สำหรับกรณี 5\%} \quad t_s = \frac{3}{\zeta\omega_n}$$

ζ คือ อัตราหน่วงของระบบ (Damping Ratio)

ω_n คือ ความเร็วเชิงมุมในการแกว่งธรรมชาติ (Undamped natural frequency)

ω_d คือ ความเร็วเชิงมุมในการแกว่งขณะระบบมีความหน่วง (Damped natural frequency)

2.2.2 ผลตอบสนองที่สภาวะคงตัว

ให้ $e(t)$ คือ สัญญาณความผิดพลาดซึ่งเป็นผลต่างระหว่างผลตอบสนอง และสัญญาณอ้างอิง นั่นคือ

$$e(t) = r(t) - y(t) \quad (2.16)$$

สัญญาณความผิดพลาดที่สภาวะคงตัว คือ ค่าลิมิตของ $e(t)$ เมื่อ t มีค่าเข้าใกล้ค่าอนันต์ นั่นคือ

$$e_{ss} = e(\infty) = \lim_{t \rightarrow \infty} e(t) \quad (2.17)$$

ตัวอย่างที่ 2.4 กำหนดให้ $G(s) = \frac{Y(s)}{R(s)} = \frac{3}{s+2}$ แล้วผลตอบสนอง $y(t)$ เมื่อสัญญาณอ้างอิง $r(t)$

แบบฟังก์ชันขั้นบันไดหนึ่งหน่วย คือ

$$y(t) = \frac{3}{2} - \frac{3}{2}e^{-2t}$$

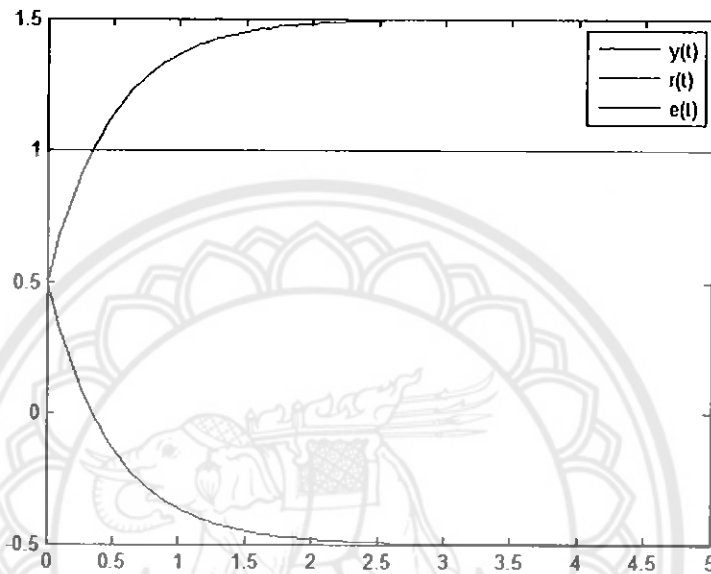
ดังนั้นสัญญาณความผิดพลาดคือ

$$e(t) = r(t) - y(t) = 1 - \frac{3}{2} + \frac{3}{2}e^{-2t} = \frac{3}{2}e^{-2t} - \frac{1}{2}$$

ซึ่งมีค่าผิดพลาดที่สภาวะคงตัวคือ

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} \left(\frac{3}{2} e^{-2t} - \frac{1}{2} \right) = -\frac{1}{2}$$

กราฟของสัญญาณอ้างอิง, ผลตอบสนองของระบบ และสัญญาณความผิดพลาดแสดงดังรูปที่ 2.2



รูปที่ 2.2 ผลตอบสนองของระบบ, สัญญาณอ้างอิง และสัญญาณความผิดพลาด

จากตัวอย่างข้างต้น พบว่าในการหาค่าผิดพลาดที่สภาวะคงตัวนั้นต้องคำนวณหาผลตอบสนองของระบบเป็นอันดับแรกบ่อยครั้งการคำนวณหาผลตอบสนองมีความยุ่งยากซับซ้อนจึงมีผู้เสนอแนวคิดในการคำนวณหาค่าผิดพลาดที่สภาวะคงตัวจากค่า $E(s)$ ซึ่งเป็นสัญญาณความผิดพลาดระหว่างผลตอบสนองและสัญญาณอ้างอิงในลาปลาซโดเมน นั่นคือ

$$E(s) = R(s) - Y(s) \quad (2.18)$$

แต่เนื่องจาก $Y(s) = G(s)R(s)$ จึงได้ว่า

$$E(s) = [1 - G(s)]R(s) \quad (2.19)$$

ในกรณีที่ $G(s)$ มีเสถียรภาพ ส่งผลให้ $[1 - G(s)]$ มีเสถียรภาพด้วยเช่นกัน ดังนั้นจึงสามารถประยุกต์ใช้ทฤษฎีบทค่าสุดท้าย (Final Value Theorem) ได้ดังนี้

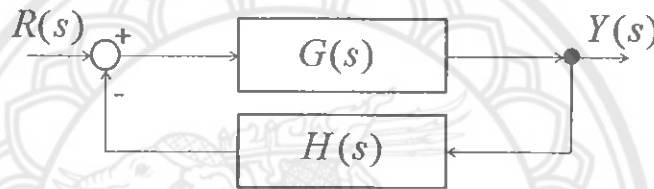
$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s[1 - G(s)]R(s) \quad (2.20)$$

พิจารณาตัวอย่างที่ 2.4 ซึ่งมี $G(s) = \frac{3}{s+2}$ ดังนั้น

$$e_{ss} = \lim_{s \rightarrow 0} s[1 - G(s)]R(s) = \lim_{s \rightarrow 0} s \left[1 - \frac{3}{s+2} \right] \frac{1}{s} = -\frac{1}{2}$$

ผู้อ่านสามารถสังเกตได้ว่า e_{ss} ที่ได้มีค่าสอดคล้องกับค่าที่ได้ในตัวอย่างที่ 2.4 แต่มีความสะดวกในการคำนวณมากกว่า

พิจารณาระบบวงปิดดังแสดงในรูปที่ 2.3



รูปที่ 2.3 แผนภาพกล่องของระบบปิด

จากรูปที่ 2.3 พบว่า

$$Y(s) = \frac{G(s)}{1 + G(s)H(s)} R(s) \quad (2.21)$$

ให้ $E(s) = R(s) - Y(s)$ จะได้ว่า

$$E(s) = R(s) - \frac{G(s)}{1 + G(s)H(s)} R(s) = \frac{1 + G(s)[H(s) - 1]}{1 + G(s)H(s)} R(s) \quad (2.22)$$

ภายใต้สมมติฐานที่ว่าระบบป้อนกลับมีเสถียรภาพจึงสามารถนำทฤษฎีบทค่าสุดท้ายมาประยุกต์ได้ดังนี้

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s \left(\frac{1 + G(s)[H(s) - 1]}{1 + G(s)H(s)} \right) R(s) \quad (2.23)$$

ในกรณีของระบบที่มีการป้อนกลับแบบหนึ่ง (Unity feedback) จากสมการที่ 2.23 จึงลดรูปได้เป็น

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s \left(\frac{1}{1+G(s)} \right) R(s) \quad (2.24)$$

สำหรับสูตรการหาค่าความผิดพลาดที่สภาวะคงตัวของระบบปิดที่มีการป้อนกลับแบบหนึ่งโดยใช้สัญญาณอ้างอิงที่นิยมมี 3 แบบดังนี้

1. ค่าความผิดพลาดที่ได้จากสัญญาณอ้างอิงแบบขั้นบันไดหนึ่งหน่วย

$$e_{ss} = \lim_{s \rightarrow 0} \frac{s}{1+G(s)} \times \frac{1}{s} = \lim_{s \rightarrow 0} \frac{1}{1+G(s)} \quad (2.25)$$

2. ค่าความผิดพลาดที่ได้จากสัญญาณอ้างอิงแบบทะยานหนึ่งหน่วย

$$e_{ss} = \lim_{s \rightarrow 0} \frac{s}{1+G(s)} \times \frac{1}{s^2} = \lim_{s \rightarrow 0} \frac{1}{s+sG(s)} = \lim_{s \rightarrow 0} \frac{1}{sG(s)} \quad (2.26)$$

3. ค่าความผิดพลาดที่ได้จากสัญญาณอ้างอิงแบบพาราโบลา

$$e_{ss} = \lim_{s \rightarrow 0} \frac{s}{1+G(s)} \times \frac{1}{s^3} = \lim_{s \rightarrow 0} \frac{1}{s^2+s^2G(s)} = \lim_{s \rightarrow 0} \frac{1}{s^2G(s)} \quad (2.27)$$

ตัวอย่างที่ 2.5 กำหนดให้ $G(s) = \frac{3}{s+2}$ แล้วค่าความผิดพลาดที่สภาวะคงตัวเมื่อนำ $G(s)$ มาป้อนกลับแบบหนึ่งมีค่าดังนี้

1. เมื่อสัญญาณอ้างอิงเป็นแบบฟังก์ชันขั้นบันได

$$e_{ss} = \lim_{s \rightarrow 0} \frac{1}{1 + \left(\frac{3}{s+2} \right)} = \lim_{s \rightarrow 0} \frac{s+2}{s+5} = \frac{2}{5} = 0.4$$

2. เมื่อสัญญาณอ้างอิงเป็นแบบฟังก์ชันทะยานหนึ่งหน่วย

$$e_{ss} = \lim_{s \rightarrow 0} \frac{1}{s \left(\frac{3}{s+2} \right)} = \lim_{s \rightarrow 0} \frac{s+2}{3s} = \infty$$

3. เมื่อสัญญาณอ้างอิงเป็นแบบฟังก์ชันพาราโบล่า

$$e_{ss} = \lim_{s \rightarrow 0} \frac{1}{s^2 \left(\frac{3}{s+2} \right)} = \lim_{s \rightarrow 0} \frac{s+2}{3s^2} = \infty$$

2.3 การออกแบบตัวควบคุมแบบ PID

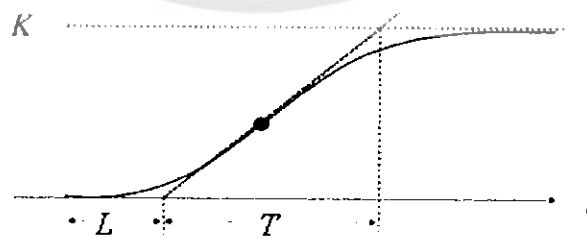
ในการออกแบบระบบควบคุม PID มีอยู่หลายวิธี แต่วิธีการที่ได้รับความนิยมมี 3 วิธี คือ วิธีการซีกเลอร์-นิโคลส์ (Ziegler – Nichols: ZN) วิธีของเชน – ฮรอน – เรสวิก (Chien – Hrones – Reswick: CHR) และวิธีของหวาง – จวง – ฉาน (The Wang – Juang – Chan: WJC) ซึ่งทั้งสามวิธีมีรายละเอียด ดังนี้

2.3.1 วิธีการของซีกเลอร์-นิโคลส์ (Ziegler – Nichols: ZN)

วิธีการนี้ถูกนำเสนอโดย John G. Ziegler และ Nathaniel B. Nichols ในปี 1940 เป็นวิธีการที่ใช้สำหรับปรับปรุงหรือพัฒนาตัวควบคุมเพื่อให้ระบบควบคุมมีผลตอบสนองที่ดีไม่เกิดค่าพุ่งสูงสุดมากเกินไป และตอบสนองเร็ว โดยซีกเลอร์-นิโคลส์ได้พัฒนาทฤษฎีของเขาจากการทดลองและวิเคราะห์ผลโดยเปรียบเทียบกับกระบวนการต่างๆ ในอุตสาหกรรม

2.3.1.1 วิธีการที่หนึ่ง (First Method)

ในวิธีการนี้ใช้ได้กับระบบที่ไม่มีตัวปริพันธ์ และผลตอบสนองของระบบต้องอยู่ในรูปของตัวเอสเท่านั้น โดยทำการป้อนสัญญาณเข้าแบบสัญญาณขั้นบันไดหนึ่งหน่วย แล้ววัดผลตอบสนองที่ได้ ตัวอย่างของผลตอบสนองแสดงในรูปที่ 2.4



รูปที่ 2.4 ผลตอบสนองรูปตัวเอส

จากผลตอบสนองในรูปที่ 2.4 นั้นมีพารามิเตอร์ที่สำคัญอยู่ 3 ตัวคือ K , L และ T ถูกกำหนดโดยลากเส้นสัมผัสกับผลตอบสนองที่จุดเปลี่ยนเว้าดังรูปที่ 2.4 ดังนั้นระบบ $G(s)$ สามารถประมาณได้ด้วยระบบอันดับหนึ่งร่วมกับ transport lag ดังนี้

$$G(s) = \frac{Y(s)}{U(s)} \cong \frac{Ke^{-Ls}}{Ts+1}$$

จากพารามิเตอร์ L และ T ที่ได้จากผลตอบสนอง ชิกเลอร์และนิโคสได้
แนะนำวิธีการหาค่าพารามิเตอร์ K_p , T_i และ T_d ตามตารางที่ 2.1

ตารางที่ 2.1 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่หนึ่ง)

ชนิดของ Controller	K_p	T_i	T_d
P – Controller	$\frac{T}{L}$	∞	0
PI – Controller	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID – Controller	$1.2\frac{T}{L}$	$2L$	$0.5L$

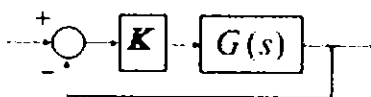
โดยที่ฟังก์ชันถ่ายโอนของตัวควบคุม PID คือ

$$G_c = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s \quad (2.28)$$

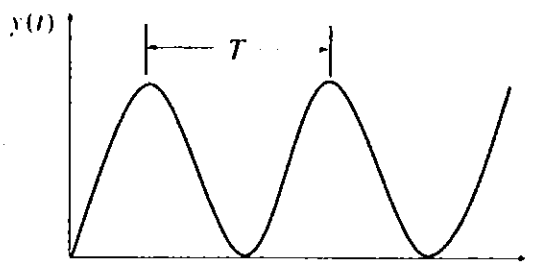
เมื่อ $K_i = \frac{K_p}{T_i}$ และ $K_d = K_p T_d$

2.3.1.2 วิธีการที่สอง (Second Method)

วิธีการนี้ใช้ได้กับระบบวงปิดดังรูปที่ 2.5 โดยทำการปรับค่า K จนถึงค่าวิกฤต K^* ซึ่งเป็นค่าที่ทำให้ผลตอบสนองต่อสัญญาณเข้าแบบฟังก์ชันขั้นบันไดหนึ่งหน่วยเกิดการแกว่งกวัด (Oscillation) จากนั้นนำผลตอบสนองที่ได้มาวัดคาบ T ดังแสดงในรูปที่ 2.6



รูปที่ 2.5 แผนภาพกล่องของระบบวงปิดที่มีตัวควบคุม



รูปที่ 2.6 ผลตอบสนองที่เกิดจากปรับค่า K จนเกิดการแกว่งกวัด

นำค่า K^* และ T มาคำนวณหาค่าพารามิเตอร์ K_p , T_i และ T_d ตามตารางที่ 2.2

ตารางที่ 2.2 การหาค่า K_p , T_i และ T_d ด้วยวิธี ZN (วิธีการที่สอง)

ชนิดของ Controller	K_p	T_i	T_d
P – Controller	$0.5K^*$	∞	0
PI – Controller	$0.45K^*$	$\frac{1}{1.2}T$	0
PID – Controller	$0.6K^*$	$0.5T$	$0.125T$

2.3.2 วิธีการของเซน-ฮรอน-เรสวิก (Chien-Hrones-Reswick: CHR)

วิธีการนี้เป็นวิธีการที่พัฒนามาจากวิธีการ ZN วิธีที่หนึ่งโดยวิธีการ CHR จะเลือกชนิดของการควบคุมตามพารามิเตอร์ R ดังตารางที่ 2.3

ตารางที่ 2.3 การหาค่า R

Controller	$R = T/L$
P – control	$R > 10$
PI – Control	$7.5 < R < 10$
PID – Control	$3 < R < 7.5$
Other methods	$R < 3$

โดยที่พารามิเตอร์ L และ T ได้มาจากผลตอบสนองดังในรูปที่ 2.4

การออกแบบตัวควบคุมแบบ PID ด้วยวิธีการของ CHR ถูกแบ่งออกเป็นสองประเภท โดยขึ้นอยู่กับค่าพุ่งสูงสุดของผลตอบสนองกล่าวคือ ผลตอบสนองที่มีค่าพุ่งสูงสุดน้อยที่สุด (Least Overshoot) และผลตอบสนองที่มีค่าพุ่งสูงสุดร้อยละ 20 (20 % Overshoot) ดังตารางที่ 2.4

ตารางที่ 2.4 การหาค่า K_p , K_i และ K_d ด้วยวิธี CHR

Controller	Least Overshoot	20 % Overshoot
P – control	$K_p = 0.3R/K$	$K_p = 0.7R/K$
PI – Control	$K_p = 0.35R/K$ $K_i = 0.29R/KT$	$K_p = 0.6R/K$ $K_i = 0.6R/KT$
PID – Control	$K_p = 0.6R/K$ $K_i = 0.6R/KT$ $K_d = 0.3RL/K$	$K_p = 0.95R/K$ $K_i = 0.7R/KT$ $K_d = 0.45RL/K$

โดยที่ฟังก์ชันถ่ายโอนของตัวควบคุม PID คือ

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (2.29)$$

2.3.3 วิธีการของหวาง – จวง – ฉาน (Wang – Juang – Chan: WJC)

ขั้นตอนวิธีการปรับค่าพารามิเตอร์ของตัวควบคุม PID ที่นำเสนอโดยหวาง จวง และ ฉาน เป็นวิธีที่ง่ายและมีประสิทธิภาพวิธีการนี้มีลักษณะคล้ายกับวิธีของ ZN และ CHR โดยมีการวัดค่าพารามิเตอร์ K , L และ T จากผลตอบสนองดังแสดงในรูปที่ 2.4 จากนั้นจึงนำพารามิเตอร์ที่ได้ทั้งคู่มาคำนวณหาค่า K_p , T_i และ T_d ตามตารางที่ 2.5

ตารางที่ 2.5 ตารางหาค่า K_p , T_i และ T_d ด้วยวิธี WJC

Control	Formula
K_p	$\frac{(0.7303 + 0.5307T/L)(T + 0.5L)}{K(T + L)}$
T_i	$T + 0.5L$
T_d	$\frac{0.5LT}{T + 0.5L}$

ซึ่งจะได้ฟังก์ชันถ่ายโอนของตัวควบคุมแบบ PID ดังนี้

$$G_c = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = K_p + \frac{K_i}{s} + K_d s \quad (2.30)$$

เมื่อ $K_i = \frac{K_p}{T_i}$ และ $K_d = K_p T_d$

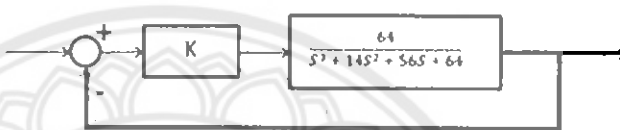
ตัวอย่างที่ 2.6 แสดงวิธีการหาค่าพารามิเตอร์ K_p , K_i และ K_d ด้วยวิธี ZN, CHR และ WJC ของระบบต่อไปนี้

$$G(s) = \frac{64}{s^3 + 14s^2 + 56s + 64}$$

วิธีการของ ZN

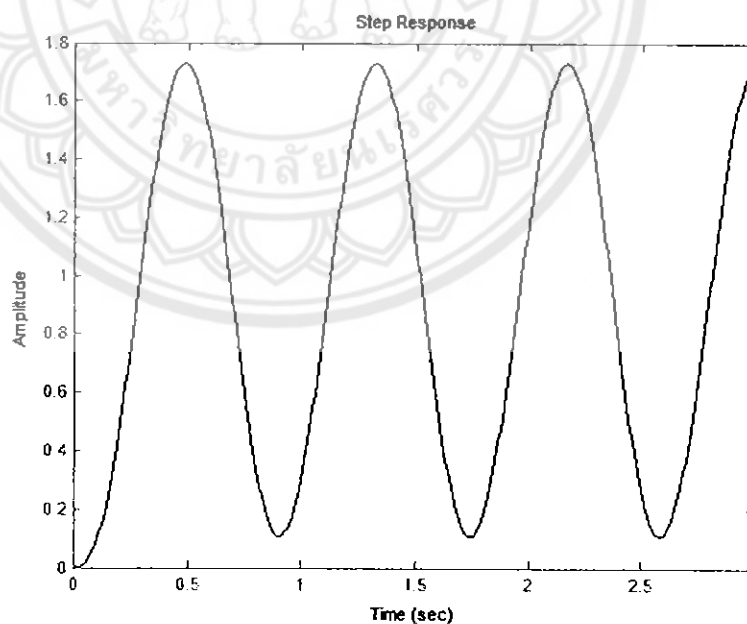
ขั้นตอนการออกแบบตัวควบคุม PID ซึ่งคำนวณด้วยวิธี ZN โดยมีขั้นตอนดังนี้

1. ทำการป้อนอัตราขยาย K เข้ากับระบบและทำการป้อนกลับด้วยหนึ่งดังรูปที่ 2.7



รูปที่ 2.7 แผนภาพกล่องแสดงการป้อนกลับของฟังก์ชันถ่ายโอน

2. ทำการปรับอัตราขยาย K จนผลตอบสนองของระบบเกิดการแกว่งกวัด ซึ่งได้ค่าอัตราขยาย K^* คือ 11.25 และคาบ T มีค่าประมาณ 0.9 วินาที ดังรูปที่ 2.8



รูปที่ 2.8 ผลตอบสนองของระบบตัวอย่าง $K^* = 11.25$

3. กำหนดค่า K_p , T_i และ T_d จะได้พารามิเตอร์ตามตารางที่ 2.6

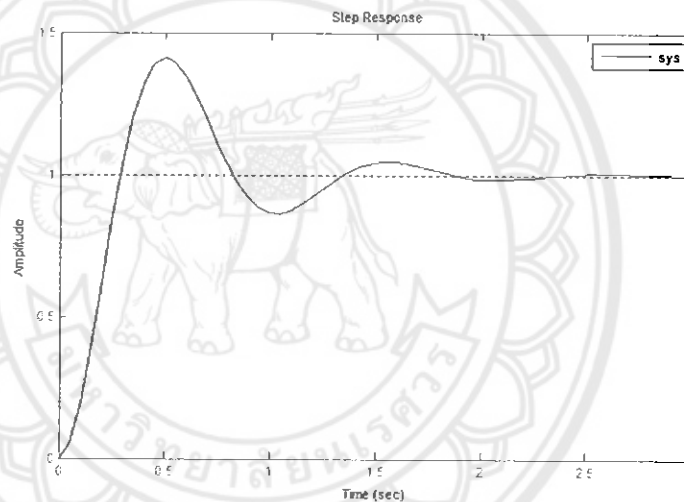
ตารางที่ 2.6 การคำนวณค่าพารามิเตอร์แบบ PID ด้วยขั้นตอนวิธี ZN

Controller	K_p	T_i	T_d
PID – control	$= 0.6K^*$	$= 0.5T$	$= 0.125T$
	$= 0.6 \times 11.25$	$= 0.5 \times 0.9$	$= 0.125 \times 0.9$
	$= 6.75$	$= 0.45$	$= 0.1125$

นำค่า K_p , T_i และ T_d แทนลงสมการที่ 2.28 จะได้ฟังก์ชันถ่ายโอนของตัวควบคุม คือ

$$G_c = 6.75 \left(1 + \frac{1}{0.45s} + 0.1125s \right) = \frac{0.76s^2 + 6.75s + 15}{s}$$

ผลตอบสนองของระบบป้อนกลับที่ใช้ตัวควบคุมแบบ PID ซึ่งคำนวณด้วยวิธี ZN แสดงในรูปที่ 2.9

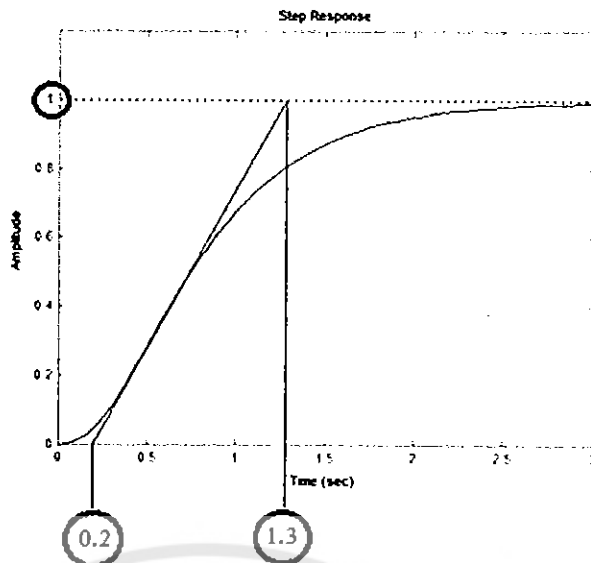


รูปที่ 2.9 ผลตอบสนองของระบบควบคุมแบบ ZN

วิธีการของ CHR

ขั้นตอนการออกแบบตัวควบคุมแบบ PID ด้วยวิธีการ CHR มีขั้นตอนการทำงานดังนี้

1. ป้อนสัญญาณเข้าแบบขั้นบันไดหนึ่งหน่วยให้กับระบบจากนั้นทำการบันทึกผลตอบสนองและลากเส้นตรงสัมผัสจุดเปลี่ยนเว้าของผลตอบสนองดังรูปที่ 2.10



รูปที่ 2.10 การหาค่า K , L และ T จากผลตอบสนองของระบบ

- ทำการบันทึกค่า K , L และ T ที่ได้จากรูปที่ 2.10 ลงในตารางที่ 2.7

ตารางที่ 2.7 พารามิเตอร์ที่ใช้ในการคำนวณด้วยวิธี CHR

พารามิเตอร์	K	T	L
ผลลัพธ์	1	1.1	0.2

- คำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ซึ่งแสดงไว้ในตารางที่ 2.8

ตารางที่ 2.8 การคำนวณค่าพารามิเตอร์แบบ PID ด้วยขั้นตอนวิธี CHR

Controller	20 % Overshoot	Least Overshoot
K_p	$0.95R / K = 0.95 \times 5.5 / 1 = 5.22$	$0.6R / K = 0.6 \times 5.5 / 1 = 3.3$
K_i	$0.7R / KT = 0.7 \times 5.5 / (1 \times 1.1) = 3.52$	$0.6R / KT = 0.6 \times 5.5 / (1 \times 1.1) = 3$
K_d	$0.45RL / K = 0.45 \times 5.5 \times 0.2 / 1 = 0.49$	$0.3RL / K = 0.3 \times 5.5 \times 0.2 = 0.33$

*หมายเหตุ ค่า R มีค่าเท่ากับ $\frac{T}{L} = \frac{1.1}{0.2} = 5.5$

นำค่า K_p , K_i และ K_d แทนในสมการที่ 2.29 จะได้ฟังก์ชันถ่ายโอนของตัวควบคุม

นั่นคือ

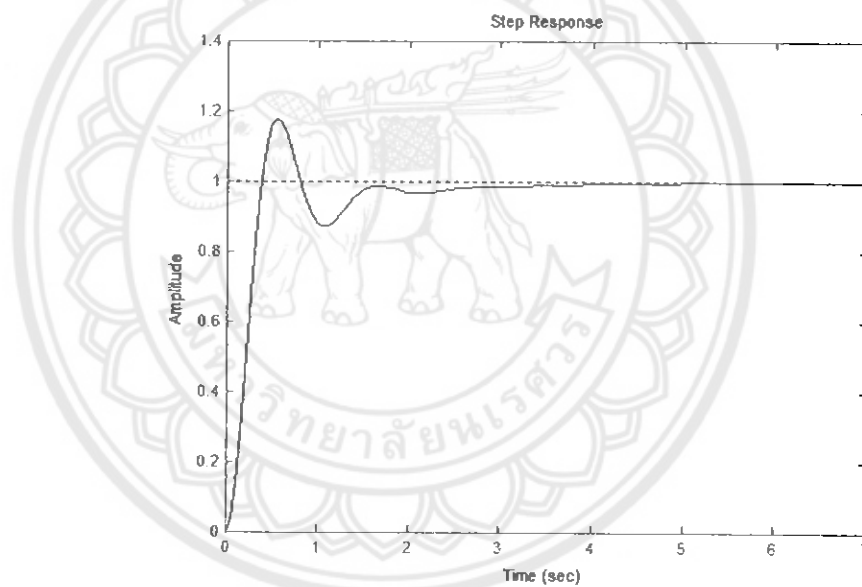
ฟังก์ชันถ่ายโอนของระบบควบคุมแบบ CHR ที่มีค่าพุงร้อยละ 20

$$G_{co} = \frac{0.49s^2 + 5.22s + 3.52}{s}$$

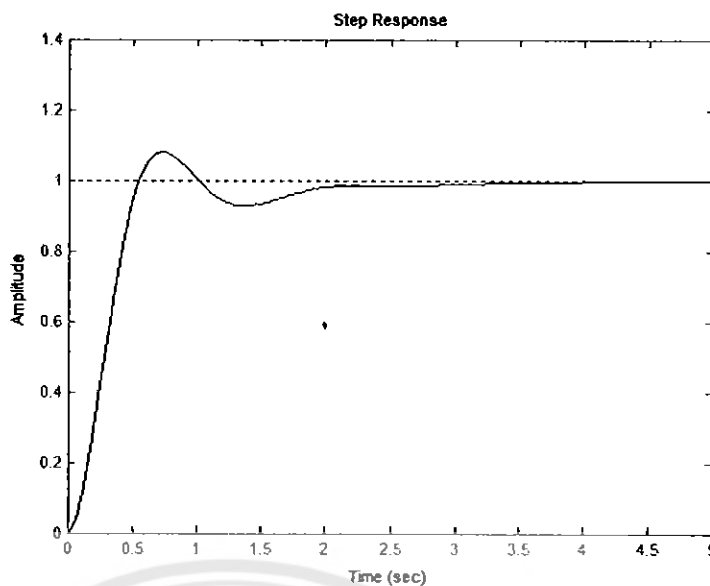
และฟังก์ชันถ่ายโอนของระบบควบคุมแบบ CHR ที่มีค่าพุงน้อยที่สุด

$$G_d = \frac{0.33s^2 + 3.3s + 3}{s}$$

ดังนั้นผลตอบสนองของระบบป้อนกลับที่มีตัวควบคุมแบบ PID ด้วยวิธีการของ CHR ทั้งสองแบบ แสดงในรูปที่ 2.11 และ 2.12 ตามลำดับ



รูปที่ 2.11 ผลตอบสนองของระบบควบคุมแบบ CHR ด้วยวิธี Overshoot ร้อยละ 20



รูปที่ 2.12 ผลตอบสนองของระบบควบคุมแบบ CHR ด้วยวิธี Least Overshoot

วิธีการของ WJC

ขั้นตอนวิธีการปรับค่าพารามิเตอร์ของตัวควบคุมแบบ PID ด้วยวิธี WJC โดยมีขั้นตอนดังนี้

1. ในการหาค่าพารามิเตอร์ K , T และ L ทำเช่นเดียวกับวิธีของ CHR ซึ่งได้ดังตารางที่ 2.9

ตารางที่ 2.9 พารามิเตอร์ที่ใช้ในการคำนวณด้วยขั้นตอนวิธี WJC

พารามิเตอร์	K	T	L
ผลลัพธ์	1	1.1	0.2

2. คำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID โดยใช้สูตรจากตารางที่ 2.5 ซึ่งจะได้พารามิเตอร์แสดงไว้ในตารางที่ 2.10

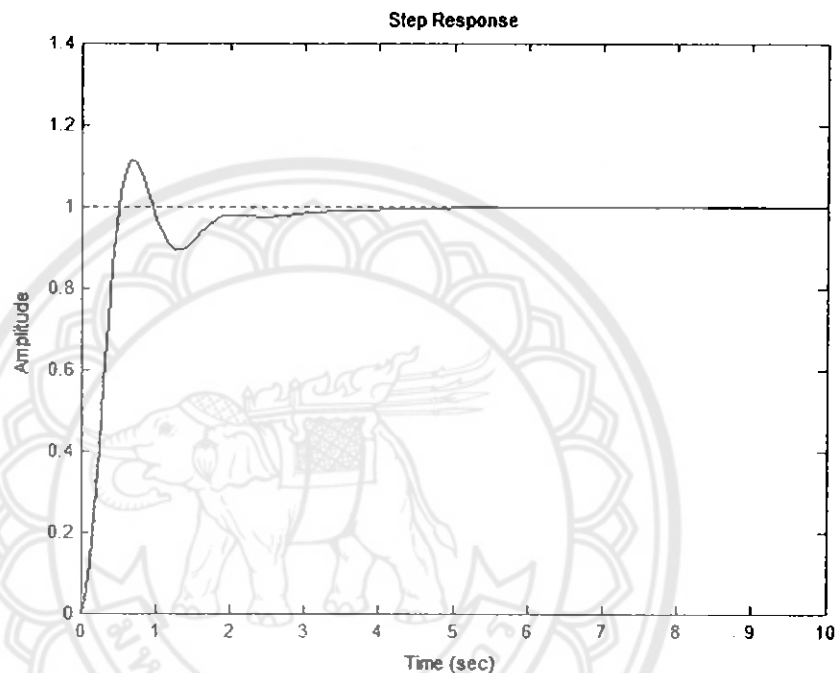
ตารางที่ 2.10 การหาค่า K_p , T_i และ T_d ด้วยขั้นตอนวิธี WJC

Control	Results
K_p	$= \frac{(0.7303 + (0.5307 \times 1.1 / 0.2))(1.1 + (0.5 \times 0.2))}{1 \times (1.1 + 0.2)}$ $= 3.36845$
T_i	$= 1.1 + (0.5 \times 0.2) = 1.2$
T_d	$= \frac{0.5 \times 0.2 \times 1.1}{1.1 + (0.5 \times 0.2)} = 0.09167$

นำค่า K_p , T_i และ T_d ไปแทนในสมการที่ 2.30 จะได้ฟังก์ชันถ่ายโอนของตัวควบคุม
คือ

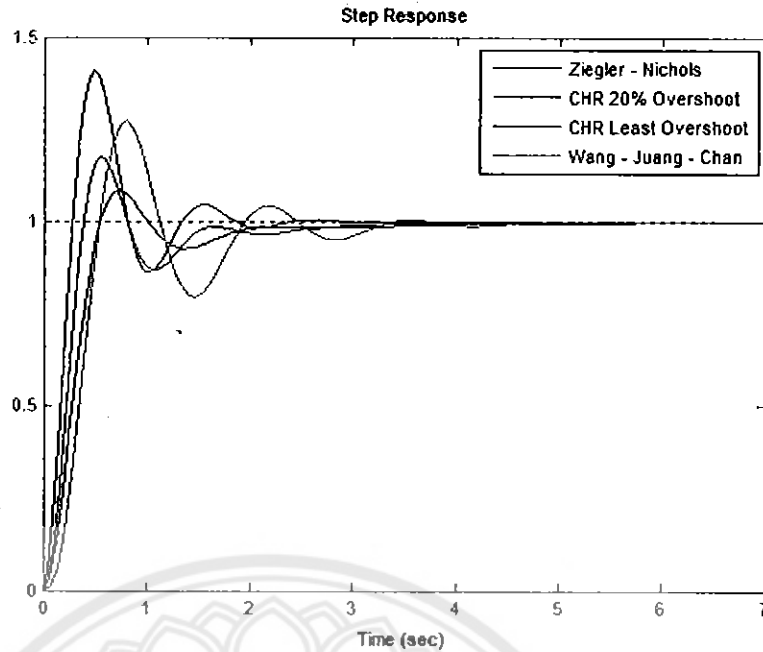
$$G_c = 3.36845 \left(1 + \frac{1}{1.2s} + 0.09167s \right) = \frac{0.3088s^2 + 3.3685s + 2.8070}{s}$$

ผลตอบสนองของระบบป้อนกลับที่ใช้ตัวควบคุมแบบ PID ซึ่งคำนวณด้วยวิธี WJC แสดงในรูปที่ 2.13



รูปที่ 2.13 ผลตอบสนองที่มีระบบควบคุมแบบ WJC

เมื่อนำผลตอบสนองของระบบที่มีการออกแบบตัวควบคุมแบบ PID โดยใช้วิธีการต่างๆ ดังที่กล่าวมาแล้วข้างต้นมาเปรียบเทียบกันพบว่าผลตอบสนองของระบบป้อนกลับที่ใช้ตัวควบคุมแบบ PID ซึ่งคำนวณด้วยวิธี ZN นั้นจะมีค่าพุ่งสูงสุดค่อนข้างสูง แต่มีเวลาขึ้นเร็วที่สุด ในทางกลับกันตัวควบคุมที่คำนวณด้วยวิธี CHR แบบ Least Overshoot มีค่าพุ่งสูงสุดน้อยที่สุดประมาณร้อยละ 5 – 20 แต่มีค่าเวลาขึ้นช้ากว่า ZN และวิธีสุดท้าย คือ วิธีของ WJC จะมีค่าพุ่งสูงสุดอยู่กลางประมาณร้อยละ 35 ซึ่งผลตอบสนองแสดงการเปรียบเทียบไว้ในรูปที่ 2.14



รูปที่ 2.14 การเปรียบเทียบผลตอบสนองของระบบที่มีตัวควบคุม PID

2.4 การค้นหาผลเฉลยแบบศึกษาสำนึก (Heuristic Search)

การค้นหาผลเฉลยแบบศึกษาสำนึก คือ การค้นหาผลเฉลยบางตัว จากนั้นจึงนำข้อมูลที่ได้จากผลเฉลยเหล่านั้นมาปรับปรุงเพื่อใช้ค้นหาผลเฉลยที่เหลือนั้น จนในที่สุดทำให้ได้ผลเฉลยที่ใกล้เคียงกับผลเฉลยที่เหมาะสมที่สุดนั่นเอง ซึ่งขั้นตอนวิธีในการค้นหาผลเฉลยแบบศึกษาสำนึกแบบต่างๆ ได้แก่ การค้นหาแบบทาบู (Taboo Search: TS), การค้นหาแบบการซิมูเลเทดแอนนีลลิ่ง (Simulated Annealing: SA), การค้นหาแบบฮาร์โมนี (Harmony Search), การค้นหาแบบการจับกลุ่มของอนุภาค (Particle Swarm Optimization: PSO), การค้นหาแบบกลไกคัดสรรพันธุ์ที่ดีตามธรรมชาติ (Genetic Algorithms: GA), การค้นหาแบบการสลัดของกบ (Shuffled Frog Leaping Algorithm: SFL) และ การค้นหาแบบอาณานิคมมด (Ant Colony Optimization: ACO) เป็นต้น

2.4.1 การค้นหาแบบทาบู (Taboo Search: TS)

เป็นวิธีการแก้ปัญหาอย่างชาญฉลาดถูกคิดค้นโดย Glover ในปี 1986 และจัดอยู่ในกลุ่มของวิธีการแบบ Metaheuristic ซึ่งใช้หลักการค้นหาแบบเฉพาะที่ (Local Search) ในการแก้ปัญหา เนื่องจากคำว่า Tabu หรือ Taboo นั้นหมายถึงการห้าม ดังนั้นวิธีการค้นหาแบบทาบู จึงหมายถึงวิธีการที่มีข้อห้ามมิให้มีการค้นหาคำตอบในบางขอบเขต การห้ามดังกล่าวเป็นการห้ามที่จะช่วยป้องกันไม่ให้ออกไปค้นหาคำตอบหรือผลเฉลยเดิมอีก ซึ่งเป็นคำตอบหรือผลเฉลยที่ไม่ใช่คำตอบที่ดีที่สุด (Local Optimum) การค้นหาแบบทาบูมีข้อดี คือ เกิดความเป็นไปได้ในการค้นหาคำตอบที่มากยิ่งขึ้นและยังลดเวลาในการค้นหาอีกด้วย

159 3 405 8

ร/ร.

๘๖๖๖๙ ๐

๒๕๖๓

ขั้นตอนการทำงานของวิธีการค้นหาแบบทวนในการแก้ปัญหาที่มีดังนี้

1. สร้างผลเฉลยเริ่มต้น ด้วยการสุ่มตัวอย่างขึ้นมาหนึ่งผลเฉลยก่อน ซึ่งในแต่ละรอบการทำซ้ำจะสร้างผลเฉลยข้างเคียง (Neighbor) ขึ้นมาจากผลเฉลยเริ่มต้น
2. คำนวณหาค่าฟังก์ชันเป้าหมาย (Fitness Function) จากผลเฉลยที่ได้ ถ้าผลเฉลยข้างเคียงมีค่าฟังก์ชันเป้าหมายที่ดีกว่า ให้ทำการเปลี่ยนผลเฉลยข้างเคียงมาเป็นผลเฉลยแทน
3. เก็บผลเฉลยที่เลือกไปแล้วลงในรายการต้องห้าม (Taboo List) เพื่อป้องกันไม่ให้เลือกผลเฉลยนั้นซ้ำในรอบถัดไป อย่างไรก็ตามผลเฉลยที่เลือกไปแล้วนั้นสามารถนำกลับมาใช้ได้อีกก็ต่อเมื่อครบกำหนดรอบในการทำซ้ำ (Iteration) ของรายการต้องห้าม
4. ทำการวนซ้ำจนพบผลเฉลยที่เหมาะสมที่สุด หรือเมื่อมีจำนวนรอบของการทำซ้ำสูงสุดถือว่าจบการทำงาน

2.4.2 การค้นหาแบบการซิมูเลเทดแอนนีลิ่ง (Simulated Annealing: SA)

การค้นหาแบบการซิมูเลเทดแอนนีลิ่ง ถูกพัฒนาขึ้นโดย Kirkpatrick และคณะ ในปี 1983 และ Kemy ในปี 1985 ซึ่งใช้หลักการค้นหาผลเฉลยแบบเฉพาะที่ วิธีการค้นหาแบบ SA ได้รับแนวคิดมาจากกระบวนการอบเหนียวของ โลหะซึ่งเป็นการลดอุณหภูมิระหว่างการหลอมละลาย โดยจะให้ความร้อนและมีการลดอุณหภูมิลงอย่างช้าๆ จนกระทั่ง โลหะอยู่ในสภาวะเหมาะสมที่สุดคือ ได้โลหะที่เหนียวไม่เปราะ ซึ่งหลักการดังกล่าวถูกนำมาประยุกต์ใช้ในการแก้ปัญหาแบบการหาค่าเหมาะสมที่สุดในเชิงการจัด (Combinatorial Optimization Problem) อย่างกว้างขวาง โดยมีขั้นตอนการทำงานดังนี้

1. สร้างผลเฉลยเริ่มต้นด้วยการสุ่มตัวอย่างขึ้นมาหนึ่งผลเฉลย
2. กำหนดค่าอุณหภูมิเริ่มต้น โดยกำหนดให้อุณหภูมิมีค่ามากกว่าศูนย์ ($T > 0$)
3. ในแต่ละรอบของการทำงานจะทำการสร้างผลเฉลยข้างเคียงขึ้นมาจากผลเฉลยเริ่มต้น โดยที่ผลต่างของผลเฉลยข้างเคียงกับผลเฉลยปัจจุบันมีค่าน้อยกว่าหรือเท่ากับศูนย์จะให้ผลเฉลยข้างเคียงนั้นมาเป็นผลเฉลยปัจจุบัน แต่ถ้าผลต่างของผลเฉลยมีค่ามากกว่าศูนย์จะให้ความน่าจะเป็นในการสร้างผลเฉลย
4. กำหนดให้ T เป็นอุณหภูมิที่เปลี่ยนแปลงไปในแต่ละรอบของการทำซ้ำตามอัตราการลดลงของค่า R สามารถคำนวณได้จาก $T_i = RT_{i-1}$
5. ทำการวนซ้ำจนพบผลเฉลยที่เหมาะสมที่สุด หรือเมื่อมีจำนวนรอบของการทำซ้ำครบตามจำนวนที่กำหนดไว้แล้วจึงจบการทำงาน

2.4.3 การค้นหาแบบฮาร์โมนี (Harmony Search)

การค้นหาแบบฮาร์โมนีนี้คิดค้นโดย Dr. Zong Woo Geem ในปี 2001 โดยได้แนวคิดจากการแต่งเพลงของนักดนตรีในการหาตัวโน้ตที่เหมาะสมสำหรับเครื่องดนตรีแต่ละชนิดในการเล่นกันเป็นวงดนตรี โดยจุดมุ่งหมายของการค้นหาแบบฮาร์โมนีนี้ คือ ต้องการสร้างขั้นตอนวิธีที่ดีกว่าและเร็วกว่าขั้นตอนการค้นหาแบบกลไกคัดสรรพันธุ์ที่ดีตามธรรมชาติ โดยการค้นหาแบบฮาร์โมนีนี้มีขั้นตอนการทำงานดังนี้

1. สร้างความจำของฮาร์โมนี (Harmony memory) สำหรับจำตัวโน้ตดนตรีที่เคยทำการเล่นมาแล้ว โดยการกำหนดขนาดของฮาร์โมนี (Harmony Size) ในตอนเริ่มต้น ให้มีจำนวนระหว่าง 10 - 100
2. พัฒนาฮาร์โมนีตัวใหม่ในสัดส่วนดังนี้ ร้อยละ 95 ถูกพัฒนาจากความจำของฮาร์โมนี และร้อยละ 5 เกิดจากการพัฒนาด้วยตนเอง จากนั้นกำหนดอัตราการปรับของระดับเสียง (Pitch Adjustment Rate) ซึ่งถูกกำหนดจากความจำของฮาร์โมนี ในอัตราร้อยละ 70 และพัฒนาตนเองในอัตราร้อยละ 30
3. ทำการเปรียบเทียบ ถ้าฮาร์โมนีตัวใหม่ (New Harmony) ดีกว่าตัวที่แย่ที่สุดของความจำฮาร์โมนี แล้วนำฮาร์โมนีตัวใหม่แทนที่ฮาร์โมนีตัวที่แย่ที่สุดในความจำของฮาร์โมนี
4. เมื่อทำงานครบกำหนดตามจำนวนขนาดของฮาร์โมนีและจำนวนรอบแล้ว จะหยุดการทำงานและเลือกตัวที่ดีที่สุดของฮาร์โมนีเป็นคำตอบ

2.4.4 การค้นหาแบบการจับกลุ่มของอนุภาค (Particle Swarm Optimization: PSO)

PSO ถูกค้นพบและพัฒนาโดย Eberhart และ Kennedy ในปี 1995 ซึ่งประกอบด้วยกลุ่มประชากร ซึ่งเรียกว่า Particle โดยแต่ละ Particle จะสำรวจค่าผลเฉลยที่เป็นไปได้ของปัญหาโดยมีการค้นหาผลเฉลยเป็นกลุ่ม ซึ่งเกิดจากการเรียนรู้ซึ่งกันและกันภายในกลุ่ม กล่าวคือภายในแต่ละ Particle จะประกอบไปด้วย ตำแหน่งของตนเอง (Position) และอัตราเร็วในการเคลื่อนที่ (Velocity) โดยในแต่ละรอบของการค้นหาผลเฉลยจะเกิดการปรับปรุงอัตราความเร็ว (Update Velocity) เพื่อนำไปปรับปรุงค่าตำแหน่ง (Update Position) ให้ได้ตำแหน่งที่ดีที่สุดหรือค่าผลเฉลยที่ดีที่สุดนั่นเอง คุณสมบัติของการเคลื่อนที่ (Swarm) เป็นกลุ่มต้องประกอบไปด้วยคุณสมบัติพื้นฐานที่สำคัญ 5 ประการดังนี้

1. Proximity Principle คือ หลักการการเคลื่อนที่ให้สัมพันธ์กันในทุกช่วงเวลา มีการรักษาระยะห่างระหว่างกันในการบินเป็นกลุ่ม ซึ่งเป็นการป้องกันการชนกันของแต่ละ Particle ที่เป็นสมาชิกในกลุ่ม

2. Quality Principle คือ หลักการการปรับตัว หรือการตอบสนองต่อปัจจัยต่างๆ จากสิ่งแวดล้อมที่เปลี่ยนแปลงตลอดเวลาของแต่ละ Particle เพื่อให้สามารถมีชีวิตอยู่ได้ในทุกสภาวะ
 3. Diverse Response คือ ความหลากหลายของค่าคำตอบที่เหมาะสม ซึ่งเป็นการไม่ยึดติดกับค่าคำตอบใดคำตอบหนึ่ง โดยคำตอบที่ได้สามารถปรับปรุงและเปลี่ยนแปลงไปตามปัจจัยที่เหมาะสม ณ เวลาหนึ่ง
 4. Stability Principle คือ หลักการมีเสถียรภาพทางด้านพฤติกรรมของแต่ละ Particle ซึ่งพฤติกรรมของสมาชิกในสังคมจะไม่เปลี่ยนแปลงตามสภาพแวดล้อมที่เปลี่ยนไป
 5. Principle of Adaptability คือ หลักการการปรับปรุงเพื่อพัฒนาตัวเองให้สามารถคำนวณค่าที่เหมาะสมหรือค่าคำตอบที่ใกล้เคียงที่สุด ได้อย่างมีประสิทธิภาพ
- จากหลักการทั้ง 5 ประการดังกล่าวนำมาอธิบายขั้นตอนการทำงานของ PSO ดังนี้

1. กำหนดค่าเริ่มต้นให้แต่ละ Particle ดังนี้
 - กำหนดค่าของมิติ (Dimension) ซึ่งมีค่าตั้งแต่ 1 จนถึง n
 - กำหนดค่าของตำแหน่ง (Position)
 - กำหนดค่าของอัตราเร็ว (Velocity) ในการบิน
 - กำหนดค่าคงที่ของระดับความสามารถในการเรียนรู้ของ Particle
 - กำหนดค่าถ่วงน้ำหนัก (Inertia Weight) ของอัตราเร็วในการบินในการหาค่าผลเฉลยของสมการ
2. ประเมินค่าความเหมาะสมของแต่ละ Particle ซึ่งประเมินจากการนำผลเฉลยไปคำนวณในฟังก์ชันเป้าหมาย (Fitness Function)
3. เปรียบเทียบผลเฉลยที่ดีที่สุดที่ได้ (Current Values) จาก Particle รอบในนั้นกับค่าผลเฉลยที่ดีที่สุดจาก Particle รอบที่ผ่านมาว่าผลเฉลยของรอบไหนดีกว่า แล้วทำการเลือกผลเฉลยนั้นเป็นผลเฉลยที่ดีที่สุด
4. ปรับปรุงค่าของอัตราเร็วในการเคลื่อนที่ในรอบการทำงานของแต่ละ Particle
5. ปรับปรุงค่าของตำแหน่งในรอบการทำงานของแต่ละ Particle
6. ทำการวนซ้ำจนกระทั่งพบคำตอบที่เหมาะสมที่สุด หรือเมื่อมีจำนวนรอบของการทำซ้ำครบตามจำนวนที่กำหนดไว้แล้วจึงจบการทำงาน

2.4.5 การค้นหาแบบกลไกคัดสรรพันธุ์ที่ดีตามธรรมชาติ (Genetic Algorithms: GA)

แนวคิดของ GA ถูกคิดค้นโดยจอห์น ฮอลแลนด์ (John Holland) ในปี 1975 โดยที่ GA เป็นเทคนิคการหาค่าเหมาะที่สุด มีรากฐานแนวคิดมาจากทฤษฎีวิวัฒนาการของชาร์ลส์ดาร์วิน (Charles Darwin) โดยอิงจากแนวความคิดการอยู่รอดของผู้ที่แข็งแกร่งที่สุด (Survival of the fittest) ซึ่งการทำงานของ GA เป็นไปในลักษณะการหาคำตอบแบบคู่ขนาน (Parallel Search) โดยคำตอบที่ได้หนึ่งรุ่น (Generation) จะผ่านการแปลง (Transformation) เพื่อนำไปสู่การค้นหาคำตอบที่ดีขึ้นในรุ่นถัดไป ในปัจจุบันมีการประยุกต์ใช้ GA อย่างแพร่หลาย เช่น การใช้ GA ในการแก้ปัญหาทางคณิตศาสตร์ การแก้ปัญหาเส้นทางที่มีระยะทางสั้นที่สุด การจัดการตารางสอน การควบคุมหุ่นยนต์ รวมทั้งปัญหาทางด้านวิศวกรรมอื่นอีกมากมาย โดยที่องค์ประกอบหลักที่สำคัญของ GA มี 5 องค์ประกอบ คือ

1. รูปแบบโครโมโซมที่ใช้ในการนำเสนอทางเลือกที่สามารถเป็นไปได้ของแต่ละปัญหา
2. วิธีการสร้างประชากรต้นกำเนิด (Initial population) ของทางเลือกที่สามารถเป็นไปได้
3. ฟังก์ชันสำหรับประเมินค่าความเหมาะสม (Fitness function) เพื่อให้คะแนนแต่ละทางเลือก
4. กระบวนการทางพันธุกรรม (Genetic Operator) ซึ่งใช้ในการปรับเปลี่ยนองค์ประกอบของข้อมูลตลอดกระบวนการ ได้แก่ การคัดเลือก การจับคู่ การมิวเตชัน เป็นต้น
5. ค่าพารามิเตอร์ต่างๆ ที่จำเป็นสำหรับ GA เช่น ขนาดของประชากร ความน่าจะเป็นของตัวดำเนินการ และจำนวนรุ่น เป็นต้น

หลักการทำงานของ GA ประกอบด้วย

1. สร้างผลเฉลย (Generate Solution) ด้วยวิธีการสุ่มผลเฉลยที่เป็นไปได้ออกมาหลายผลเฉลย แต่ไม่จำเป็นต้องครบทุกรูปแบบของผลเฉลย
2. สร้างฟังก์ชันเป้าหมาย (Fitness Function) เพื่อคัดเลือกกลุ่มของผลเฉลยที่ดีเก็บไว้จำนวนหนึ่ง
3. นำผลเฉลยที่ทำการคัดเลือกจากฟังก์ชันเป้าหมายมาเป็นต้นแบบในการหาผลเฉลยของรุ่นถัดไป โดยการผสมผลเฉลยซึ่งเกิดจากเลียนแบบกระบวนการทางพันธุกรรม
4. นำคำตอบในรุ่นถัดไปมาคัดเลือกและผสมไปเรื่อยๆ จนกว่าจะได้คำตอบที่น่าพอใจ หรือ มีจำนวนรอบของการทำซ้ำครบตามที่กำหนดไว้แล้วจึงหยุดการทำงาน และทำการเลือกผลเฉลยที่ดีที่สุดออกมา

2.4.6 การค้นหาแบบการสลับของกบ (Shuffled Frog Leaping Algorithm: SFL)

SFL เป็นทฤษฎีที่ลอกเลียนแบบพฤติกรรมทางธรรมชาติในการค้นหาอาหารของกบซึ่งเป็นทฤษฎีที่มีการรวมส่วนดีพื้นฐานของ GA, วิธีมีมิติก (Memetic Algorithms: MA) และพื้นฐานพฤติกรรมสัตว์สังคมของ PSO โดย SFL มีกระบวนการทำงานตามขั้นตอนดังนี้

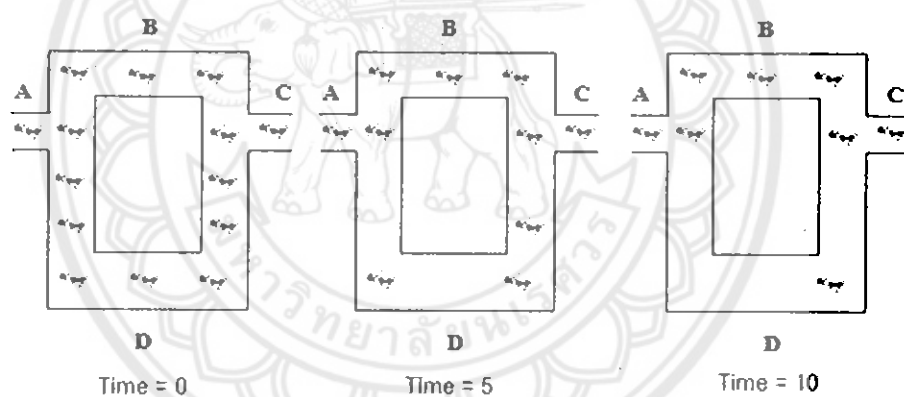
1. ขั้นตอนการสร้างประชากรเริ่มต้นของกบ (Population Initialization) ซึ่งทำโดยการสุ่มจำนวนกบ โดยให้กบแต่ละตัวแทนผลลัพธ์ (Solution) ที่เป็นไปได้หนึ่งค่า ซึ่งจำนวนกบที่ถูกสร้างขึ้นเป็นไปตามสัดส่วนของประชากร (Population Size) ที่กำหนดไว้
2. ขั้นตอนการประเมินค่าความเหมาะสม (Fitness Evaluation) เป็นขั้นตอนการถอดรหัสเพื่อคำนวณหาค่าความเหมาะสมตามฟังก์ชันเป้าหมายหรือฟังก์ชันความเหมาะสมของปัญหาที่ได้กำหนดไว้ และทำการเรียงลำดับค่าความเหมาะสมของกบจากมากไปหาน้อย
3. ขั้นตอนการแบ่งกลุ่มมีมิติก (Memeplex) เป็นขั้นตอนการเริ่มต้นการสร้างรุ่นของกบหลังจากการเรียงลำดับกบตามค่าความเหมาะสมเรียบร้อยแล้ว ให้ทำการแบ่งกบออกเป็น m กลุ่มย่อย ซึ่งจะเรียกกลุ่มของการแบ่งนี้ว่า มีมิติก (Memeplex) โดยมีมิติกจะประกอบไปด้วยกบจำนวน n_i ตัวนั้นหมายความว่าประชากรทั้งหมด (P) มีจำนวนเท่ากับ $\sum_{i=1}^m n_i$
4. ขั้นตอนการปรับปรุง (Improvement) เป็นขั้นตอนในการปรับปรุงกบที่มีค่าความเหมาะสมที่แย่ที่สุดให้มีค่าความเหมาะสมดีขึ้นเรื่อยๆ จนมีค่าความเหมาะสมดีกว่าความเหมาะสมของตัวที่ดีที่สุด แต่ถ้าไม่ดีกว่าจะทำการคัดผลเฉลยที่แย่ที่สุดนั้นออกไปจากมีมิติก และทำการหาผลเฉลยเริ่มต้นตัวใหม่มาทำการแทนที่ตัวที่คัดออกไป
5. ทำการวนซ้ำจนกระทั่งพบคำตอบที่เหมาะสมที่สุด หรือเมื่อมีจำนวนรอบของการทำซ้ำครบตามจำนวนที่กำหนดไว้แล้วจึงจบการทำงาน

2.4.7 การค้นหาแบบอาณานิคมมด (Ant Colony Optimization: ACO)

ACO ได้ถูกคิดค้นโดย Dorigo และคณะ ในปี 1991 เป็นวิธีแก้ปัญหาคณิตศาสตร์ที่อาศัยหลักการและแรงดลใจมาจากความฉลาดทางธรรมชาติของมดในการหาเส้นทางที่สั้นที่สุดระหว่างแหล่งอาหารและรัง โดยทั่วไปแล้วกระบวนการสร้างผลเฉลยของ ACO สามารถแยกได้เป็น 2 รูปแบบคือ การสร้างผลเฉลยแบบขนาน (Parallel Solution Construction) และการสร้างผลเฉลยแบบลำดับ (Sequential Solution Construction) ซึ่งลักษณะการสร้างผลเฉลยแบบขนานคือ ในแต่ละขั้นตอนของการสร้างผลเฉลย มดทุกตัวในตำแหน่งเดิมจะเลือกตำแหน่งถัดไปพร้อมกัน จนกระทั่ง

มดทุกตัวได้ผลเฉลยหรือเส้นทางที่สมบูรณ์พร้อมกัน ในขณะที่การสร้างผลเฉลยแบบลำดับ มดจะถูกปล่อยให้สร้างผลเฉลยที่ละตัวจนกระทั่งมดดังกล่าวได้ผลเฉลยที่สมบูรณ์ จากนั้นจึงให้มดตัวถัดไปเริ่มสร้างผลเฉลยใหม่

มดจะใช้การติดต่อสื่อสารระหว่างกันทางอ้อม (Indirect Communication or Stigmergy) ด้วยสารเคมีชนิดหนึ่งที่มีชื่อว่า สารฟีโรโมน (Pheromone) ซึ่งเป็นสารเคมีที่มดจะทิ้งไว้บนเส้นทางเดิน ในขณะที่มดเดินทาง และเมื่อมดตัวอื่น ได้กลิ่นสารฟีโรโมนจะทำการเพิ่มความเป็นไปได้ (Probability) ที่มดตัวนั้นจะเลือกเดินทางเดียวกับมดตัวแรก พร้อมกับปล่อยสารฟีโรโมนไว้บนเส้นทางนั้นเพิ่มขึ้น ไปอีกทำให้เส้นทางดังกล่าวมีปริมาณสารฟีโรโมนเพิ่มมากขึ้น ซึ่งกลไกนี้เรียกว่า กระบวนการป้อนกลับแบบบวก (Positive Feedback or Autocatalytic Mechanism) นอกจากนี้ธรรมชาติของสารฟีโรโมนยังมีการระเหย (Pheromone Decay) อยู่ตลอดเวลา เส้นทางใดก็ตามที่มีปริมาณสารฟีโรโมนน้อยลง ความเป็นไปได้ที่มดจะเลือกเส้นทางดังกล่าวจึงมีน้อยลง จนท้ายที่สุดปริมาณสารฟีโรโมนบนเส้นทางนั้นจะถูกระเหยไปจนหมด ซึ่งเรียกกลไกนี้ว่า กระบวนการป้อนกลับแบบลบ (Negative Feedback)



รูปที่ 2.15 แผนภาพแสดงการเลือกทางเดินที่สั้นที่สุดของมด

จากรูปที่ 2.15 ซึ่งเป็นตัวอย่างการหาเส้นทางเดินของมดสามารถอธิบายได้ว่าที่เวลาเท่ากับศูนย์มดจะทำการเดินจากจุด A ไปยังจุด C โดยใช้หลักการสุ่ม (Random) สังเกตได้ว่าจำนวนของมดที่เส้นทาง A-B-C และเส้นทาง A-D-C มีปริมาณใกล้เคียงกัน เมื่อเวลาผ่านไป Time = 5 ปริมาณของมดในเส้นทาง A-B-C เริ่มมีจำนวนมากกว่าปริมาณมดในเส้นทาง A-D-C เนื่องจากเส้นทาง A-B-C มีระยะทางที่สั้นกว่าเส้นทาง A-D-C จึงมีจำนวนมดเดินผ่านมากกว่า ส่งผลให้มีปริมาณสารฟีโรโมนบนเส้นทาง A-B-C มากขึ้นพิจารณา Time = 10 พบว่ามดส่วนใหญ่เลือกเส้นทาง A-B-C ซึ่งเป็นเส้นทางที่สั้นที่สุดได้

จากแนวคิดนี้จึงได้เกิดวิธีการมดเทียม (Artificial Ants) ขึ้นมาโดยการจำลองวิธีการแก้ปัญหาในการค้นหาเส้นทางที่ดีที่สุดของมดจริงตามธรรมชาติมาประยุกต์ใช้ในการแก้ปัญหา

ต่างๆของมนุษย์ซึ่งยังคงไว้ตามหลักการของมดจริง ดังนั้นจึงเรียกวีธีการแก้ปัญหาโดยอาศัยหลักการของมดนี้ว่า การค้นหาแบบอาณานิคมมด จากนั้นวิธีการค้นหาแบบอาณานิคมมด ได้มีการพัฒนาอย่างต่อเนื่อง ดังแสดงไว้ในตารางที่ 2.11

ตารางที่ 2.11 ACO และผู้คิดค้น

วิธีการ	ผู้คิดค้น
Ant system (AS)	Dorigo, Maniezzo and Colonies(1991)
Elitist ant system (EAS)	Dorigo, Maniezzo and Colonies(1992)
Ant – Q	Gambadella and Dorigo (1995)
Rank – Based ant system (AS-rank)	Bullnheimer, Harlt and Strauss(1997)
Max-Min ant system (MMAS)	Stutzle and Hoos(1997)
Ant colony system (ACS)	Dorigo and Gambardella(1997)

สำหรับขอบเขตของการศึกษาในโครงการนี้จะศึกษาเฉพาะวิธีการระบบของอาณานิคมมด (Ant Colony System: ACS) สำหรับรายละเอียดของขั้นตอนวิธีทาง ACS รวมทั้งแนวทางประยุกต์ในการปรับปรุงค่าพารามิเตอร์ในตัวควบคุมแบบ PID ถูกนำเสนอในบทที่ 3

บทที่ 3

ขั้นตอนการดำเนินงาน

จากการศึกษาการออกแบบระบบควบคุมแบบ PID ดังที่ได้กล่าวในบทที่ 2 นั้นผลตอบสนองที่ได้ยังไม่เป็นที่น่าพอใจเนื่องจากยังมีค่าเวลาขึ้น (Rise Time), เวลาสูงสุด (Peak Time), ค่าพุ่งสูงสุด (Maximum Overshoot) และเวลาสู่สมดุล (Setting Time) อยู่ในระดับที่ไม่ดีเท่าที่ควร ดังนั้นโครงการนี้จึงได้นำการค้นหาค่าพารามิเตอร์ด้วยขั้นตอนวิธีของอาณานิคมมาใช้ในการปรับพารามิเตอร์ของระบบควบคุมแบบ PID เพื่อให้ผลตอบสนองของระบบที่ได้ดีขึ้น

3.1 การประยุกต์ระบบของอาณานิคม

วิธีการของระบบของอาณานิคม (ACS) นั้นแตกต่างจากระบบมด (AS) อยู่ 3 ประการ กล่าวคือ ประการแรกการทำงานในแต่ละรอบจะมีการเพิ่มปริมาณฟีโรโมนเฉพาะเส้นทางของมดที่ดีที่สุดของแต่ละรอบเท่านั้น ประการที่สอง วิธีการของ ACS มีการใช้ประโยชน์จากข้อมูลการเดินทางของมดที่ผ่านมาทั้งหมด (Pheromone Trail) เพื่อใช้ในการเลือกตำแหน่งถัดไป และประการสุดท้ายในแต่ละการเดินทางของมดจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งจะมีการระเหยของปริมาณฟีโรโมนบางส่วนออกเพื่อให้มดมีโอกาสเลือกเส้นทางและเพิ่มโอกาสในการสำรวจเส้นทางใหม่ทั้งหมด

เมื่อนำวิธีการของ ACS มาประยุกต์ใช้กับการปรับพารามิเตอร์ของระบบควบคุมแบบ PID มีขั้นตอนการทำงานดังต่อไปนี้

1. การสร้างตารางและเส้นทางอ้างอิงเริ่มต้น

การสร้างตารางและเส้นทางอ้างอิงเริ่มต้นเกิดจากการนำมาค่าพารามิเตอร์ของตัวควบคุม PID มาใช้โดยสมมติให้ K_p , K_i และ K_d เขียนอยู่ในรูปแบบต่อไปนี้

$$\begin{aligned} K_p &= (K_{p,n_p} \times 10^{n_p}) + (K_{p,n_p-1} \times 10^{n_p-1}) + \dots + (K_{p,0} \times 10^0) + (K_{p,-1} \times 10^{-1}) + \dots \\ &\quad + (K_{p,-m_p} \times 10^{-m_p}) \\ &= \sum_{j=0}^{n_p} K_{p,n_p-j} \times 10^{n_p-j} + \sum_{j=1}^{m_p} K_{p,-j} \times 10^{-j} \end{aligned} \quad (3.1)$$

$$\begin{aligned} K_i &= (K_{i,n_i} \times 10^{n_i}) + (K_{i,n_i-1} \times 10^{n_i-1}) + \dots + (K_{i,0} \times 10^0) + (K_{i,-1} \times 10^{-1}) + \dots \\ &\quad + (K_{i,-m_i} \times 10^{-m_i}) \end{aligned}$$

$$= \sum_{j=0}^{n_i} K_{i,n_i-j} \times 10^{n_i-j} + \sum_{j=1}^{m_i} K_{i,-j} \times 10^{-j} \quad (3.2)$$

$$K_d = (K_{d,n_d} \times 10^{n_d}) + (K_{d,n_d-1} \times 10^{n_d-1}) + \dots + (K_{d,0} \times 10^0) + (K_{d,-1} \times 10^{-1}) + \dots \\ + (K_{d,-m_d} \times 10^{-m_d})$$

$$= \sum_{j=0}^{n_d} K_{d,n_d-j} \times 10^{n_d-j} + \sum_{j=1}^{m_d} K_{d,-j} \times 10^{-j} \quad (3.3)$$

โดยที่ n_p คือจำนวนเต็มหน้าจุดทศนิยมของ K_p

n_i คือจำนวนเต็มหน้าจุดทศนิยมของ K_i

n_d คือจำนวนเต็มหน้าจุดทศนิยมของ K_d

m_p คือจำนวนทศนิยมของ K_p

m_i คือจำนวนทศนิยมของ K_i

m_d คือจำนวนทศนิยมของ K_d

เพื่อความสะดวกในการคำนวณจึงกำหนดให้จำนวนของตำแหน่งทศนิยมเท่ากัน นั่นคือ $m_p = m_i = m_d = m$ ซึ่งจะได้ว่า

$$K_p = \sum_{j=0}^{n_p} K_{p,n_p-j} \times 10^{n_p-j} + \sum_{j=1}^m K_{p,-j} \times 10^{-j} \quad (3.4)$$

$$K_i = \sum_{j=0}^{n_i} K_{i,n_i-j} \times 10^{n_i-j} + \sum_{j=1}^m K_{i,-j} \times 10^{-j} \quad (3.5)$$

$$K_d = \sum_{j=0}^{n_d} K_{d,n_d-j} \times 10^{n_d-j} + \sum_{j=1}^m K_{d,-j} \times 10^{-j} \quad (3.6)$$

นำสัมประสิทธิ์ $K_{p,j}$, $K_{i,j}$ และ $K_{d,j}$ มาเขียนอยู่ในรูปของเวกเตอร์ \tilde{K}_p , \tilde{K}_i และ \tilde{K}_d ตามลำดับดังนี้

$$\tilde{K}_p = [K_{p,n_p} \quad K_{p,n_p-1} \quad \dots \quad K_{p,0} \vdots K_{p,-1} \quad \dots \quad K_{p,-m}] \quad (3.7)$$

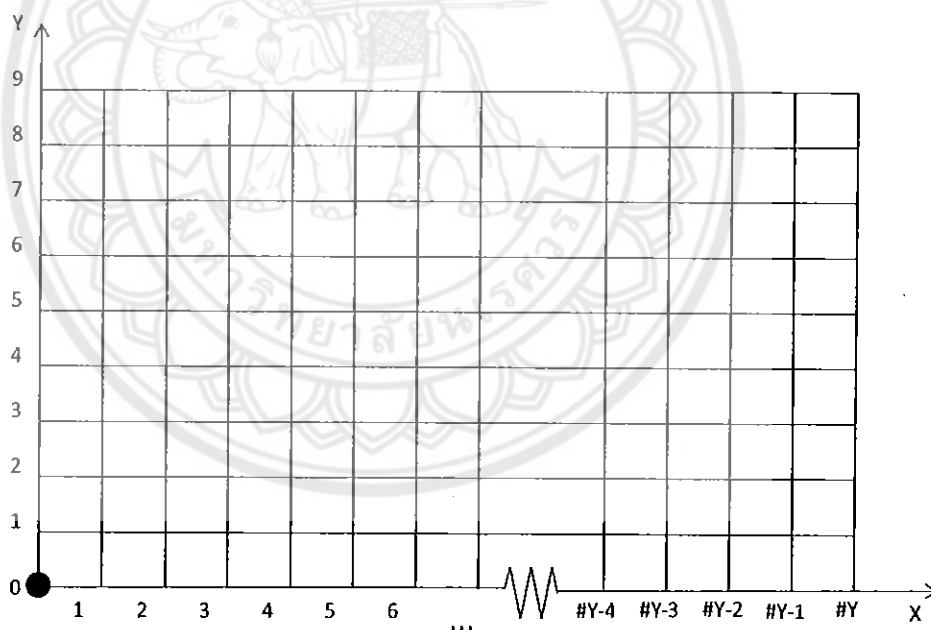
$$\tilde{K}_i = [K_{i,n_i} \quad K_{i,n_i-1} \quad \dots \quad K_{i,0} \vdots K_{i,-1} \quad \dots \quad K_{i,-m}] \quad (3.8)$$

$$\tilde{K}_d = [K_{d,n_d} \quad K_{d,n_d-1} \quad \cdots \quad K_{d,0} : K_{d,-1} \quad \cdots \quad K_{d,-m}] \quad (3.9)$$

จากนั้นสร้างเวกเตอร์ \tilde{Y} ให้อยู่ในรูปแบบดังนี้

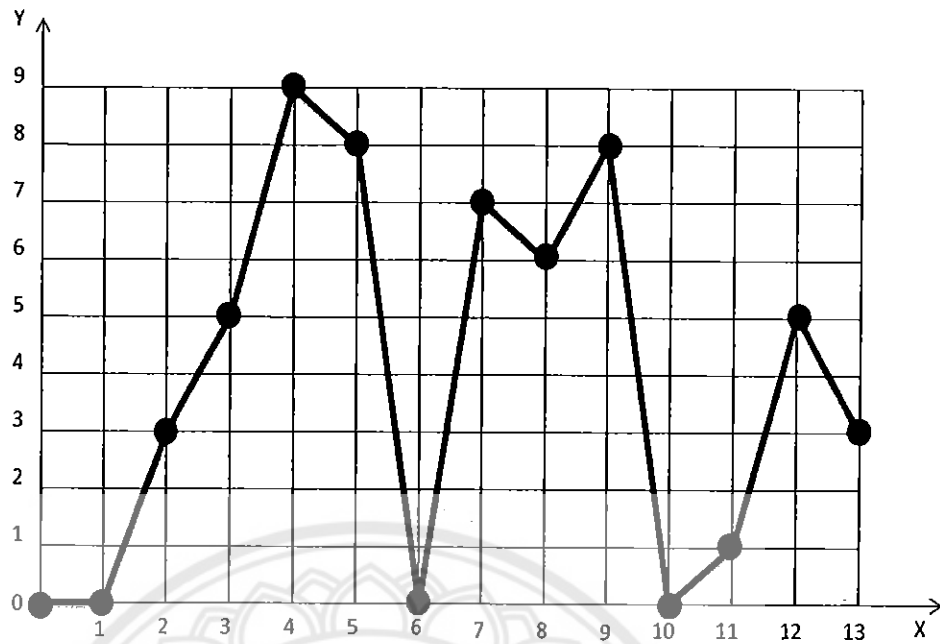
$$\tilde{Y} = [0 : \tilde{K}_p : 0 : \tilde{K}_i : 0 : \tilde{K}_d] \quad (3.10)$$

เนื่องจาก \tilde{Y} มีขนาด $1 \times (3 + [n_p + 1 + m] + [n_i + 1 + m] + [n_d + 1 + m])$ นั่นคือ \tilde{Y} มีจำนวนสมาชิกอยู่ $n_p + n_i + n_d + 3m + 6$ ตัว เพื่อความสะดวกจึงกำหนดให้ $\#Y$ คือ จำนวนสมาชิกของ \tilde{Y} ดังนั้นจะได้ว่า $\#Y = n_p + n_i + n_d + 3m + 6$ จากนั้นนำมาสร้างตารางโดยให้แกน x มีค่าตั้งแต่ $0, 1, 2, \dots, \#Y$ ส่วนในแกน y มีค่า $0, 1, 2, \dots, 9$ ซึ่งเป็นสัญลักษณ์ในเลขฐานสิบ ตารางดังกล่าวแสดงไว้ในรูปที่ 3.1 โดยจุดเริ่มต้นของการเดินอยู่ตำแหน่ง $(x, y) = (0, 0)$



รูปที่ 3.1 เส้นทางเดินของมด

เมื่อได้ตารางเป็นที่เรียบร้อยแล้วการสร้างเส้นทางอ้างอิงทำได้โดยนำค่าในเวกเตอร์ \tilde{Y} มาจุดลงในตารางแล้วลากเส้นเชื่อมถึงกันดังรูปที่ 3.2



รูปที่ 3.2 ตัวอย่างทำการปล่อยมดในเส้นทางเดิน

ตัวอย่างในรูปที่ 3.2 แสดงเส้นทางอ้างอิง เมื่อ

$$\tilde{Y} = [0 \ : \ 3 \ 5 \ 9 \ 8 \ : \ 0 \ : \ 7 \ 6 \ 8 \ : \ 0 \ : \ 1 \ 5 \ 3]$$

2. การเลือกเส้นทางของมด (Tour Construction)

เนื่องจากแต่ละจุดในตารางสอดคล้องกับค่า K_p , K_i และ K_d ดังนั้นการปรับตำแหน่งของจุดย่อมส่งผลให้ค่า K_p , K_i และ K_d มีการเปลี่ยนแปลงไปด้วย

ในการปรับตำแหน่งของจุดนั้นต้องอาศัยหลักการของความน่าจะเป็น กล่าวคือ หากตำแหน่งปัจจุบันของมดอยู่ที่ (i, j) การที่มดจะเดินทางไปตำแหน่ง $(i+1, j')$ โดยที่ $j' = 0, 1, 2, \dots, 9$ ขึ้นกับค่าความน่าจะเป็นดังสมการที่ 3.11

$$P_K(i+1, j') = \begin{cases} \frac{\tau_K^\alpha(i+1, j') \eta_K^\beta(i+1, j')}{\sum_{j=0}^9 \tau_K^\alpha(i+1, j) \eta_K^\beta(i+1, j)} & , \text{if } q_j \leq q^* \\ 0 & , \text{otherwise} \end{cases} \quad (3.11)$$

- โดยที่ $P_K(i+1, j')$ คือ ความน่าจะเป็นของมดตัวที่ K
- $\tau_K(i+1, j')$ คือ ค่าฟีโรโมนของมดตัวที่ K
- α คือ ค่าถ่วงน้ำหนักของปริมาณฟีโรโมน

β	คือ ค่าถ่วงน้ำหนักของข้อมูลการศึกษาสำนึก
q^*	คือ ค่าความเชื่อมั่นในเส้นทางอ้างอิง
q_j	คือ ค่าที่ได้จากการสุ่มแบบเอกภาพ (Uniform)
$\eta_K(i+1, j')$	คือ ข้อมูลการศึกษาสำนึก (Heuristic information) ของ มดตัวที่ K คำนวณได้จากสูตร

$$\eta_K(i+1, j') = \left(\frac{10 - |j' - j^*|}{10} \right)_K \quad (3.12)$$

เมื่อ j^* คือ ค่าบนแกน y ของเส้นทางอ้างอิง ณ ตำแหน่ง $x=i+1$

ยกตัวอย่างเช่น ในการคำนวณหาค่าความน่าจะเป็นจากจุด $(1, 0)$ ไปยังจุด $(2, j')$ เมื่อ $j' = 0, 1, 2, \dots, 9$ โดยสมมติให้ $j^* = 3$ และ $\tau(2, j') = 1$ ทุกค่า j' จะได้ว่า

$$\begin{aligned} \eta(2, 0) &= \frac{10 - |0 - 3|}{10} = 0.7 \quad \text{และ} \quad P(2, 0) = \frac{0.7}{7.3} = 0.0959 \\ \eta(2, 1) &= \frac{10 - |1 - 3|}{10} = 0.8 \quad \text{และ} \quad P(2, 1) = \frac{0.8}{7.3} = 0.1096 \\ \eta(2, 2) &= \frac{10 - |2 - 3|}{10} = 0.9 \quad \text{และ} \quad P(2, 2) = \frac{0.9}{7.3} = 0.1233 \\ \eta(2, 3) &= \frac{10 - |3 - 3|}{10} = 1.0 \quad \text{และ} \quad P(2, 3) = \frac{1}{7.3} = 0.1369 \\ \eta(2, 4) &= \frac{10 - |4 - 3|}{10} = 0.9 \quad \text{และ} \quad P(2, 4) = \frac{0.9}{7.3} = 0.1233 \\ \eta(2, 5) &= \frac{10 - |5 - 3|}{10} = 0.8 \quad \text{และ} \quad P(2, 5) = \frac{0.8}{7.3} = 0.1096 \\ \eta(2, 6) &= \frac{10 - |6 - 3|}{10} = 0.7 \quad \text{และ} \quad P(2, 6) = \frac{0.7}{7.3} = 0.0959 \\ \eta(2, 7) &= \frac{10 - |7 - 3|}{10} = 0.6 \quad \text{และ} \quad P(2, 7) = \frac{0.6}{7.3} = 0.0822 \\ \eta(2, 8) &= \frac{10 - |8 - 3|}{10} = 0.5 \quad \text{และ} \quad P(2, 8) = \frac{0.5}{7.3} = 0.0685 \\ \eta(2, 9) &= \frac{10 - |9 - 3|}{10} = 0.4 \quad \text{และ} \quad P(2, 9) = \frac{0.4}{7.3} = 0.0548 \end{aligned}$$

ซึ่งผลที่ได้นำมาสรุปไว้ในตารางที่ 3.1

ตารางที่ 3.1 การคำนวณค่าความน่าจะเป็น

j'	$\eta(2, j')$	$\tau(2, j')$	$P(2, j')$
0	0.7	1	0.0959
1	0.8	1	0.1096
2	0.9	1	0.1233
3	1	1	0.1369
4	0.9	1	0.1233
5	0.8	1	0.1096
6	0.7	1	0.0959
7	0.6	1	0.0822
8	0.5	1	0.0685
9	0.4	1	0.0548

จากตารางที่ 3.1 ให้เลือกค่า $P(2, j')$ ที่มีค่ามากที่สุด ซึ่งจากตัวอย่าง $P(2, 3)$ มีค่ามากที่สุด ดังนั้นจะทำการเดินจากจุด (1, 0) ไปยังจุด (2, 3)

3. การคำนวณหาฟังก์ชันเป้าหมาย (Fitness Function)

เมื่อมคเดินทางครบจนได้เส้นทางที่สมบูรณ์แล้ว จึงนำเส้นทางที่ได้มาแปลค่าเป็นพารามิเตอร์ K_p , K_i และ K_d เพื่อใส่ให้กับตัวควบคุมแบบ PID จากนั้นจึงทำการจำลอง (simulate) หาผลตอบสนองที่ได้ แล้วนำมาคำนวณหาค่าความคลาดเคลื่อน ซึ่งเป็นฟังก์ชันเป้าหมายดังสมการที่ 3.13

$$f^K = ITAE = \int_0^{\infty} |e^K(t)| dt \quad (3.13)$$

โดยที่ $e^K(t) = r(t) - y^K(t)$ คือ ผลต่างระหว่างสัญญาณอ้างอิง และผลตอบสนองอันเกิดจากมคตัวที่ K

เนื่องจากการหาค่าปริพันธ์จาก 0 ถึง ∞ ไม่สามารถคำนวณได้ในทางปฏิบัติ ดังนั้นจึงทำการประมาณค่าได้ดังนี้

แบ่งช่วงของเวลาที่ต้องการจำลองออกเป็น $\#P$ จุด โดยแต่ละจุดมีระยะห่างเท่ากัน ดังนั้นที่เวลา t_k ค่าความผิดพลาดอันเกิดจากมดตัวที่ K คือ $e_k^K = r_k - y_k^K$ และค่าความคลาดเคลื่อนจึงประมาณได้เป็น

$$f^K \cong \left(\sum_{k=0}^{\#P} (r_k - y_k^K) \times t_k \right) \times \frac{T_f}{\#P} \quad (3.14)$$

โดยที่	r_k	คือ สัญญาณเข้า ณ เวลา t_k
	y_k	คือ สัญญาณออก ณ เวลา t_k
	T_f	คือ จุดสิ้นสุดของการหาปริพันธ์
	$\#P$	คือ จำนวนความละเอียดของช่วงการหาปริพันธ์

4. การเปลี่ยนแปลงปริมาณของฟีโรโมน (Update of Pheromone Trails)

เมื่อกำหนดค่าความคลาดเคลื่อนจนครบมดทุกตัวแล้ว ให้เลือกมดตัวที่ K^* ที่ซึ่งค่าความคลาดเคลื่อนมีค่าน้อยที่สุดถ้าความคลาดเคลื่อนอันเกิดจากเส้นทางของมดตัวที่ K^* ดังกล่าวมีค่ามากกว่าค่าความคลาดเคลื่อนอันเกิดจากเส้นทางอ้างอิง ให้ทำการปล่อยมดในรอบถัดไปโดยไม่มีการปรับฟีโรโมน ในทางตรงกันข้ามถ้าค่าความคลาดเคลื่อนอันเกิดจากเส้นทางของมดตัวที่ K^* มีค่าน้อยกว่าหรือเท่ากับค่าความคลาดเคลื่อนอันเกิดจากเส้นทางอ้างอิง ให้ใช้เส้นทางของมดตัวที่ K^* เป็นเส้นทางอ้างอิงในการคำนวณรอบถัดไป พร้อมทั้งปรับปริมาณของฟีโรโมน ตามสมการที่ 3.15

$$\tau_{K^*}^{\alpha}(i+1, j') = [(1-\rho)\tau(i+1, j') + \rho\Delta\tau(i+1, j')]_{K^*}^{\alpha} \quad (3.15)$$

โดยที่ $\Delta\tau(i+1, j') = 1 / \text{fitness} = 1 / ITAE$

การคำนวณจะทำการวนซ้ำจนกระทั่งค่าความคลาดเคลื่อนอยู่ในช่วงที่ยอมรับได้ หรือครบจำนวนรอบที่กำหนดไว้

หมายเหตุ:

การกำหนดค่าพารามิเตอร์ในวิธีการอาณานิคมมดสามารถแยกได้เป็น 2 ประเภท คือ พารามิเตอร์ของปัจจัยหลัก (Primary Factors) คือ ปัจจัยที่ใช้กับอาณานิคมมดทุกรูปแบบ ซึ่งปัจจัยเหล่านี้ได้แก่ ค่าถ่วงน้ำหนักของปริมาณฟีโรโมน (α) ค่าถ่วงน้ำหนักของข้อมูลที่ใช้ในการสุ่ม (β) และอัตราการระเหยของฟีโรโมน (ρ) โดยทั่วไปค่า α , β และ ρ มีค่าอยู่ในช่วง 0.01-1, 0-5 และ 0.01-1 ตามลำดับและพารามิเตอร์ของปัจจัยเฉพาะ (Specific Factors) คือ ปัจจัยที่ใช้ได้เฉพาะกับวิธีการอาณานิคมมดในแต่ละรูปแบบเท่านั้น ยกตัวอย่างเช่น ค่าการสุ่ม (Random Exploration: q_0) ใช้เฉพาะกับวิธีการของ ACS เท่านั้น

ในการประยุกต์การค้นหาผลเฉลยแบบศึกษาสำนึกด้วยขั้นตอนวิธีของอาณานิคมมด มาใช้ในการปรับปรุงค่าพารามิเตอร์ PID เพื่อให้ผลตอบสนองของระบบดีขึ้นมีรหัสเทียม (Pseudo Code) ได้ดังนี้

Main()

```

input (num, den) or (A, B, C, D)
input  $K_p$ ,  $K_i$ ,  $K_d$ 
Y_Star = [0:Kp:0:Ki:0:Kd]
for round = 1 to round_max
  for ant = 1 to ant_max
    find path()
    find error of path()
  end
  choose minimum_error
  if minimum_error < error of Y_Star
    update pheromone()
    updateY_Star()
  end
end
end
end

```

รูปที่ 3.3 แสดงรหัสเทียมของฟังก์ชันหลัก

```

find path()
  for node = 1 to node_max
    for y = 0 to 9
      random q (0 < q < 1)
      if q < q0
        prob(y) = calculate prob according to eq (3.11)
      else
        prob(y) = 0
      end
    end
    choose maximum prob
  end
  remember path
end

```

รูปที่ 3.4 แสดงรหัสเทียมของฟังก์ชันหาเส้นทาง

```

find error of path()
  for k = 1: length(T)
    ITAE = calculate ITAE according to eq (3.14)
  end

```

รูปที่ 3.5 แสดงรหัสเทียมของฟังก์ชันหาความคลาดเคลื่อน

```

update pheromone()
  for node = 1 to node_max
    pheromone = calculate pheromone according to eq (3.15)
  end

```

รูปที่ 3.6 แสดงรหัสเทียมของฟังก์ชันปรับฟีโรโมน (Update pheromone)

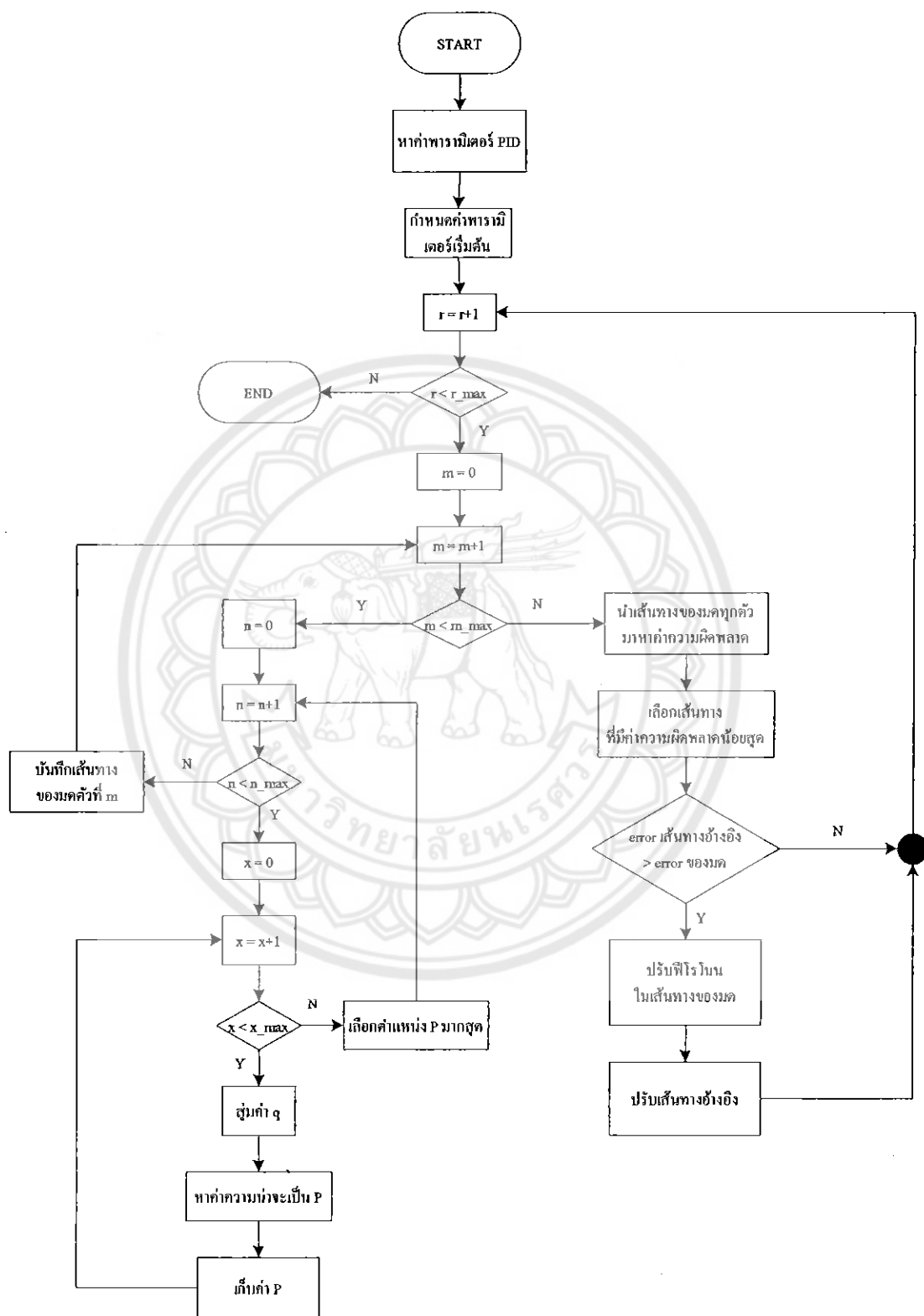
```

updateY_Star()
  Y_Star = choose path that yields minimum_error
end

```

รูปที่ 3.7 แสดงรหัสเทียมของฟังก์ชันปรับเส้นทางอ้างอิง

จากรหัสเทียมข้างต้นสามารถแสดงเป็นแผนภาพการไหลได้ดังรูปที่ 3.8



รูปที่ 3.8 แผนภาพการปรับพารามิเตอร์ PID ด้วยขั้นตอนวิธีของ ACS

โดยที่ r	คือ รอบของการทำงาน
r_max	คือ จำนวนรอบสูงสุด
m	คือ มดตัวที่ m
m_max	คือ จำนวนมดสูงสุด
n	คือ ค่าของแกน x
n_max	คือ ค่าของ $\#Y$
x	คือ สัญลักษณ์ในเลขฐานสิบ นั่นคือ $y = 0, 1, \dots, 9$
x_max	คือ จำนวนตัวเลขของสัญลักษณ์ในเลขฐานสิบ
P	คือ ค่าความน่าจะเป็นซึ่งได้จากการคำนวณตามสมการที่ 3.11
q	คือ ค่าสุ่มแบบเอกภาพอยู่ในช่วงปิด $0 - 1$
error	คือ ค่าความผิดพลาด

สรุปการทำงานของโปรแกรม

1. หลังจากรับค่าพารามิเตอร์ K_p, K_i และ K_d แล้วจึงนำค่าดังกล่าวมาแปลงเป็นเลขโดดเพื่อนำไปเก็บไว้ใน \tilde{Y} ตามสมการที่ 3.10 ใช้เป็นเส้นทางอ้างอิง
2. สังเคราะห์ตัวควบคุมแบบ PID โดยใช้พารามิเตอร์ K_p, K_i และ K_d จากขั้นตอนที่ 1 พร้อมคำนวณหาผลตอบสนองของระบบวงปิดเพื่อนำมาใช้คำนวณหาค่าความคลาดเคลื่อนตามสมการที่ 3.14 ซึ่งเป็นค่าความคลาดเคลื่อนของเส้นทางอ้างอิง
3. ทำการปล่อยมดจากจุดเริ่มต้น
4. ทำการสุ่มค่า q_j เพื่อเปรียบเทียบกับค่าที่กำหนดไว้ q^* ในการหาความน่าจะเป็นที่มดจะเลือกเส้นทางในจุดถัดไปตามสมการที่ 3.11 โดย j' มีค่าเท่ากับ 0 ถึง 9 เมื่อได้ค่าความน่าจะเป็นทั้งหมดจึงเลือก j' ที่มีความน่าจะเป็นมากที่สุดแล้วให้มดเลือกเดินไปยังตำแหน่งดังกล่าว
5. ทำการตรวจสอบว่ามดเดินทางจนถึงจุดสุดท้ายหรือไม่ ถ้ามดยังเดินทางไม่ถึงจุดสุดท้ายให้ย้อนกลับไปทำขั้นตอนที่ 4 จนกระทั่งมดเดินทางถึงจุดสุดท้าย เมื่อมดเดินทางถึงจุดสุดท้ายแล้วให้ทำการบันทึกเส้นทางของมดพร้อมตรวจสอบว่ามดเดินทางครบทุกตัวแล้วหรือไม่ ถ้ามดยังเดินทางไม่ครบทุกตัวให้ทำการปล่อยมดตัวถัดไปแล้วกลับไปทำในขั้นตอนที่ 3 แต่ถ้าปล่อยมดครบแล้วให้ทำงานในขั้นตอนที่ 6 ต่อไป

6. นำเส้นทางของมดแต่ละตัวมาแปลงให้เป็นค่าพารามิเตอร์ K_p , K_i และ K_d จากนั้นนำค่าพารามิเตอร์ที่ได้จากมดแต่ละตัว มาคำนวณหาค่าความคลาดเคลื่อน จากนั้นเลือกมดตัวที่มีค่าความคลาดเคลื่อนน้อยที่สุด
7. นำมดตัวที่มีค่าความคลาดเคลื่อนน้อยที่สุดมาเปรียบเทียบกับค่าความผิดพลาดของเส้นทางอ้างอิง ถ้าค่าความผิดพลาดของมดตัวดังกล่าวมีค่าน้อยกว่าค่าความผิดพลาดของเส้นทางอ้างอิง ให้ทำงานในขั้นตอนที่ 8 ต่อไป แต่ถ้าไม่เป็นเช่นนั้นให้ขึ้นรอบการทำงานใหม่แล้วกลับไปทำงานในขั้นตอนที่ 3
8. ทำการปรับพีโรโมนตามเส้นทางของมดตัวที่มีค่าความคลาดเคลื่อนน้อยที่สุดตามสมการที่ 3.15 จากนั้นทำการเปลี่ยนเส้นทางอ้างอิงใหม่เป็นเส้นทางเดินของมดตัวดังกล่าว
9. ตรวจสอบรอบการทำงานว่าครบรอบหรือไม่ ถ้ารอบการทำงานยังไม่ครบรอบให้ขึ้นรอบการทำงานรอบใหม่แล้วกลับไปทำงานในขั้นตอนที่ 3 แต่ถ้ารอบการทำงานครบแล้วให้ทำงานในขั้นตอนที่ 10 ต่อไป
10. นำค่าเส้นทางอ้างอิงที่บันทึกไว้มาแปลงให้เป็นค่าพารามิเตอร์ K_p , K_i และ K_d จากนั้นแสดงผลค่าดังกล่าวออกมาเป็นค่าพารามิเตอร์ K_p , K_i และ K_d หลังจากปรับปรุ้งด้วย ACS

3.2 ตัวอย่างการประยุกต์ระบบของอาณานิคมมด

จากตัวอย่างที่ 2.6 ซึ่งเป็นการคำนวณค่าพารามิเตอร์ K_p , K_i และ K_d ด้วยวิธีของ ZN, วิธีของ CHR และวิธีของ WJC โดยมีผลตอบสนองแสดงไว้ในรูปที่ 2.14 ในหัวข้อนี้เป็นการประยุกต์วิธีของ ACS ในการปรับปรุงพารามิเตอร์ ซึ่งคำนวณได้จากวิธีดังกล่าวข้างต้น

พิจารณาระบบ

$$G(s) = \frac{64}{s^3 + 14s^2 + 56s + 64}$$

โดยตัวควบคุมที่คำนวณด้วยวิธีของ ZN คือ

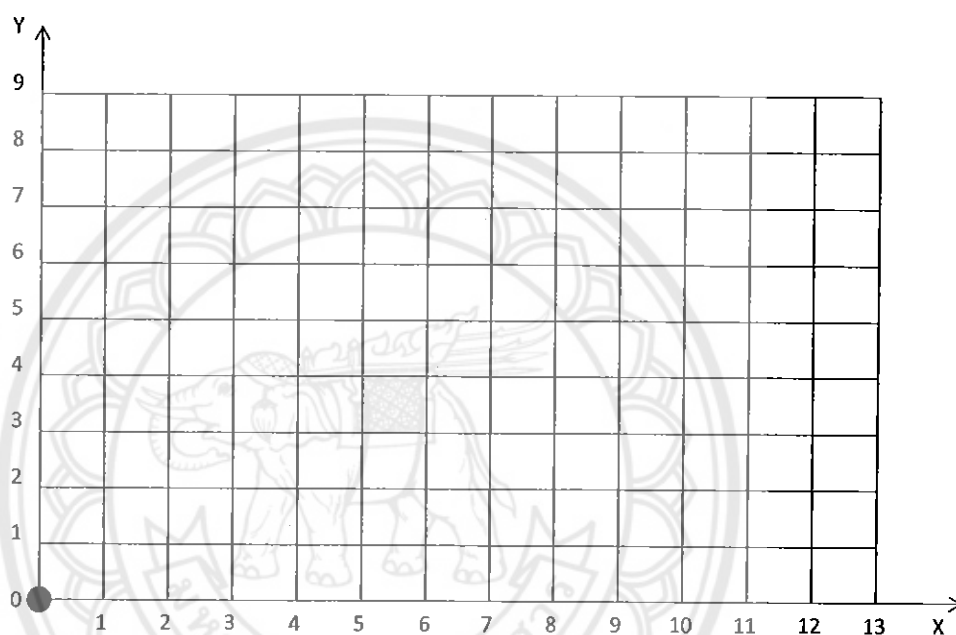
$$G_c(s) = 0.76s + 6.75 + \frac{15}{s}$$

นั่นคือ $K_p = 6.75$, $K_i = 15$ และ $K_d = 0.76$

1. จากข้อมูลข้างต้นสามารถทำการสร้างเวกเตอร์ \tilde{Y} โดยกำหนดให้ $m_p = m_i = m_d = m = 2$ ซึ่งจะได้ว่า

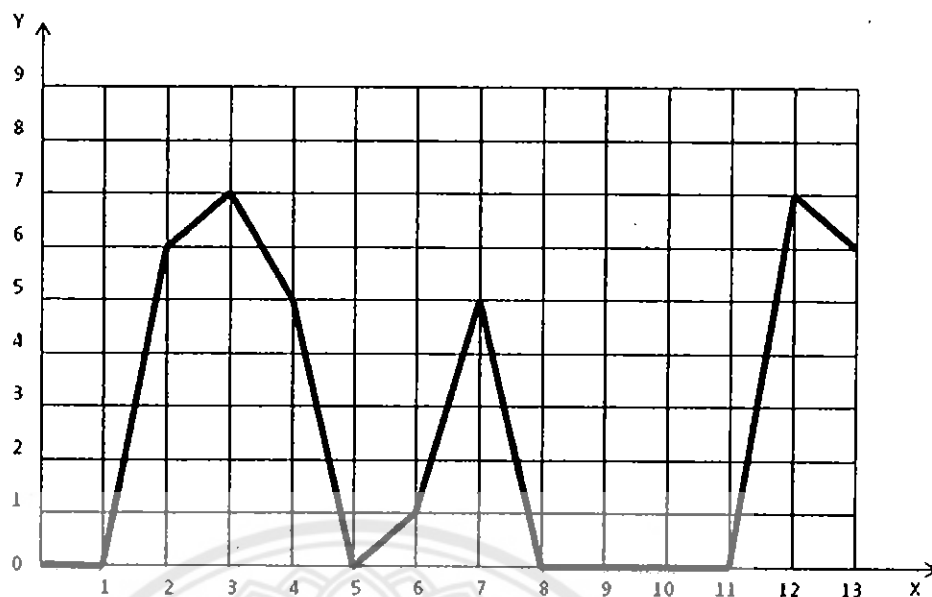
$$\tilde{Y} = [0 \ ; \ 6 \ 7 \ 5 \ ; \ 0 \ ; \ 1 \ 5 \ 0 \ 0 \ ; \ 0 \ ; \ 0 \ 7 \ 6]$$

เนื่องจาก \tilde{Y} มีสมาชิก 13 ตัว ดังนั้นจึงได้ตารางดังต่อไปนี้



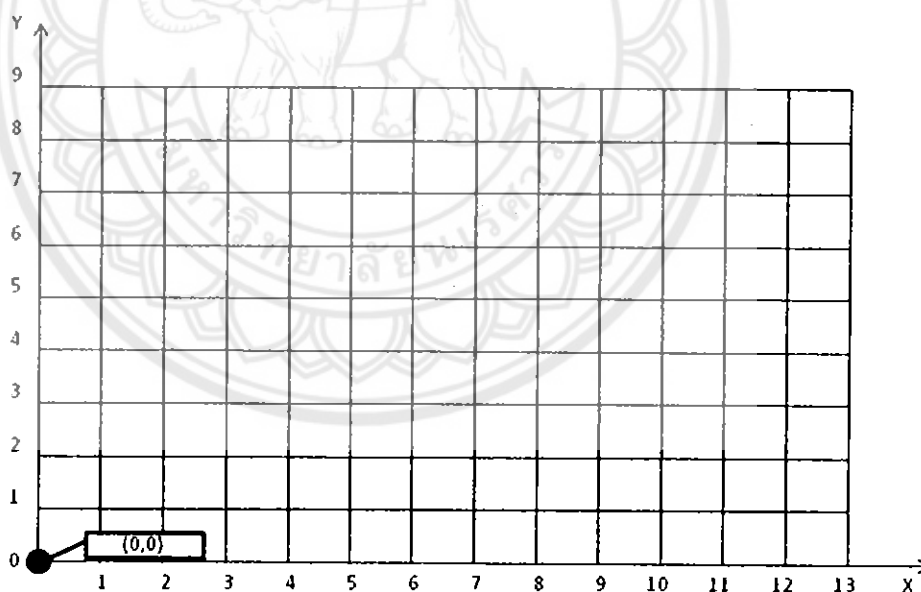
รูปที่ 3.9 ตารางเส้นทางเดินของมด

2. เมื่อนำค่าสมาชิกใน \tilde{Y} มาจุดลงในตารางจะได้เส้นทางอ้างอิงดังรูปที่ 3.10



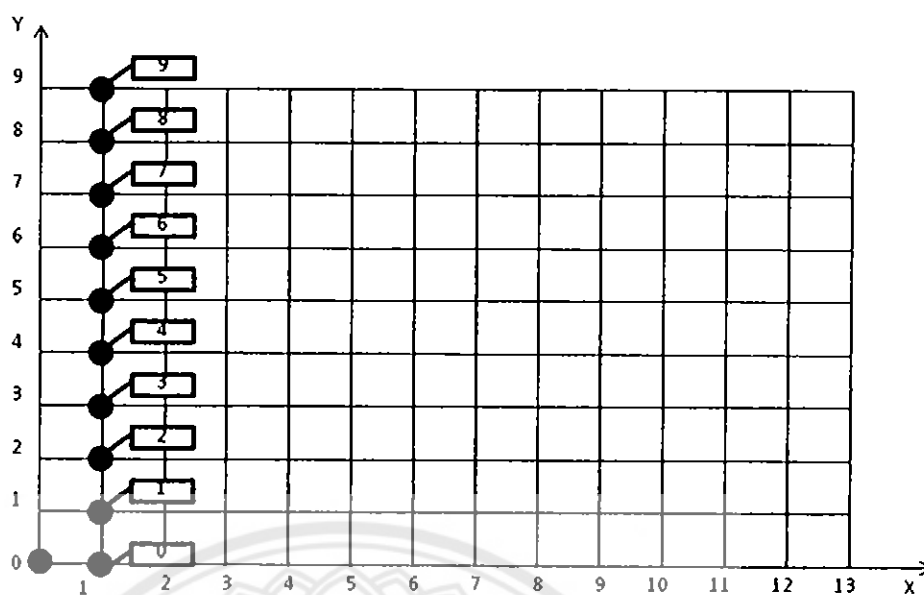
รูปที่ 3.10 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 1

3. ให้มิตตัวแรกอยู่ที่ตำแหน่ง $(0,0)$ ซึ่งเป็นจุดเริ่มต้นของทุกเส้นทาง



รูปที่ 3.11 จุดเริ่มต้นของเส้นทาง

4. ที่ตำแหน่งถัดไปเมื่อ $x=1$ มีความเป็นไปได้ที่มิตจะเดินไปยังตำแหน่ง $y = 0, 1, \dots, 9$ ซึ่งมีทั้งหมด 10 กรณี ดังรูปที่ 3.12



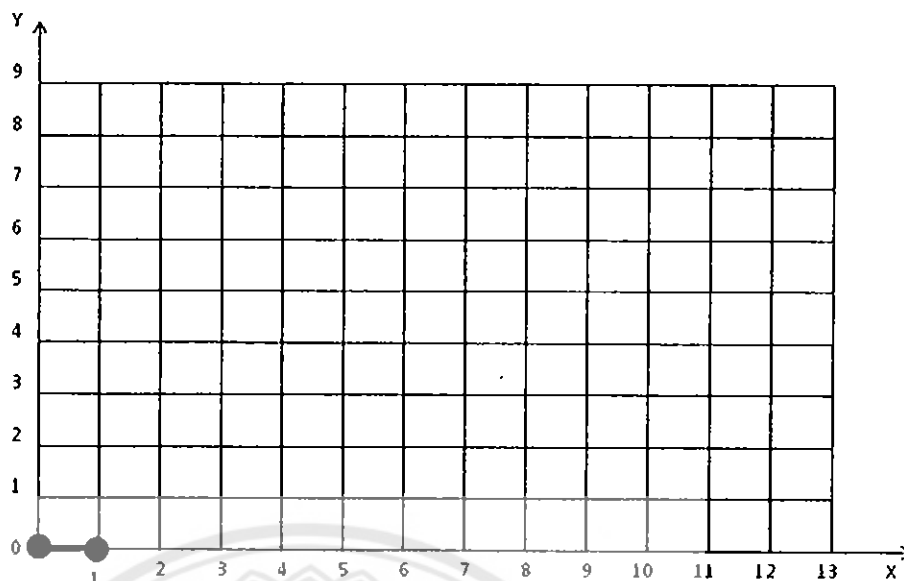
รูปที่ 3.12 ความน่าจะเป็นที่มดจะเลือก ณ จุด $x=1$

4.1 เลือก $q^* = 0.9$ แล้วทำการสุ่มค่า q_j แบบเอกภาพ จากนั้นคำนวณหาค่าความน่าจะเป็นตามสมการที่ 3.11 จะได้

ตารางที่ 3.2 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=1$ ของรอบที่ 1

j'	q_j	$\eta_1(1, j')$	$\tau_1(1, j')$	$P_1(1, j')$	หมายเหตุ
0	0.6944	1	1	0.1818	$q_0 < q^*$
1	0.2124	0.9	1	0.1636	$q_1 < q^*$
2	0.5433	0.8	1	0.1455	$q_2 < q^*$
3	0.7025	0.7	1	0.1273	$q_3 < q^*$
4	0.9564	0.6	1	0	$q_4 > q^*$
5	0.4445	0.5	1	0.0909	$q_5 < q^*$
6	0.0854	0.4	1	0.0727	$q_6 < q^*$
7	0.0573	0.3	1	0.0545	$q_7 < q^*$
8	0.6295	0.2	1	0.0364	$q_8 < q^*$
9	0.7962	0.1	1	0.0182	$q_9 < q^*$

จากตารางที่ 3.2 พบว่าที่ $j' = 0$ มีความน่าจะเป็นมากที่สุด ดังนั้นมดจึงเลือกเดินจากตำแหน่ง $(0,0)$ ไปยัง $(1,0)$ ดังแสดงไว้ในรูปที่ 3.13



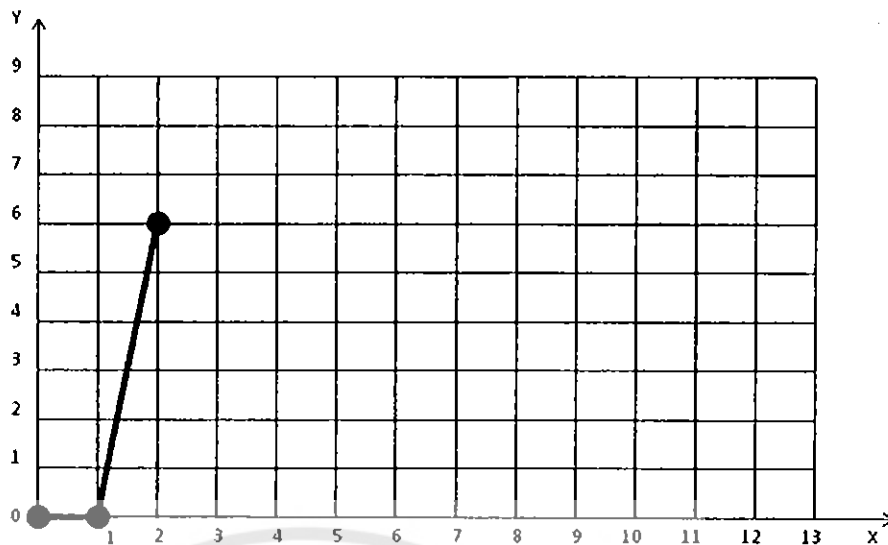
รูปที่ 3.13 เส้นทางการเดินของมดช่วงที่ 1 ของมดตัวที่ 1 ในรอบที่ 1

4.2 เนื่องจากมดยังเดินไม่ถึงจุดสุดท้ายจึงต้องคำนวณหาตำแหน่งถัดไป ซึ่งมีความเป็นไปได้ 10 กรณีเช่นกัน ทำการสุ่มค่า q_j แบบเอกภาพ แล้วคำนวณหาความน่าจะเป็น $P_1(2, j')$ จะได้ตามตารางที่ 3.3

ตารางที่ 3.3 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=2$ ของรอบที่ 1

j'	q_j	$\eta_1(2, j')$	$\tau_1(2, j')$	$P_1(2, j')$	หมายเหตุ
0	0.6991	0.4	1	0.0548	$q_0 < q^*$
1	0.7963	0.5	1	0.0685	$q_1 < q^*$
2	0.4416	0.6	1	0.0822	$q_2 < q^*$
3	0.4462	0.7	1	0.0959	$q_3 < q^*$
4	0.4657	0.8	1	0.1096	$q_4 < q^*$
5	0.2790	0.9	1	0.1233	$q_5 < q^*$
6	0.6754	1	1	0.1370	$q_6 < q^*$
7	0.9037	0.9	1	0	$q_7 > q^*$
8	0.9085	0.8	1	0	$q_8 > q^*$
9	0.7472	0.7	1	0.0959	$q_9 < q^*$

จากตารางที่ 3.3 พบว่าที่ $j'=6$ มีความน่าจะเป็นมากที่สุด ดังนั้นมดจึงเลือกเส้นทางจากตำแหน่ง (1,0) ไปยัง (2,6) ดังแสดงไว้ในรูปที่ 3.14



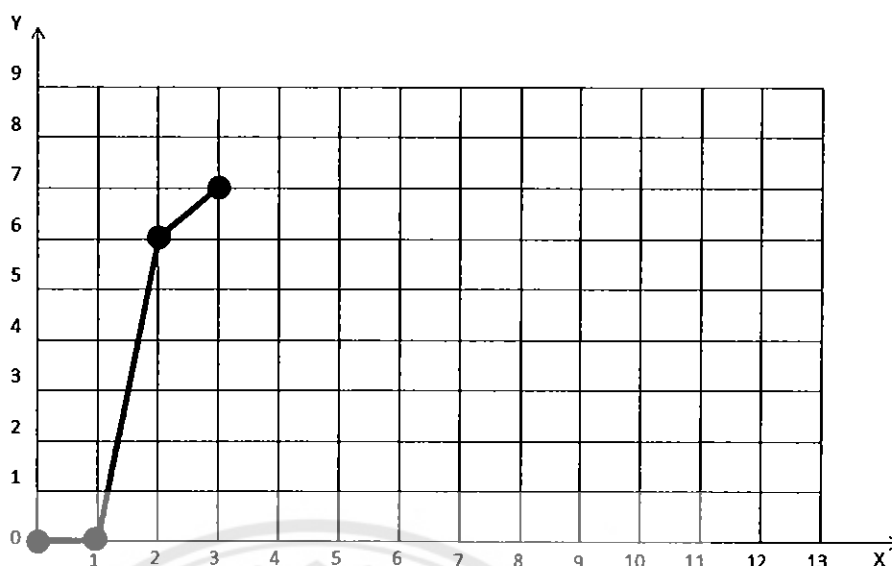
รูปที่ 3.14 เส้นทางการเดินทางของมดช่วงที่ 2 ของมดตัวที่ 1 ในรอบที่ 1

4.3 จากรูปที่ 3.14 มดยังเดินไม่ถึงจุดสุดท้าย จึงคำนวณหาตำแหน่งถัดไปตามหัวข้อ 4.2 จะได้ตามตารางที่ 3.4

ตารางที่ 3.4 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=3$ ของรอบที่ 1

j'	q_j	$\eta_1(3, j')$	$\tau_1(3, j')$	$P_1(3, j')$	หมายเหตุ
0	0.8147	0.3	1	0.0435	$q_0 < q^*$
1	0.6058	0.4	1	0.0580	$q_1 < q^*$
2	0.1270	0.5	1	0.0725	$q_2 < q^*$
3	0.0134	0.6	1	0.0870	$q_3 < q^*$
4	0.6324	0.7	1	0.1014	$q_4 < q^*$
5	0.0975	0.8	1	0.1159	$q_5 < q^*$
6	0.2785	0.9	1	0.1304	$q_6 < q^*$
7	0.5469	1	1	0.1449	$q_7 < q^*$
8	0.9575	0.9	1	0	$q_8 > q^*$
9	0.1576	0.8	1	0.1159	$q_9 < q^*$

จากตารางที่ 3.4 พบว่าที่ $j'=7$ มีความน่าจะเป็นมากที่สุด ดังนั้นมดจึงเลือกเส้นทางจากตำแหน่ง (2,6) ไปยัง (3,7) ดังแสดงไว้ในรูปที่ 3.15



รูปที่ 3.15 เส้นทางการเดินทางของมดช่วงที่ 3 ของมดตัวที่ 1 ในรอบที่ 1

5. ทำซ้ำไปจนกระทั่งมดตัวแรกเดินครบทุกค่า x จากนั้นเริ่มปล่อยมดตัวที่ 2, 3, 4 จนครบมดทุกตัว ในตัวอย่างนี้ เลือกใช้มด 10 ตัว ซึ่งเส้นทางของมดแต่ละตัวแสดงไว้ในตารางที่ 3.5

ตารางที่ 3.5 การบันทึกเส้นทางของมดแต่ละตัวในรอบที่ 1

มดตัวที่	เส้นทาง
1	{0,6,7,9,0,1,5,2,0,0,0,7,6}
2	{0,5,7,8,0,1,5,1,0,0,0,7,6}
3	{0,6,7,8,0,1,5,2,1,0,0,7,6}
4	{0,6,7,7,0,1,5,2,0,0,1,7,6}
5	{1,6,7,8,0,1,5,2,0,0,0,7,6}
6	{0,6,7,8,0,1,5,2,0,0,0,7,6}
7	{0,6,6,8,0,1,5,2,0,0,0,7,6}
8	{1,6,7,8,0,0,5,2,0,0,0,7,6}
9	{0,6,7,8,0,1,4,2,1,0,0,7,6}
10	{0,6,7,8,0,1,5,2,0,0,1,7,6}

6. ทำการแปลความหมายของแต่ละเส้นทางให้เป็นค่าพารามิเตอร์ K_p , K_i และ K_d ที่สอดคล้องกับมดแต่ละตัวดังนี้

ตารางที่ 3.6 ค่า K_p , K_i และ K_d ที่ได้จากมดแต่ละตัวในรอบที่ 1

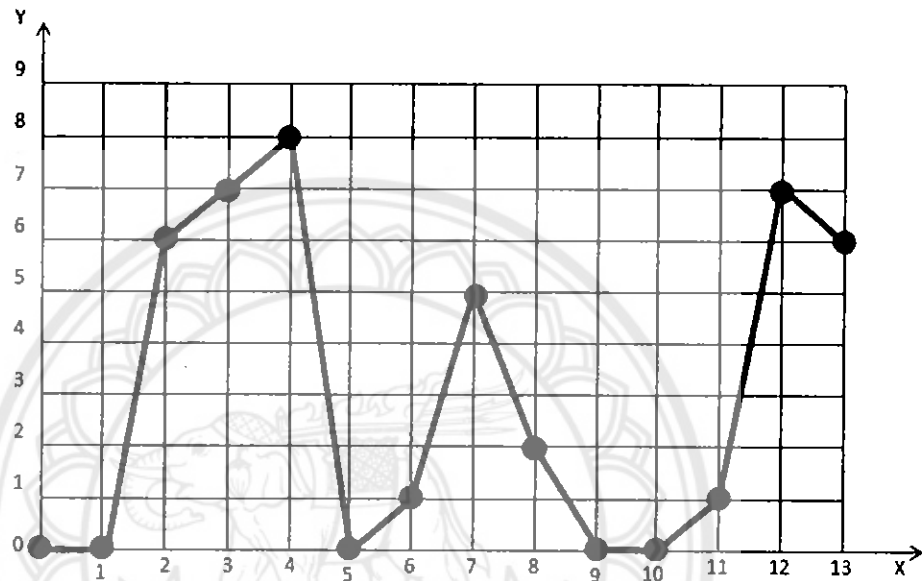
มดตัวที่	K_p	K_i	K_d
1	6.79	15.20	0.76
2	5.78	15.10	0.76
3	6.78	15.21	0.76
4	6.77	15.20	1.76
5	16.78	15.20	0.76
6	6.78	15.20	0.76
7	6.68	15.20	0.76
8	16.78	5.20	0.76
9	6.78	14.21	0.76
10	6.78	15.20	1.76

จากนั้นนำค่า K_p , K_i และ K_d มาคำนวณหาตัวควบคุมแบบ PID แล้วทำการจำลองผลตอบสนองของระบบ นำผลตอบสนองที่ได้มาคำนวณหาค่าความคลาดเคลื่อนตามสมการที่ 3.14 ซึ่งจะได้ความคลาดเคลื่อนของมดแต่ละตัวดังนี้

ตารางที่ 3.7 การคำนวณค่าความคลาดเคลื่อนในรอบที่ 1

มดตัวที่	ค่าความคลาดเคลื่อน
1	0.1820
2	0.1744
3	0.1820
4	0.0592
5	4.4571
6	0.1818
7	0.1797
8	1.8773
9	0.1653
10	0.0590

7. เนื่องจากมดตัวที่ 10 ให้ค่าความคลาดเคลื่อนต่ำที่สุด จึงนำเส้นทางของมดตัวที่ 10 มาเปรียบเทียบกับเส้นทางอ้างอิงเริ่มต้น แต่เนื่องจากเส้นทางอ้างอิงเริ่มต้นมีค่าความคลาดเคลื่อนเป็น 0.1776 ซึ่งมีค่ามากกว่าเมื่อเทียบกับความคลาดเคลื่อนอันเกิดจากมดตัวที่ 10 ดังนั้นจึงให้เส้นทางของมดตัวที่ 10 เป็นเส้นทางอ้างอิงที่จะใช้ในรอบถัดไป พร้อมทำการปรับค่าฟีโรโมนตามสมการที่ 3.15 โดยกำหนดให้ $\rho = 0.9$



รูปที่ 3.16 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 2

8. จากรูปที่ 3.16 จะนำเส้นทางของมดตัวที่ 10 มาทำการปรับฟีโรโมนจะได้ดังตารางที่ 3.8

ตารางที่ 3.8 ตำแหน่งที่ทำการปรับพีโรโมนในรอบที่ 1

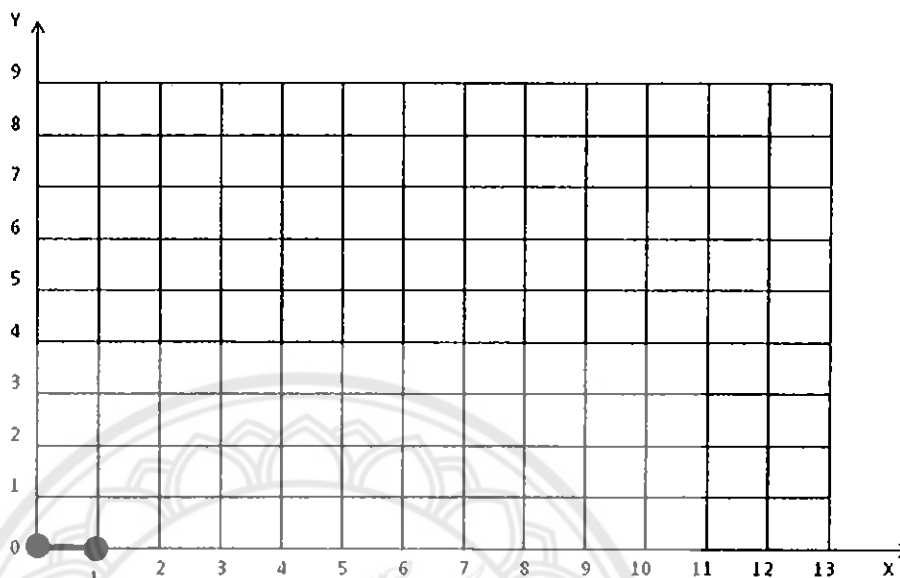
ตำแหน่ง	$\tau(i, j)$ เดิม	$\Delta\tau(i, j)$	$\tau(i, j)$ ใหม่
(1,0)	1	16.9492	15.3543
(2,6)	1	16.9492	15.3543
(3,7)	1	16.9492	15.3543
(4,8)	1	16.9492	15.3543
(5,0)	1	16.9492	15.3543
(6,1)	1	16.9492	15.3543
(7,5)	1	16.9492	15.3543
(8,2)	1	16.9492	15.3543
(9,0)	1	16.9492	15.3543
(10,0)	1	16.9492	15.3543
(11,1)	1	16.9492	15.3543
(12,7)	1	16.9492	15.3543
(13,6)	1	16.9492	15.3543

9. ในรอบถัดมาทำการปล่อยมดตัวแรกที่ตำแหน่ง (0,0) จากนั้นทำการสุ่มค่า q_j พร้อมคำนวณหาความน่าจะเป็น $P_1(1, j')$ จะได้ตามตารางที่ 3.9

ตารางที่ 3.9 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x=1$ ของรอบที่ 2

j'	q_j	$\eta_1(1, j')$	$\tau_1(1, j')$	$P_1(1, j')$	หมายเหตุ
0	0.4491	1	15.3543	0.7733	$q_0 < q^*$
1	0.1535	0.9	1	0.0453	$q_1 < q^*$
2	0.2112	0.8	1	0.0403	$q_2 < q^*$
3	0.7003	0.7	1	0.0353	$q_3 < q^*$
4	0.6355	0.6	1	0.0302	$q_4 < q^*$
5	0.9562	0.5	1	0	$q_5 > q^*$
6	0.8913	0.4	1	0.0201	$q_6 < q^*$
7	0.2163	0.3	1	0.0151	$q_7 < q^*$
8	0.6835	0.2	1	0.0101	$q_8 < q^*$
9	0.3151	0.1	1	0.0050	$q_9 < q^*$

เนื่องจากที่ $j' = 0$ มีความน่าจะเป็นมากที่สุด จึงให้มุดเดินจาก $(0, 0)$ ไปยัง $(1, 0)$ ดังแสดงในรูปที่ 3.17



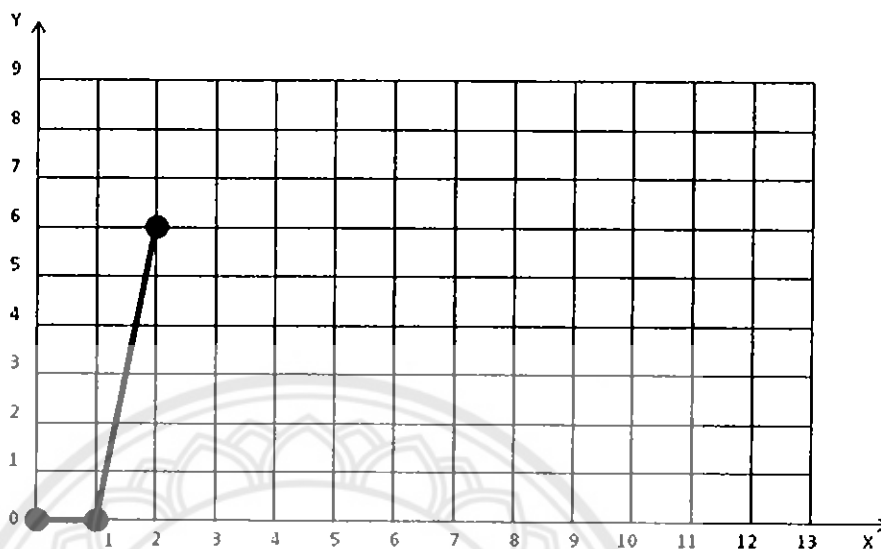
รูปที่ 3.17 เส้นทางเดินของมุดช่วงที่ 1 ของมุดตัวที่ 1 ในรอบที่ 2

10. ขั้นตอนถัดไปให้หาความน่าจะเป็นที่มุดจะเดินจากตำแหน่ง $(1, 0)$ ไปยังตำแหน่งถัดไป ซึ่งจะได้ดังตารางที่ 3.10

ตารางที่ 3.10 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x = 2$ ของรอบที่ 2

j'	$q_{j'}$	$\eta_1(2, j')$	$\tau_1(2, j')$	$P_1(2, j')$	หมายเหตุ
0	0.8202	0.4	1	0.0185	$q_0 < q^*$
1	0.4327	0.5	1	0.0231	$q_1 < q^*$
2	0.0887	0.6	1	0.0277	$q_2 < q^*$
3	0.9351	0.7	1	0	$q_3 > q^*$
4	0.3908	0.8	1	0.0369	$q_4 < q^*$
5	0.5879	0.9	1	0.0416	$q_5 < q^*$
6	0.2398	1	15.3543	0.7090	$q_6 < q^*$
7	0.6536	0.9	1	0.0416	$q_7 < q^*$
8	0.3665	0.8	1	0.0369	$q_8 < q^*$
9	0.4812	0.7	1	0.0323	$q_9 < q^*$

จากตารางที่ 3.10 พบว่าที่ $j' = 6$ มีความน่าจะเป็นมากที่สุด ดังนั้นจึงเลือกเส้นทางจากตำแหน่ง (1,0) ไปยัง (2,6) ดังแสดงไว้ในรูปที่ 3.18



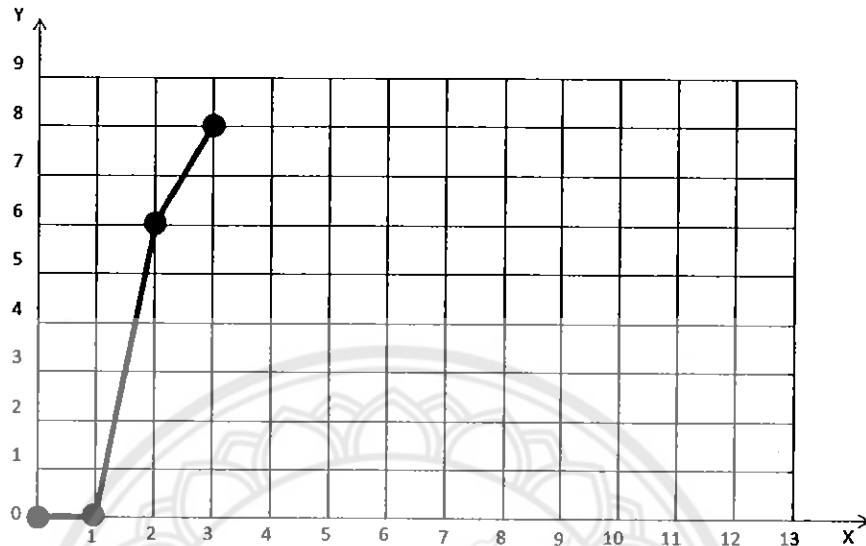
รูปที่ 3.18 เส้นทางเดินของมดช่วงที่ 2 ของมดตัวที่ 1 ในรอบที่ 2

11. ขั้นตอนถัดไปให้หาความน่าจะเป็นที่มดจะเดินจากตำแหน่ง (2,6) ไปยังตำแหน่งถัดไปซึ่งจะได้อะไร

ตารางที่ 3.11 การคำนวณความน่าจะเป็น ณ ตำแหน่ง $x = 3$ ของรอบที่ 2

j'	q_j	$\eta_1(3, j')$	$\tau_1(3, j')$	$P_1(3, j')$	หมายเหตุ
0	0.3112	0.3	1	0.0141	$q_0 < q^*$
1	0.2290	0.4	1	0.0188	$q_1 < q^*$
2	0.0046	0.5	1	0.0235	$q_2 < q^*$
3	0.1818	0.6	1	0.0282	$q_3 < q^*$
4	0.3510	0.7	1	0.0329	$q_4 < q^*$
5	0.6684	0.8	1	0.0376	$q_5 < q^*$
6	0.9027	0.9	1	0	$q_6 > q^*$
7	0.9212	1	15.3543	0	$q_7 > q^*$
8	0.7447	0.9	1	0.0423	$q_8 < q^*$
9	0.4868	0.8	1	0.0376	$q_9 < q^*$

จากตารางที่ 3.11 พบว่าที่ $j' = 8$ มีความน่าจะเป็นมากที่สุด ดังนั้นมดจึงเลือกเส้นทางจากตำแหน่ง (2,6) ไปยัง (3,8) ดังแสดงไว้ในรูปที่ 3.19



รูปที่ 3.19 เส้นทางเดินของมดช่วงที่ 3 ของมดตัวที่ 1 ในรอบที่ 2

12. เมื่อมดเดินครบ 10 ตัวแล้วจะได้เส้นทางของมดแต่ละตัวดังนี้

ตารางที่ 3.12 การบันทึกเส้นทางของมดแต่ละตัวในรอบที่ 2

มดตัวที่	เส้นทาง
1	{0, 6, 8, 7, 0, 2, 4, 4, 2, 0, 0, 7, 9}
2	{0, 6, 8, 7, 0, 1, 5, 4, 2, 0, 1, 7, 9}
3	{0, 6, 8, 7, 1, 1, 5, 4, 2, 0, 1, 7, 8}
4	{0, 6, 8, 7, 0, 0, 5, 4, 2, 0, 1, 7, 9}
5	{0, 6, 8, 7, 0, 1, 5, 4, 2, 0, 1, 8, 9}
6	{0, 6, 8, 7, 0, 1, 5, 4, 2, 0, 1, 7, 9}
7	{0, 6, 6, 7, 0, 1, 5, 4, 2, 1, 0, 7, 9}
8	{0, 6, 8, 7, 0, 1, 5, 4, 1, 0, 1, 7, 9}
9	{0, 6, 8, 7, 1, 1, 5, 4, 1, 0, 0, 7, 9}
10	{0, 6, 9, 7, 0, 1, 5, 4, 2, 0, 1, 7, 9}

ซึ่งเส้นทางที่ได้สอดคล้องกับค่า K_p , K_i และ K_d คือ

ตารางที่ 3.13 ค่า K_p , K_i และ K_d ที่ได้จากมดแต่ละตัวในรอบที่ 2

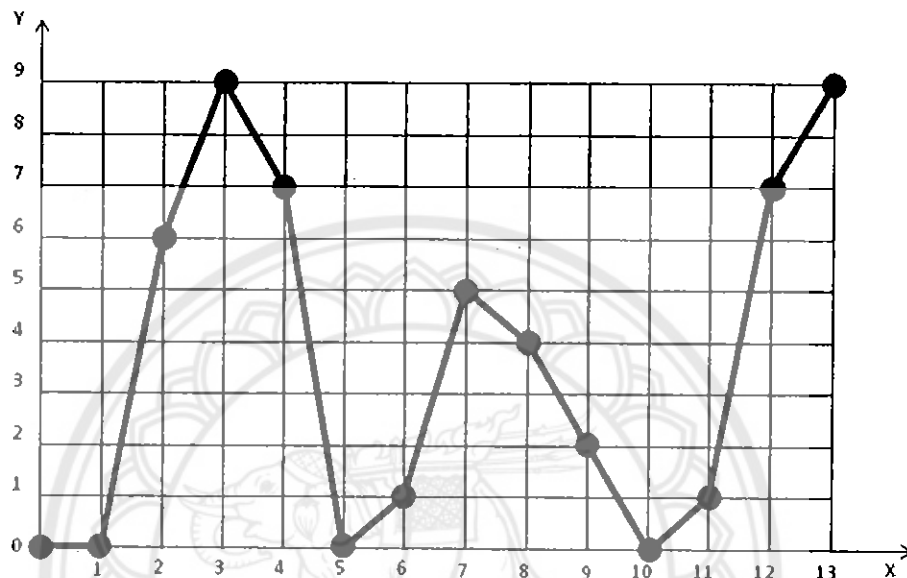
มด(ตัวที่)	K_p	K_i	K_d
1	6.87	24.42	0.79
2	6.87	15.42	1.79
3	6.87	115.42	1.78
4	6.87	5.42	1.79
5	6.87	15.42	1.89
6	6.87	15.42	1.79
7	6.68	15.42	10.79
8	6.87	15.41	1.79
9	6.87	115.41	0.79
10	6.97	15.42	1.79

13. จากนั้นนำค่า K_p , K_i และ K_d มาคำนวณหาตัวควบคุมแบบ PID แล้วทำการจำลองผลตอบสนองของระบบ นำผลตอบสนองที่ได้มาคำนวณหาค่าความคลาดเคลื่อนตามสมการที่ 3.14 ซึ่งจะได้ความคลาดเคลื่อนของมดแต่ละตัวดังนี้

ตารางที่ 3.14 การคำนวณค่าความคลาดเคลื่อนในรอบที่ 2

มดตัวที่	ค่าความคลาดเคลื่อน
1	0.5218
2	0.0582
3	4.0875e+009
4	0.1136
5	0.0588
6	0.0582
7	0.3874
8	0.0582
9	5.0084e+009
10	0.0561

14. เนื่องจากมดตัวที่ 10 ให้ค่าความคลาดเคลื่อนต่ำที่สุด จึงนำเส้นทางของมดตัวที่ 10 มาเปรียบเทียบกับเส้นทางอ้างอิง แต่เนื่องจากเส้นทางอ้างอิงมีค่าความคลาดเคลื่อนเป็น 0.0590 ซึ่งมีค่ามากกว่าเมื่อเทียบกับความคลาดเคลื่อนอันเกิดจากมดตัวที่ 10 ดังนั้นจึงให้เส้นทางของมดตัวที่ 10 เป็นเส้นทางอ้างอิงที่จะใช้ในรอบถัดไป พร้อมทำการปรับค่าฟีโรโมนตามสมการที่ 3.15



รูปที่ 3.20 เส้นทางอ้างอิงเพื่อใช้คำนวณในรอบที่ 3

15. จากรูปที่ 3.20 นำเส้นทางของมดตัวที่ 10 มาทำการปรับฟีโรโมนจะได้ดังตารางที่ 3.15

ตารางที่ 3.15 ตำแหน่งที่ทำการปรับฟีโรโมนในรอบที่ 2

ตำแหน่ง	$\tau(i, j)$ เดิม	$\Delta\tau(i, j)$	$\tau(i, j)$ ใหม่
(1,0)	15.3543	17.8253	17.5782
(2,6)	15.3543	17.8253	17.5782
(3,9)	1	17.8253	16.1428
(4,7)	1	17.8253	16.1428
(5,0)	15.3543	17.8253	17.5782
(6,1)	15.3543	17.8253	17.5782
(7,5)	15.3543	17.8253	17.5782
(8,4)	1	17.8253	16.1428
(9,2)	1	17.8253	16.1428

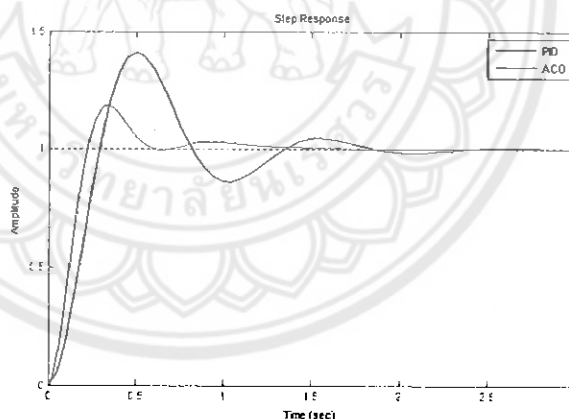
ตารางที่ 3.16 (ต่อ) ตำแหน่งที่ทำการปรับพีโรโมนในรอบที่ 2

ตำแหน่ง	$\tau(i, j)$ เดิม	$\Delta\tau(i, j)$	$\tau(i, j)$ ใหม่
(10,0)	15.3543	17.8253	17.5782
(11,1)	15.3543	17.8253	17.5782
(12,7)	15.3543	17.8253	17.5782
(13,9)	1	17.8253	16.1428

16. ทำการวนซ้ำจนครบจำนวนรอบ หรือได้ค่าความคลาดเคลื่อนอยู่ในช่วงที่ยอมรับได้ โดยผลที่ได้จากการปรับพารามิเตอร์ด้วยวิธีของ ACS คือ $K_p = 7.99$, $K_i = 13.03$ และ $K_d = 1.74$ ซึ่งได้ตัวควบคุมแบบ PID ดังนี้

$$G_c(s) = 1.74s + 7.99 + \frac{13.03}{s} = \frac{1.74s^2 + 7.99s + 13.03}{s}$$

และผลตอบสนองที่ได้แสดงไว้ในรูปที่ 3.21



รูปที่ 3.21 ผลตอบสนองของระบบที่ใช้ ACS เมื่อเทียบกับการออกแบบด้วยวิธี ZN

จากรูปที่ 3.21 เห็นได้ว่าผลตอบสนองของระบบที่ผ่านการปรับปรุงพารามิเตอร์ด้วยขั้นตอนวิธีของ ACS มีผลตอบสนองดีกว่าการออกแบบตัวควบคุม PID แบบ ZN

3.3 ขั้นตอนการออกแบบโปรแกรม GUI

ในหัวข้อนี้นำเสนอขั้นตอนการออกแบบส่วนประสานกราฟิกกับผู้ใช้ (GUI) โดยมีขั้นตอนการออกแบบดังนี้

ขั้นที่ 1 วิเคราะห์ปัญหา

ปัญหาการปรับปรุงค่าพารามิเตอร์ K_p , K_i และ K_d ของตัวควบคุมแบบ PID ในหัวข้อที่ 3.1 และ 3.2 ได้เลือกใช้ ACS มาใช้ในการแก้ไขปัญห

ขั้นที่ 2 ข้อมูลที่ป้อนสู่โปรแกรม (Input)

การนำ ACS มาประยุกต์ใช้ในการปรับปรุงพารามิเตอร์ของตัวควบคุมแบบ PID ต้องมีการรับข้อมูล คือ จำนวนรอบ จำนวนมด ค่าความเชื่อในเส้นทางอ้างอิง ค่าฟีโรโมนเริ่มต้น ค่าโร ค่าถ่วงน้ำหนักฟีโรโมน ค่าถ่วงน้ำหนักความเชื่อในเส้นทางอ้างอิง ค่าพารามิเตอร์ K_p , K_i , K_d และระบบ ซึ่งโปรแกรมที่ถูกพัฒนาขึ้นจะรับข้อมูลเฉพาะค่าพารามิเตอร์ K_p , K_i , K_d และระบบโดยสามารถป้อนระบบได้ทั้งในรูปแบบของฟังก์ชันถ่ายโอน และปริภูมิสถานะ ในส่วนค่าพารามิเตอร์อื่นๆโปรแกรมที่พัฒนาขึ้นได้กำหนดค่าไว้ดังนี้

ตารางที่ 3.17 ค่าพารามิเตอร์เริ่มต้นของโปรแกรมสำเร็จ

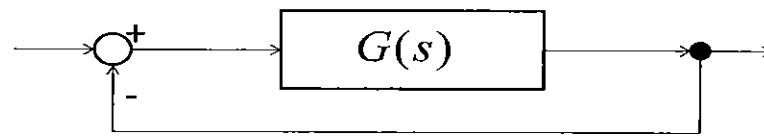
พารามิเตอร์	ค่าที่กำหนดไว้
จำนวนรอบ	100
จำนวนมด	30
ค่าความเชื่อในเส้นทางอ้างอิง	0.98
ค่าฟีโรโมนเริ่มต้น	1
ค่าโร	0.9
ค่าถ่วงน้ำหนักฟีโรโมน	1
ค่าถ่วงน้ำหนักความเชื่อในเส้นทางอ้างอิง	1

ขั้นที่ 3 ผลลัพธ์ของโปรแกรม (Output)

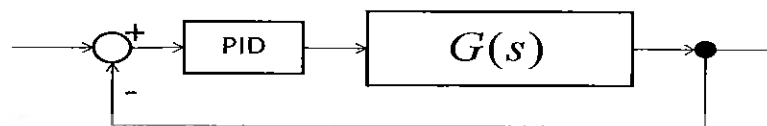
ผลที่ได้จากโปรแกรมคือค่าพารามิเตอร์ K_p , K_i และ K_d ที่ผ่านการปรับปรุงแล้ว และแสดงผลในรูปแบบของกราฟโดยแบ่งกราฟเป็น 2 รูปคือ

1. กราฟของระบบป้อนกลับแบบหนึ่งเพื่อใช้ในการวิเคราะห์ระบบที่ป้อนเข้าสู่โปรแกรมว่ามีผลตอบสนองของระบบมีเสถียรภาพหรือไม่มีเสถียรภาพ
2. กราฟของระบบที่ทำการออกแบบตัวควบคุมแบบ PID โดยในรูปนี้เป็นการเปรียบเทียบระหว่างกราฟของระบบที่ทำการออกแบบตัวควบคุมแบบ PID ทั้งก่อนและหลังการปรับปรุงด้วย ACS

ซึ่งกราฟทั้ง 2 รูปที่ได้ คือ ผลตอบสนองของระบบที่แสดงดังรูปที่ 3.22 และ 3.23 ตามลำดับ



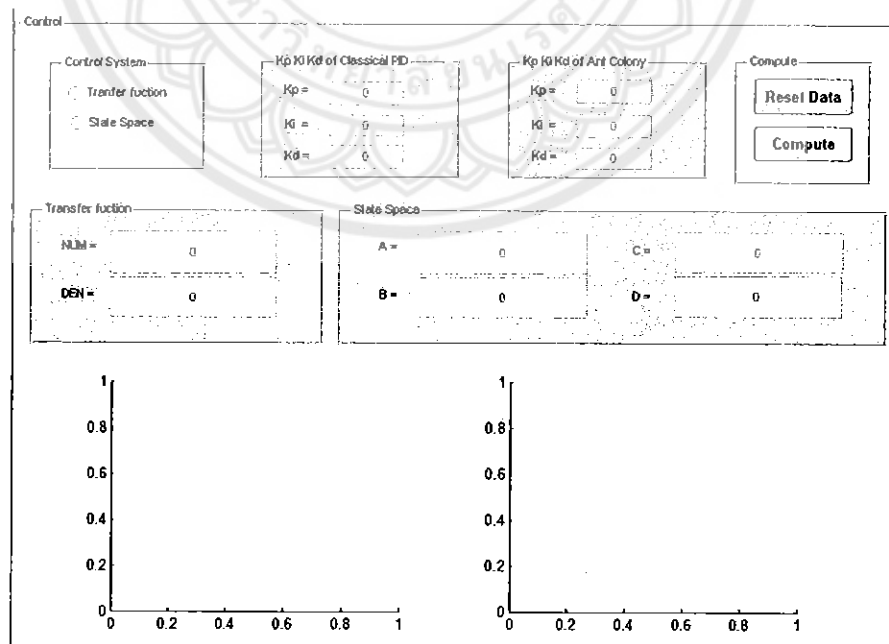
รูปที่ 3.22 แผนภาพกล่องของระบบป้อนกลับแบบหนึ่งที่ไม่มีตัวควบคุม



รูปที่ 3.23 แผนภาพกล่องของระบบป้อนกลับแบบหนึ่งที่มีตัวควบคุมแบบ PID

ขั้นที่ 4 ทำการออกแบบหน้าจอของโปรแกรม

ในส่วนหน้าจอของโปรแกรมควรประกอบไปด้วยข้อมูลที่ป้อนเข้าสู่โปรแกรม นั่นคือระบบที่สามารถป้อนเข้าสู่โปรแกรมได้ทั้งในรูปแบบของฟังก์ชันถ่ายโอน หรือปริภูมิสถานะ และค่าพารามิเตอร์ K_p , K_i , K_d ส่วนผลลัพธ์ของโปรแกรมประกอบไปด้วยค่าพารามิเตอร์ K_p , K_i และ K_d ที่ถูกปรับปรุงด้วย ACS กราฟของระบบป้อนกลับแบบหนึ่ง และกราฟของระบบป้อนกลับแบบหนึ่งที่มีตัวควบคุมแบบ PID ดังรูปที่ 3.24



รูปที่ 3.24 หน้าจอของโปรแกรม

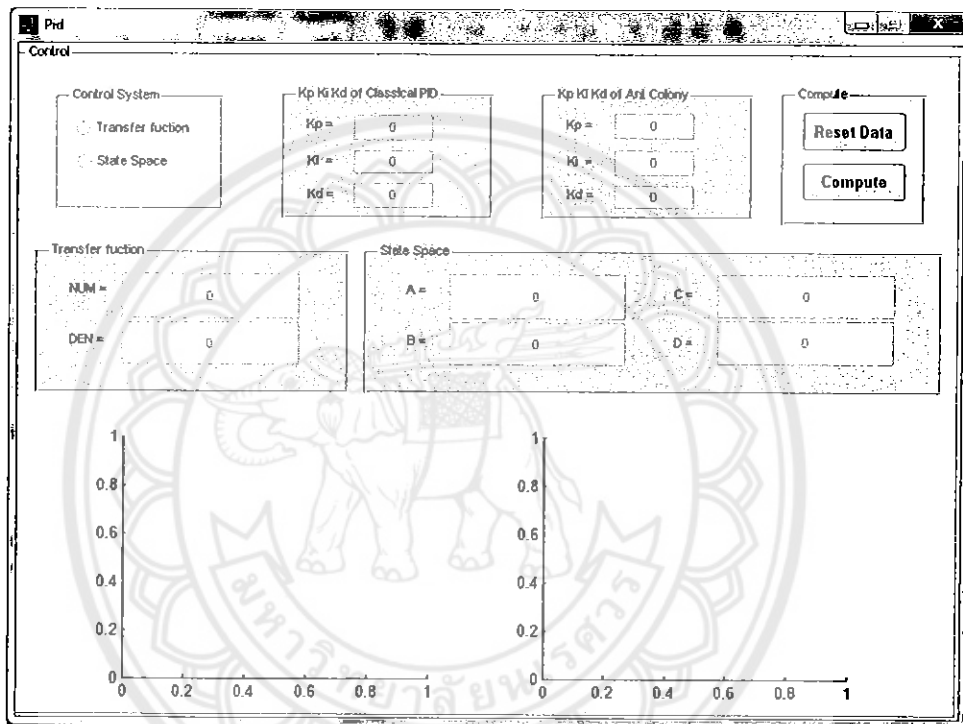
จากรูปที่ 3.24 เมื่อกดปุ่ม Compute โปรแกรมจะทำปรับปรุ่งค่าพารามิเตอร์ K_p , K_i และ K_d ด้วยขั้นตอนวิธี ACS จากนั้นโปรแกรมจะแสดงกราฟของผลตอบสนอง โดยกราฟทางรูปซ้ายมือจะแสดงผลตอบสนองของระบบป้อนกลับแบบหนึ่ง และกราฟทางรูปขวามือจะแสดงผลตอบสนองของระบบป้อนกลับแบบหนึ่งที่มีตัวควบคุมแบบ PID ทั้งก่อนและหลังการปรับปรุ่งด้วย ACS



บทที่ 4

ผลการทดลอง

ในบทนี้เป็นการนำทฤษฎีในบทที่ 2 และแนวคิดในการนำ ACS มาประยุกต์ใช้ในการแก้ไขปัญหาการปรับค่าพารามิเตอร์ PID ในบทที่ 3 มาทำการทดลองกับระบบควบคุมแบบต่างๆ ซึ่งมีผลการทดลองดังต่อไปนี้



รูปที่ 4.1 หน้าต่างโปรแกรมปรับพารามิเตอร์ด้วยขั้นตอนวิธีของ ACS

4.1 ผลการทดลองกับระบบที่ 1

ระบบที่ 1 เป็นระบบของการควบคุมการทรงตัวของจักรยานหุ่นยนต์ซึ่งเขียนอยู่ในรูปแบบของปริภูมิสถานะ $\dot{X} = Ax + Bu$ และ $y = Cx + Du$

โดยที่

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 15.1485 & 0 & 1.3397 & 0 \\ 0 & -545.8682 & -0.015 & 312.0943 \\ 0 & 0 & -4769.2 & -4615.4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3846.2 \end{bmatrix}, C = [1 \ 0 \ 0 \ 0], D = 0$$

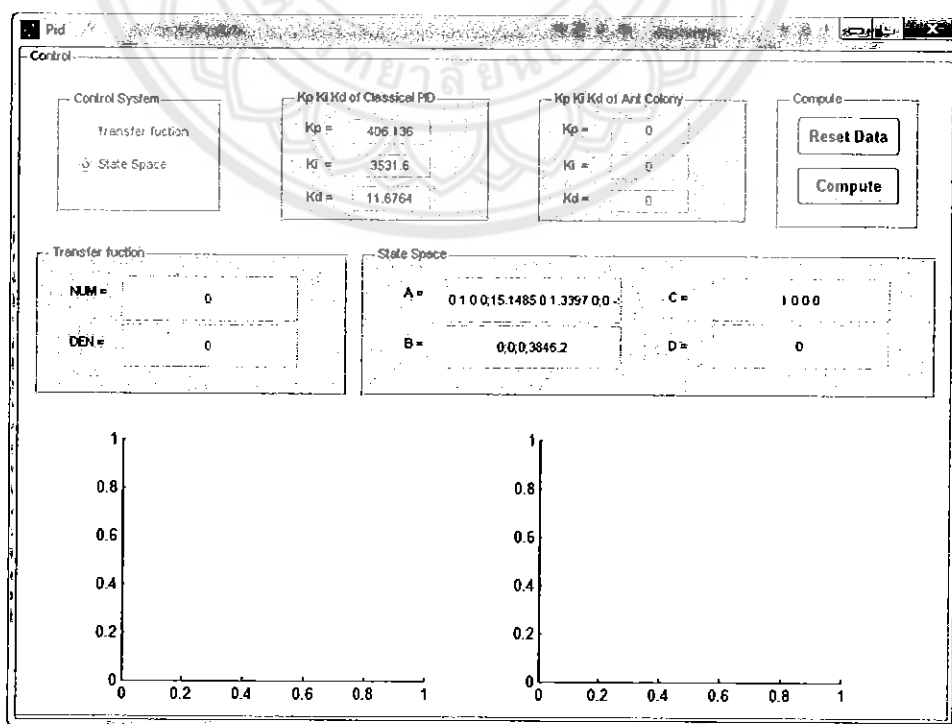
ทำการออกแบบตัวควบคุมแบบ PID ซึ่งจะได้พารามิเตอร์ K_p , K_i และ K_d ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 1 ด้วยวิธี ZN

พารามิเตอร์	K_p	K_i	K_d
ค่าพารามิเตอร์ PID	406.1360	3531.6	11.6764

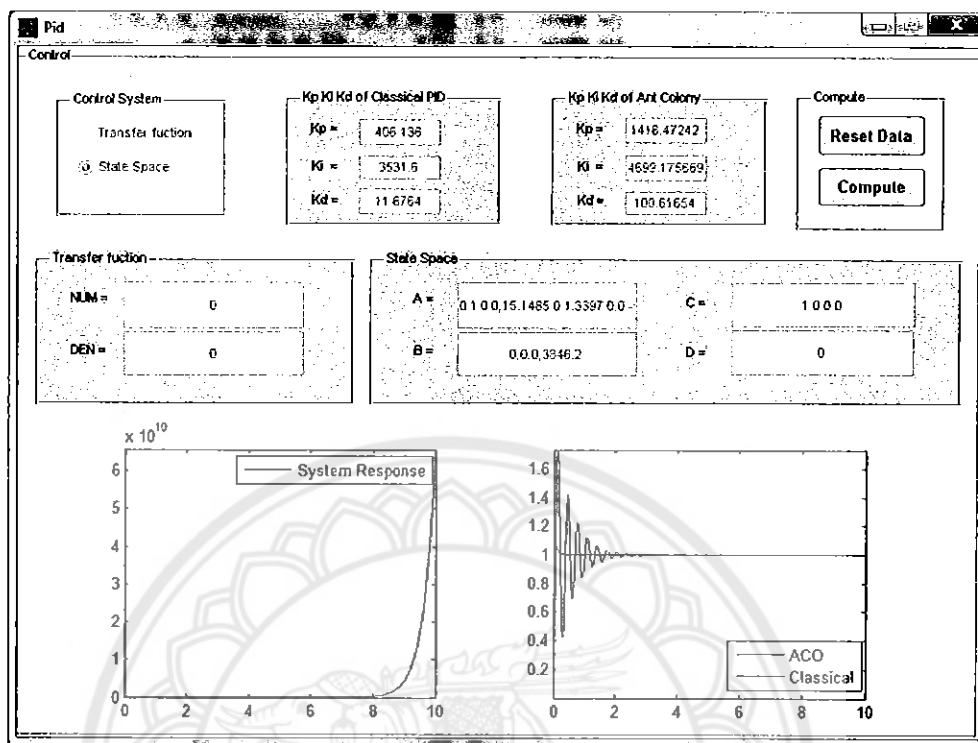
ในการปรับปรุงพารามิเตอร์ของตัวควบคุมแบบ PID โดยใช้โปรแกรมที่พัฒนาขึ้นมานั้น มีขั้นตอนดังนี้

1. กดปุ่มเลือก State Space ในหัวข้อ Control System แล้วทำการกรอกค่า A, B, C และ D ลงในหัวข้อ State Space
2. นำค่า K_p , K_i และ K_d ใส่ลงในช่อง Kp Ki Kd of Classical PID ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการป้อนข้อมูลลงโปรแกรมของระบบที่ 1

3. ให้ทำการเลือกปุ่ม “Compute” ซึ่งผลที่ได้จะเป็นไปตามรูปที่ 4.3



รูปที่ 4.3 แสดงผลลัพธ์โปรแกรมของระบบที่ 1

พิจารณารูปในรูปที่ 4.3 กราฟทางด้านซ้ายมือแสดงผลตอบสนองของระบบที่มีการป้อนกลับแบบหนึ่ง ซึ่งพบว่าผลตอบสนองมีค่าเข้าใกล้อนันต์แสดงว่าระบบป้อนกลับแบบหนึ่งเป็นระบบที่ไม่เสถียร ในขณะที่กราฟทางด้านขวามือแสดงผลตอบสนองของระบบวงปิดที่มีการควบคุมด้วยตัวควบคุมแบบ PID (เป็นสี่เหลี่ยม) พบว่าระบบมีเสถียรภาพแต่ผลตอบสนองมีการแกว่งกวัดมาก และมีค่าพุ่งสูงสุดค่อนข้างสูง แต่เมื่อนำพารามิเตอร์ K_p , K_i และ K_d มาปรับปรุงด้วยขั้นตอนวิธี ACS แล้วพบว่าผลตอบสนองดีขึ้นมาก (ดูกราฟเส้นสีน้ำเงิน)

ตารางที่ 4.2 ผลการทดลองของระบบที่ 1

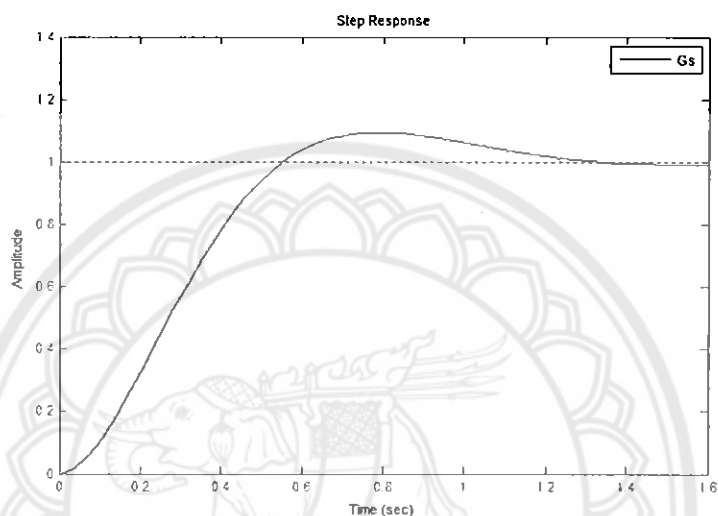
พารามิเตอร์	K_p	K_i	K_d
ก่อนการปรับปรุง	406.1360	3531.6	11.6764
หลังการปรับปรุง	1418.47242	4699.17566	100.61654

4.2 ผลการทดลองกับระบบที่ 2

ระบบที่ 2 เป็นระบบอันดับสองซึ่งนำมาทดลองใช้ในโครงการนี้คือ

$$G(s) = \frac{25}{s^2 + 6s + 25}$$

เมื่อป้อนสัญญาณแบบขั้นบันไดหนึ่งหน่วยจะได้ผลตอบสนองที่แสดงในรูปที่ 4.4 ซึ่งจะพบว่าผลตอบสนองที่ได้ค่อนข้างช้า



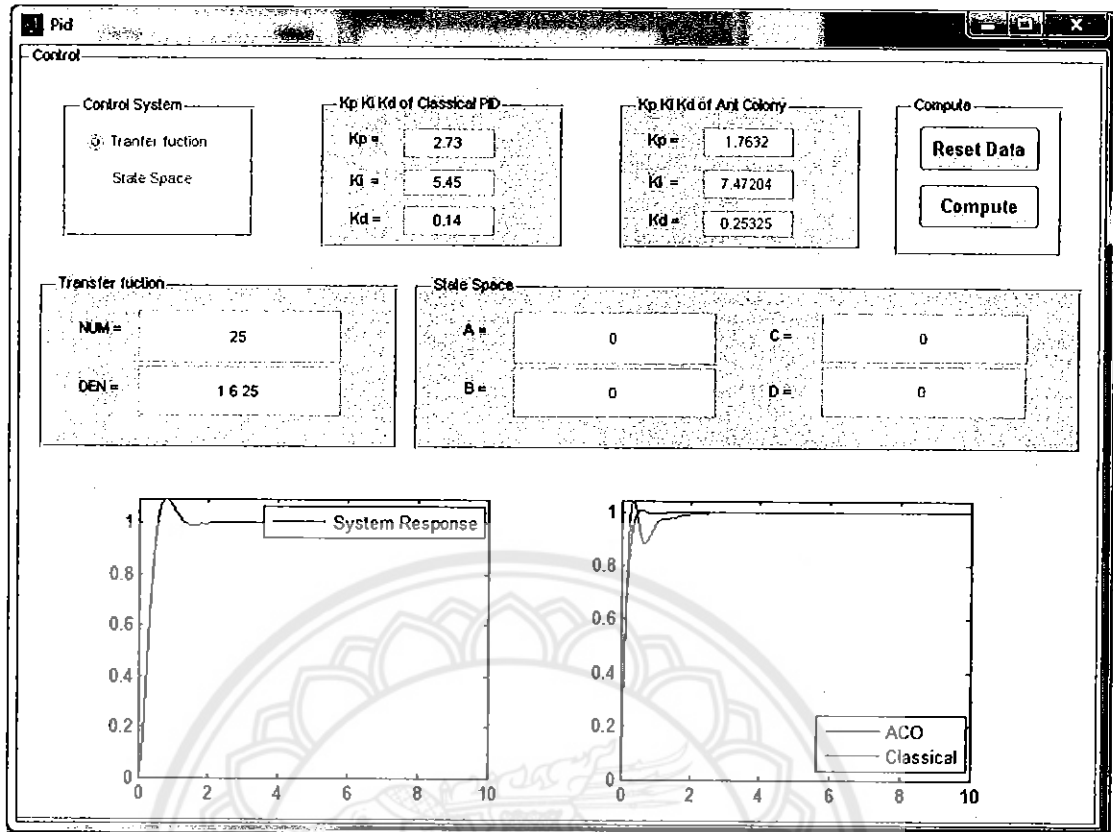
รูปที่ 4.4 ผลตอบสนองของระบบที่ 2

เมื่อทำการออกแบบตัวควบคุม PID ด้วยวิธีของ CHR (Least Overshoot) จะได้ว่าค่า $K = 1.1$, $L = 0.1$ และ $T = 0.5$ เมื่อนำมาคำนวณโดยใช้ตารางที่ 2.4 หน้า 18 จะได้พารามิเตอร์ของตัวควบคุมแบบ PID ตามตารางที่ 4.3

ตารางที่ 4.3 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 2

พารามิเตอร์	K_p	K_i	K_d
ค่าพารามิเตอร์ PID	2.73	5.45	0.14

นำค่า K_p , K_i และ K_d ที่ได้ป้อนลงในโปรแกรมสำเร็จพร้อมทั้งเลือกประเภทของระบบเป็นแบบ Transfer Function แล้วทำการจำลองหาผลตอบสนองจะได้ภาพดังรูปที่ 4.5



รูปที่ 4.5 แสดงผลลัพธ์ โปรแกรมของระบบที่ 2

จากรูปที่ 4.4 พบว่าผลตอบสนองมีเสถียรภาพแล้วแต่ผลตอบสนองเข้าสู่สมมูลค่อนข้างช้า จึงนำระบบมาป้อนกลับแบบหนึ่งซึ่งพบว่าผลตอบสนองของระบบเข้าสู่สมมูลเร็วขึ้น แต่เวลาขึ้นของระบบยังช้าอยู่ดังกราฟในรูปที่ 4.5 ทางด้านซ้ายมือ ในขณะที่กราฟทางด้านขวามือแสดงผลตอบสนองของระบบวงปิดที่มีการควบคุมด้วยตัวควบคุมแบบ PID (กราฟสีเขียว) พบว่าระบบยังเข้าสู่สมมูลค่อนข้างช้าอยู่ แต่เมื่อทำการปรับปรุงด้วยขั้นตอนวิธีของ ACS แล้วพบว่าผลตอบสนองของระบบดีขึ้นมาก ดังกราฟ (สีน้ำเงิน) ทางด้านขวามือในรูปที่ 4.5

ตารางที่ 4.4 ผลการทดลองของระบบที่ 2

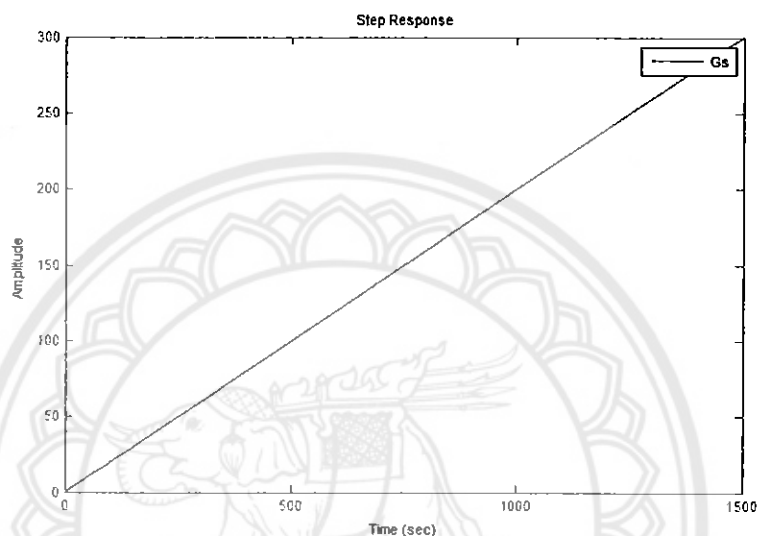
พารามิเตอร์	K_p	K_i	K_d
ก่อนการปรับปรุง	2.73	5.45	0.14
หลังการปรับปรุง	1.7632	7.47204	0.25325

4.3 ผลการทดลองกับระบบที่ 3

พิจารณาระบบอันดับ 3

$$G(s) = \frac{1}{s^3 + 6s^2 + 5s}$$

ซึ่งผลตอบสนองระบบดังกล่าวเมื่อป้อนสัญญาณเข้าเป็นแบบฟังก์ชันขั้นบันไดหนึ่งหน่วยแสดงได้ดังรูปที่ 4.6



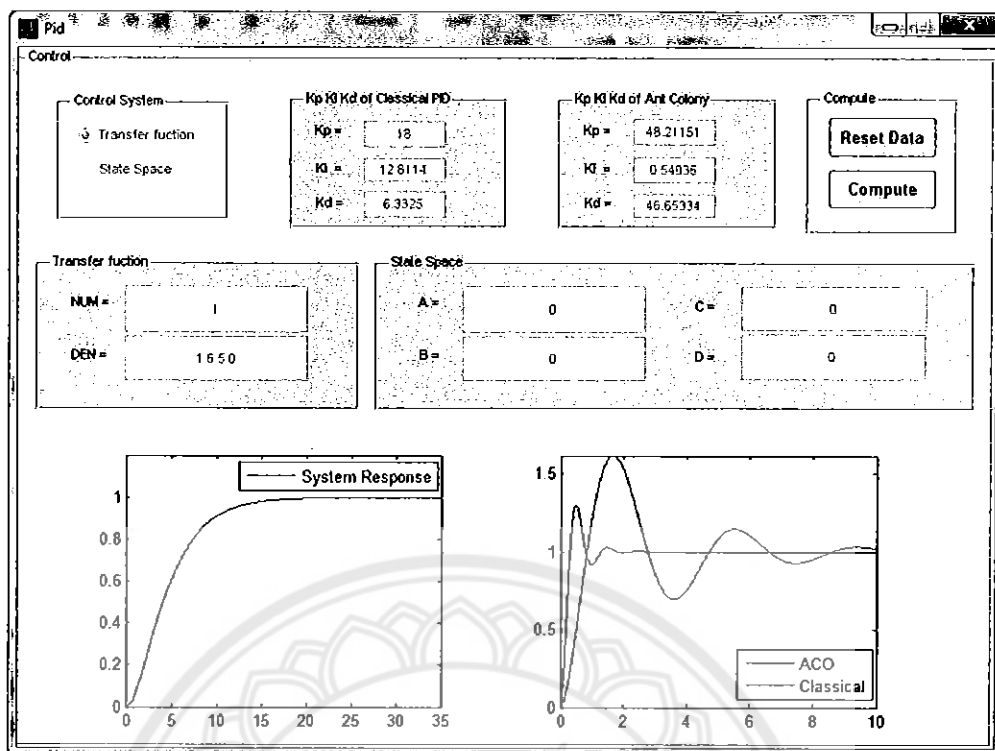
รูปที่ 4.6 ผลตอบสนองของระบบที่ 3

เนื่องจากระบบไม่มีเสถียรภาพจึงทำการออกแบบตัวควบคุมแบบ PID ด้วยวิธีของ ZN (วิธีที่ 2) ซึ่งได้พารามิเตอร์ของ PID ตามตารางที่ 4.5

ตารางที่ 4.5 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 3

พารามิเตอร์	K_p	K_i	K_d
ค่าพารามิเตอร์ PID	18	12.8114	6.3325

เมื่อป้อนพารามิเตอร์ K_p , K_i และ K_d รวมทั้ง $G(s)$ ลงโปรแกรมสำเร็จ จะได้ผลตอบสนองตามรูปที่ 4.7



รูปที่ 4.7 แสดงผลลัพธ์โปรแกรมของระบบที่ 3

จากรูปที่ 4.6 จะเห็นได้ผลตอบสนองของระบบเริ่มต้น ไม่มีเสถียรภาพ ดังนั้นจึงนำระบบมาเข้าโปรแกรมสำเร็จดังรูปที่ 4.7 พิจารณารูปทางซ้ายมือพบว่าผลตอบสนองของระบบที่มีการป้อนกลับแบบหนึ่งมีเสถียรภาพ แต่เข้าสู่สภาวะสมดุลค่อนข้างช้า ส่วนกราฟทางด้านขวามือนั้นแสดงผลตอบสนองของระบบที่มีตัวควบคุม PID (กราฟสีเขียว) เข้าสู่สภาวะสมดุลเร็วขึ้น แต่ผลตอบสนองมีการแกว่งกวัดมาก และมีค่าพุ่งสูงสุดที่สูงอยู่ ในขณะที่ผลตอบสนองของระบบที่ผ่านการปรับปรุง PID ด้วยขั้นตอนวิธีของ ACS มีลักษณะที่ดีขึ้นอย่างชัดเจน

ตารางที่ 4.6 ผลการทดลองของระบบที่ 3

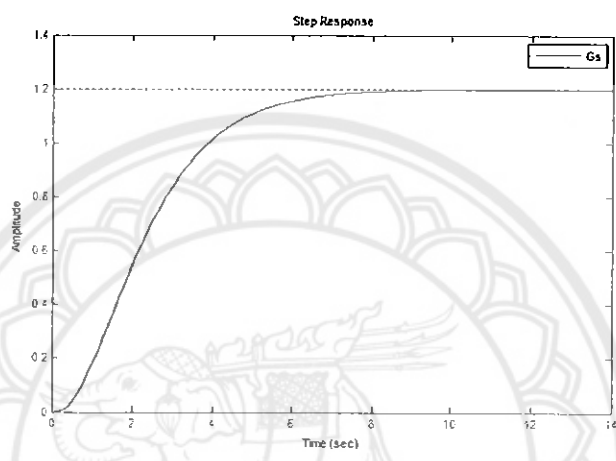
พารามิเตอร์	K_p	K_i	K_d
ก่อนการปรับปรุง	18	12.8114	6.3325
หลังการปรับปรุง	48.21151	0.54036	46.65334

4.4 ผลการทดลองกับระบบที่ 4

พิจารณาระบบอันดับ 3

$$G(s) = \frac{1.2}{0.36s^3 + 1.86s^2 + 2.5s + 1}$$

เมื่อป้อนสัญญาณแบบฟังก์ชันขั้นบันไดหน่วยจะได้ผลตอบสนองดังรูปที่ 4.8



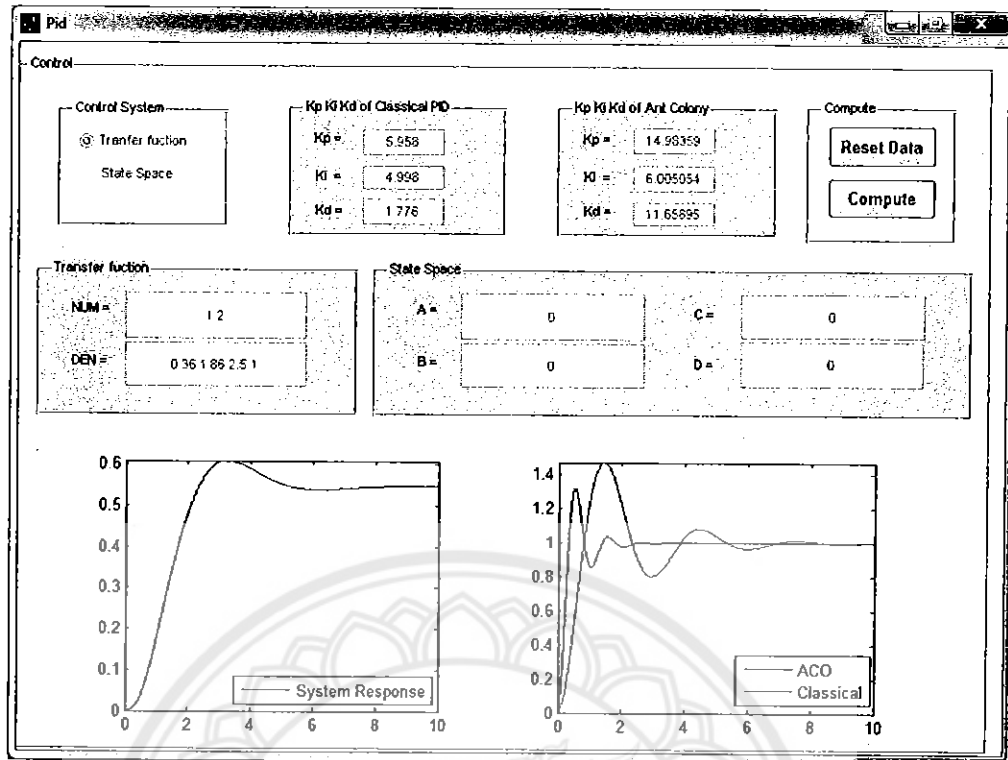
รูปที่ 4.8 ผลตอบสนองของระบบที่ 4

จากรูปที่ 4.8 จะเห็นได้ว่าระบบมีผลตอบสนองค่อนข้างช้าเนื่องจากเวลาขึ้นมีค่ามากเกินไป ในที่นี้จึงเลือกใช้วิธีการออกแบบตัวควบคุมแบบ PID ด้วยวิธี CHR (20% Overshoot) เพื่อให้ระบบดังกล่าวมีผลตอบสนองที่ดีขึ้น โดยค่าพารามิเตอร์ของ PID คำนวณได้ตามตารางที่ 4.7

ตารางที่ 4.7 ค่าพารามิเตอร์ PID ที่คำนวณได้จากระบบที่ 4

พารามิเตอร์	K_p	K_i	K_d
ค่าพารามิเตอร์ PID	5.958	4.998	1.776

นำค่า K_p , K_i และ K_d ที่ได้มาปรับปรุงด้วยขั้นตอนวิธี ACS โดยใช้โปรแกรมสำเร็จ จะได้ผลตอบสนองตามรูปที่ 4.9



รูปที่ 4.9 แสดงผลลัพธ์โปรแกรมของระบบที่ 4

พิจารณารูปในรูปที่ 4.9 จะเห็นว่าผลตอบสนองทางซ้ายมือยังมีค่าความผิดพลาดที่สถานะคงตัวอยู่ และเวลาเข้าสู่สมดุลค่อนข้างช้า ส่วนกราฟทางด้านขวามือแสดงผลตอบสนองของระบบที่มีการควบคุมแบบ PID (กราฟสีเขียว) พบว่าระบบไม่มีค่าผิดพลาดที่สถานะคงตัว และเวลาขึ้นเร็วขึ้น แต่ผลตอบสนองยังเข้าสู่สมดุลค่อนข้างช้า แต่เมื่อผ่านการปรับปรุงด้วยขั้นตอนวิธีของ ACS แล้วทำให้ผลตอบสนองของระบบดีขึ้นอย่างชัดเจนดังกราฟสีน้ำเงินทางขวามือ

ตารางที่ 4.8 ผลการทดลองของระบบที่ 4

พารามิเตอร์	K_p	K_i	K_d
ก่อนการปรับปรุง	5.958	4.998	1.776
หลังการปรับปรุง	14.98359	6.00505	11.65895

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ในโครงการนี้ได้พัฒนาโปรแกรมส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) โดยสามารถป้อนระบบที่สนใจได้ทั้งในรูปแบบของฟังก์ชันถ่ายโอนและรูปแบบปฏิกิริยาสถานะ และป้อนค่าของ K_p , K_i และ K_d ในการออกแบบตัวควบคุมแบบ PID จากนั้นโปรแกรมจึงนำพารามิเตอร์ทั้งสามมาปรับปรุงด้วยขั้นตอนวิธีแบบอาณานิคมมด (ACS) เพื่อให้ผลตอบสนองของระบบควบคุมดีขึ้น เมื่อนำโปรแกรม GUI มาทดสอบกับระบบตัวอย่างทั้ง 4 ระบบพบว่าระบบมีผลตอบสนองที่ดีขึ้น โดยมีเวลาขึ้นเวลาสูงสุด ค่าพุ่งสูงสุด และเวลาสู่สมดุล ดีขึ้นกว่าการออกแบบระบบควบคุมแบบ PID แบบดั้งเดิม

5.2 ข้อเสนอแนะและแนวทางการพัฒนาโครงการ

จากโครงการการออกแบบตัวควบคุมแบบ PID ด้วยขั้นตอนวิธีแบบอาณานิคมมด (Ant Colony System: ACS) ซึ่งให้ค่าที่เหมาะสมที่สุด ผลลัพธ์ที่ได้ทำให้ผู้ใช้พอใจอยู่ในระดับหนึ่ง ผู้จัดทำโครงการจึงมีข้อเสนอแนะและแนวทางการพัฒนาต่อดังนี้

- 5.2.1 โปรแกรมสามารถคำนวณค่า K_p , K_i และ K_d จากระบบได้โดยไม่ต้องให้ผู้ใช้คำนวณด้วยตัวเอง
- 5.2.2 สามารถนำขั้นตอนวิธีแบบอาณานิคมมดมาแก้ไขปัญหาอื่นๆ เช่น ปัญหาการจัดตารางสอน ปัญหาการวางแผนเครื่องจักรในโรงงาน และปัญหาการเดินทางของพนักงานขาย เป็นต้น
- 5.2.3 สามารถนำขั้นตอนวิธีอื่นๆ ของการค้นหาแบบศึกษาสำนึก เช่น การค้นหาแบบกลไกคัดสรรพันธุที่ดีตามธรรมชาติ (Genetic Algorithm: GA) การค้นหาแบบการสลัดของกบ (Shuffled Frog Leaping Algorithm: SFL) การค้นหาแบบการจับกลุ่มของอนุภาค (Particle Swarm Optimization: PSO) และ การค้นหาแบบทาบู (Taboo Search: TS) เป็นต้น มาใช้ในการแก้ไข ปัญหาการปรับค่าพารามิเตอร์ของตัวควบคุมแบบ PID แล้วนำผลตอบสนองที่ได้จากขั้นตอนวิธีต่างๆมาเปรียบเทียบกัน

เอกสารอ้างอิง

- [1] ธนิต มาลากร. (2552). เอกสารประกอบคำสอน รายวิชาวิศวกรรมระบบควบคุม. พิษณุโลก.
- [2] Brain R Copeland. (2551). **The Design of PID Controllers using Ziegler Nichols Tuning.** สืบค้นเมื่อ 13 ธันวาคม 2553, จาก <http://www.eng.uwi.tt>
- [3] Hong He, Fang Liu, Li Li, Jin-Rong Yang, Lei Su, Yi Wu. (2549). **Study of PID Control System for Ant Colony Algorithm.** สืบค้นเมื่อ 8 ธันวาคม 2553, จาก <http://doi.ieeecomputersociety.org/10.1109/GCIS.2009.137>
- [4] JihGauJuang, Yi Cheng Lu. (2550). **Application of Ant Colony System to an Experimental Propeller Setup.** สืบค้นเมื่อ 13 ธันวาคม 2553, จาก <http://academic.research.microsoft.com/Publication/4275249/application-of-ant-colony-system-to-an-experimental-propeller-setup>
- [5] Katsuhiko Ogata. (2545). **Modern Control Engineering.** New Jersey: Prentice Hall.
- [6] ZoranVukic. (2545). **Lectures on PID Controllers.** สืบค้นเมื่อ 30 มกราคม 2554, จาก http://arri.uta.edu/acs/jyotirmay/EE4343/Labs_Projects/pidcontrollers.pdf



ภาคผนวก ก วิธีการใช้งานโปรแกรม

จากการทำโครงการเรื่องการปรับปรุงตัวควบคุม PID ด้วยขั้นตอนวิธีอานานิคมมด ผู้ศึกษาได้ทำการเขียน โปรแกรมขึ้นมาเพื่อสะดวกในการคำนวณต่อผู้ใช้งาน ซึ่งมีวิธีการ ใช้งาน โปรแกรม ดังขั้นตอนต่อไปนี้

1. ทำการดับเบิ้ลคลิกที่ไฟล์ Setup ดังรูปที่ ก.1

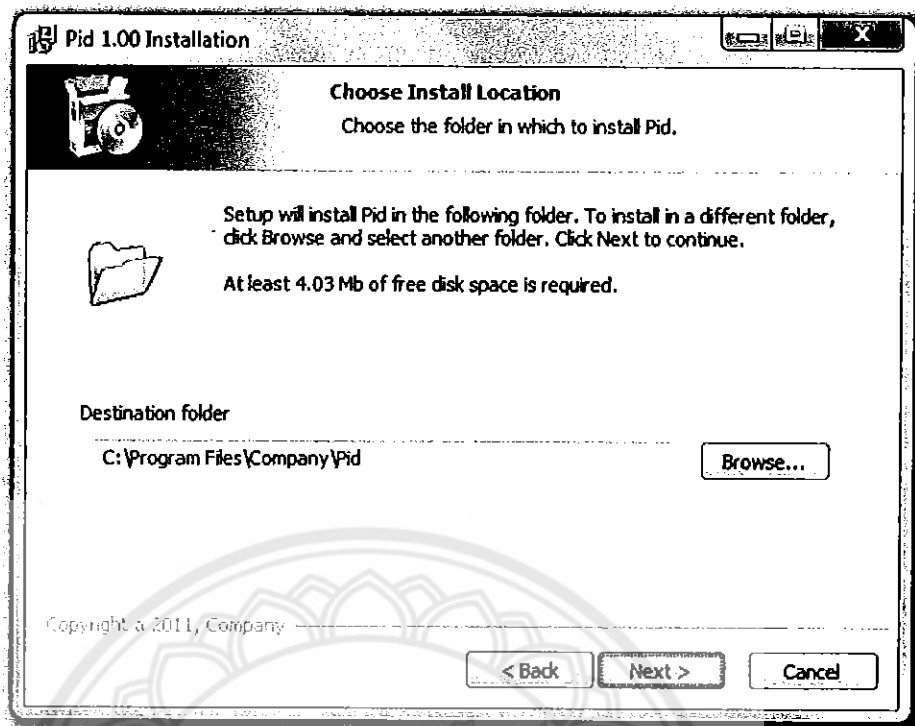
Name	Date modified	Type	Size
 Setup	20/3/2554 23:06	Application	3,875 KB

รูปที่ ก.1 ไฟล์ Setup

2. จากนั้นจะขึ้นหน้าต่างดังรูปที่ ก.2 แล้วเลือก Next แล้วทำการเลือก โฟลเดอร์ที่ต้องการติดตั้งโปรแกรมแล้วเลือก Next ดังรูปที่ ก.3

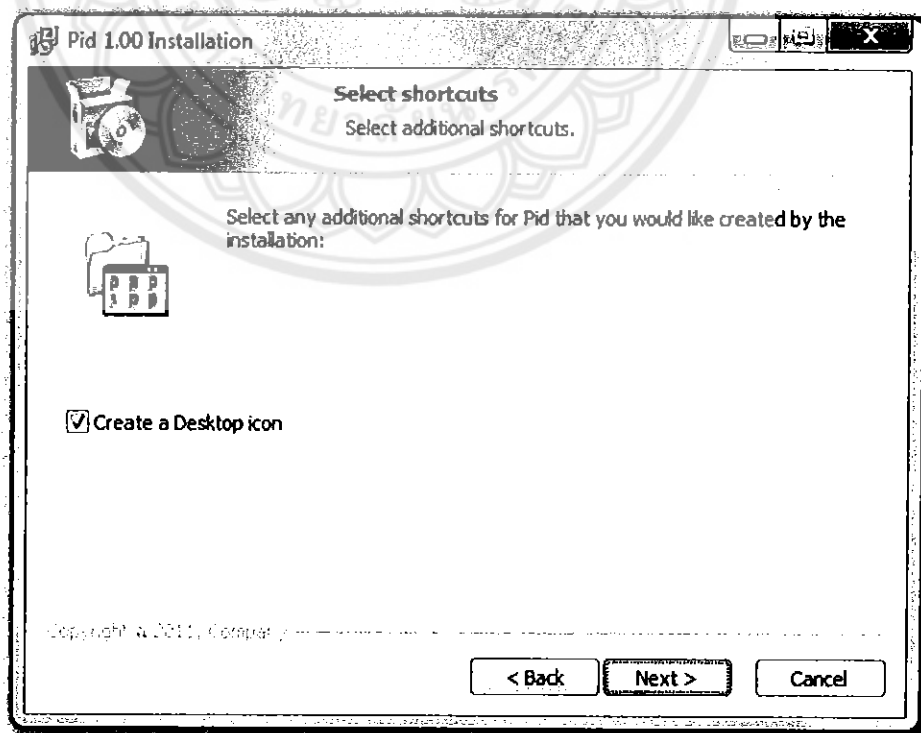


รูปที่ ก.2 หน้าแรกของโปรแกรม

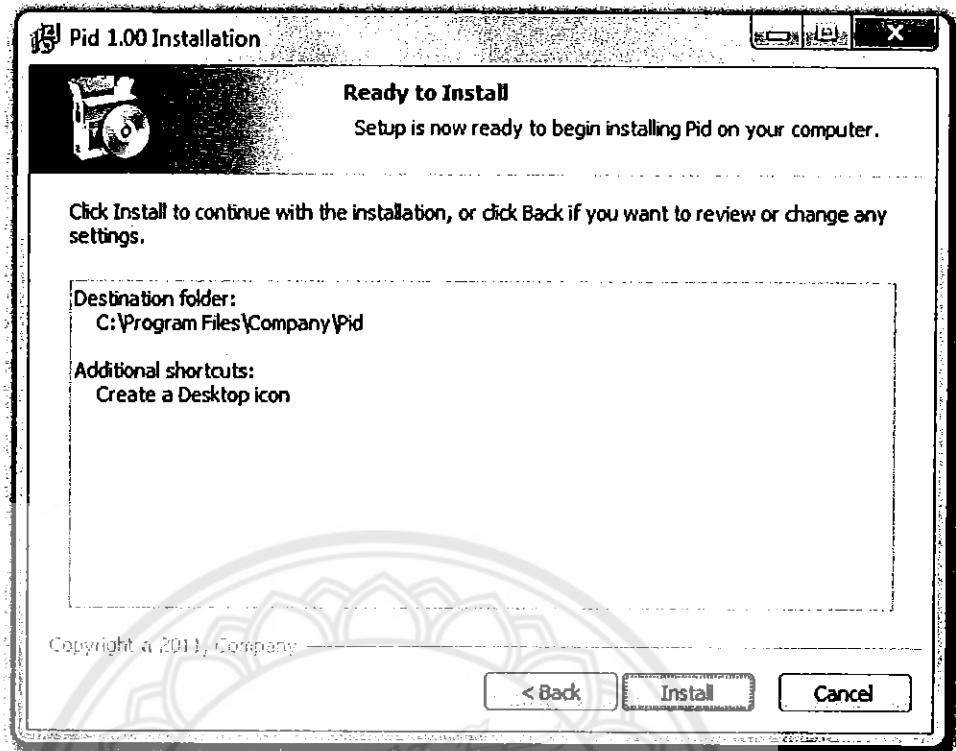


รูปที่ ก.3 ทำการเลือกโฟลเดอร์ที่ต้องการติดตั้งโปรแกรม

3. ทำการเลือกว่าจะทำการสร้างไอคอนไว้ที่หน้าจอหรือไม่แล้วเลือก Next ดังรูปที่ ก.4 และทำการเริ่มติดตั้ง โดยเลือก Install ดังรูปที่ ก.5

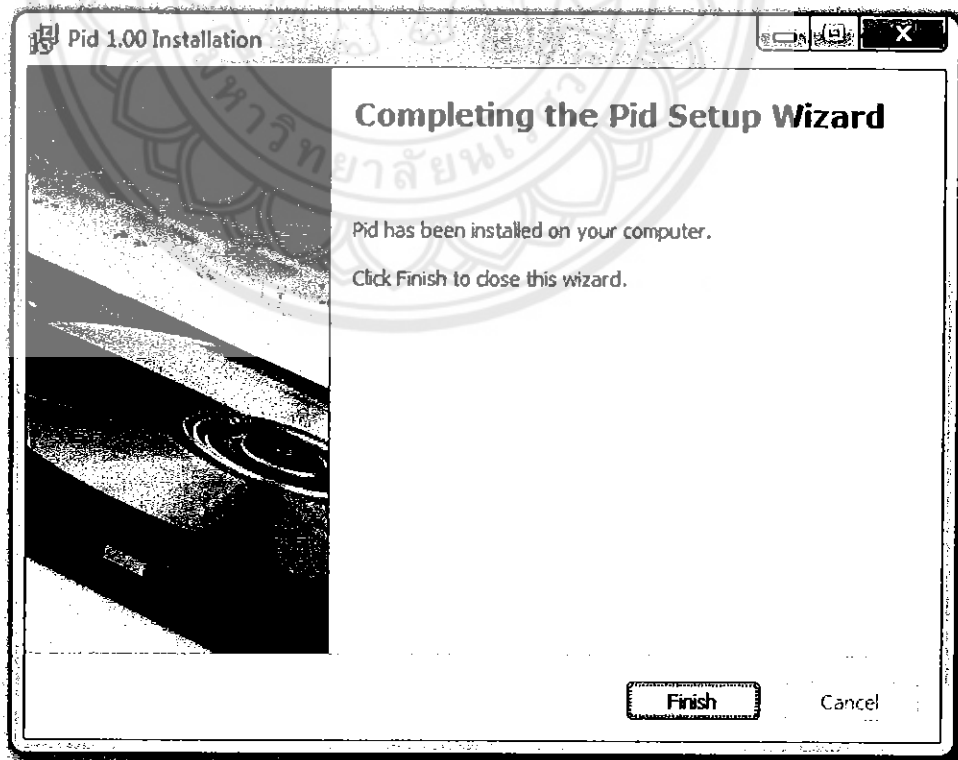


รูปที่ ก.4 ทำการสร้างไอคอนไว้ที่หน้าจอ



รูปที่ ก.5 เริ่มการติดตั้งโปรแกรม

4. การติดตั้งโปรแกรมเสร็จสิ้นทำการเลือก Finish ดังรูปที่ ก.6



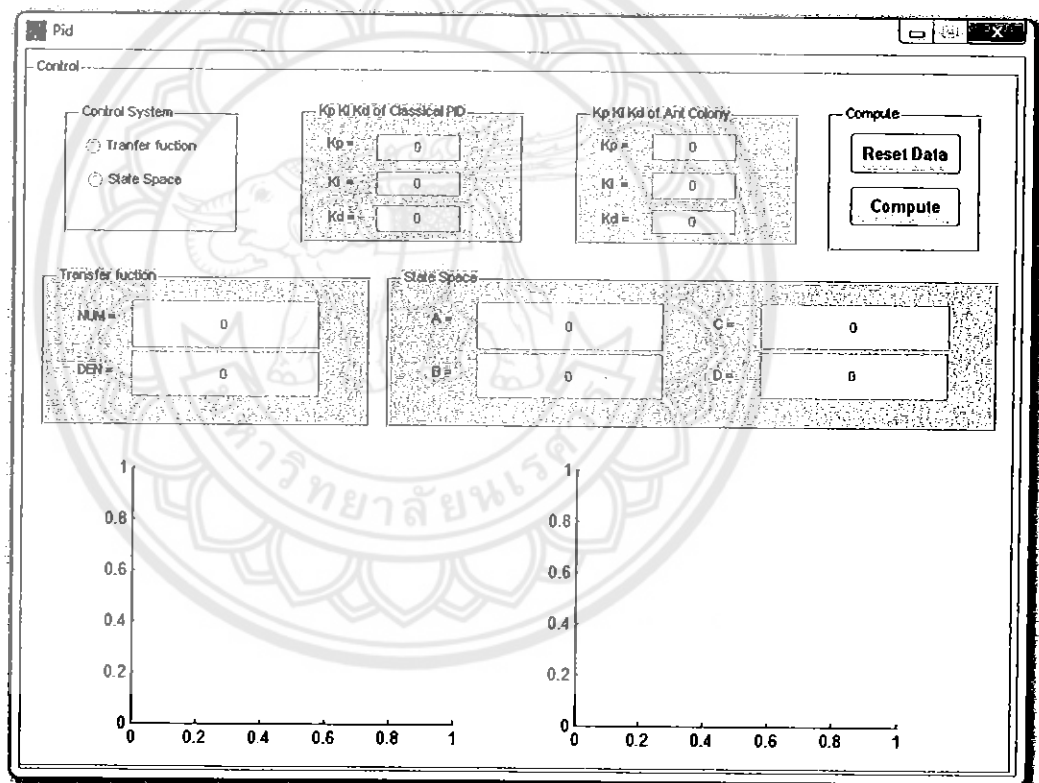
รูปที่ ก.6 การติดตั้งโปรแกรมเสร็จสิ้น

5. ไปที่โฟลเดอร์ที่ติดตั้งโปรแกรมดังรูปที่ ก.7

Computer > Local Disk (C:) > Program Files > Company > Pid			
Name	Date modified	Type	Size
Pid	20/3/2554 22:54	Application	4,059 KB
Uninstall	21/3/2554 6:19	Application	73 KB
Uninstall	21/3/2554 6:19	Configuration sett...	2 KB

รูปที่ ก.7 โฟลเดอร์ที่ทำการติดตั้ง โปรแกรม

6. ทำการดับเบิลคลิกที่ไอคอน PID จะขึ้นหน้าต่างของโปรแกรมดังรูปที่ ก.8

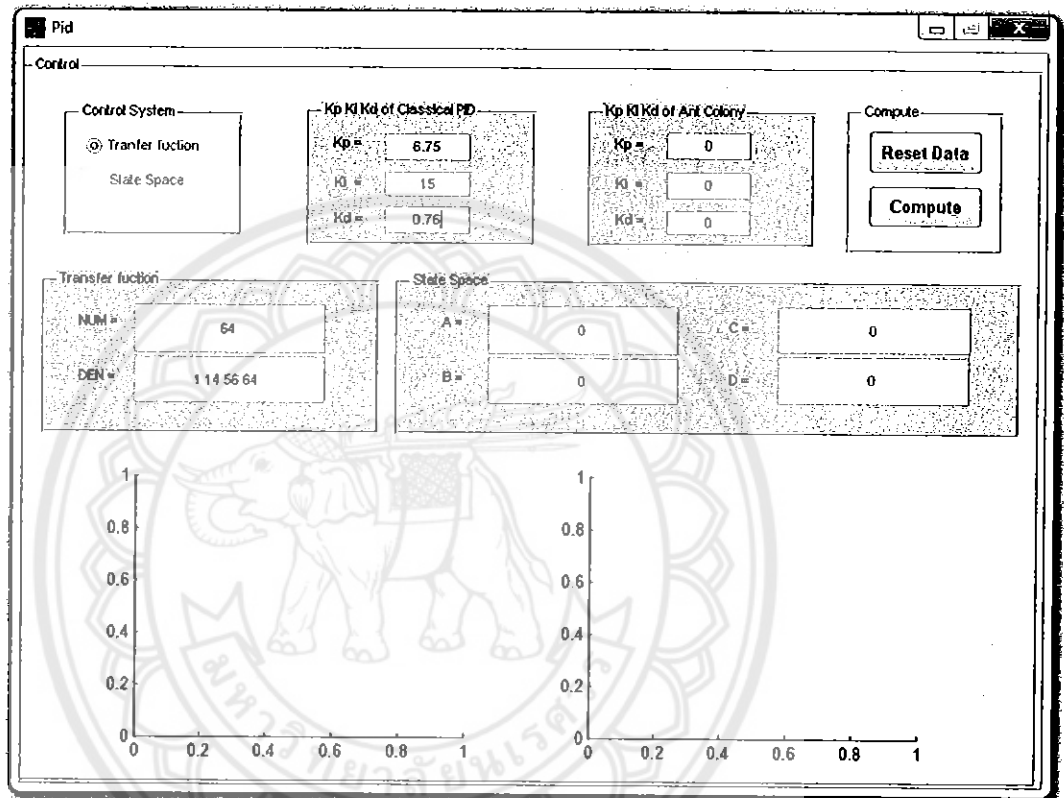


รูปที่ ก.8 หน้าต่างของโปรแกรม

7. จากรูปที่ ก.8 โปรแกรมมีหัวข้อ 6 หัวข้อและกราฟ 2 รูปซึ่งมีรายละเอียดดังต่อไปนี้

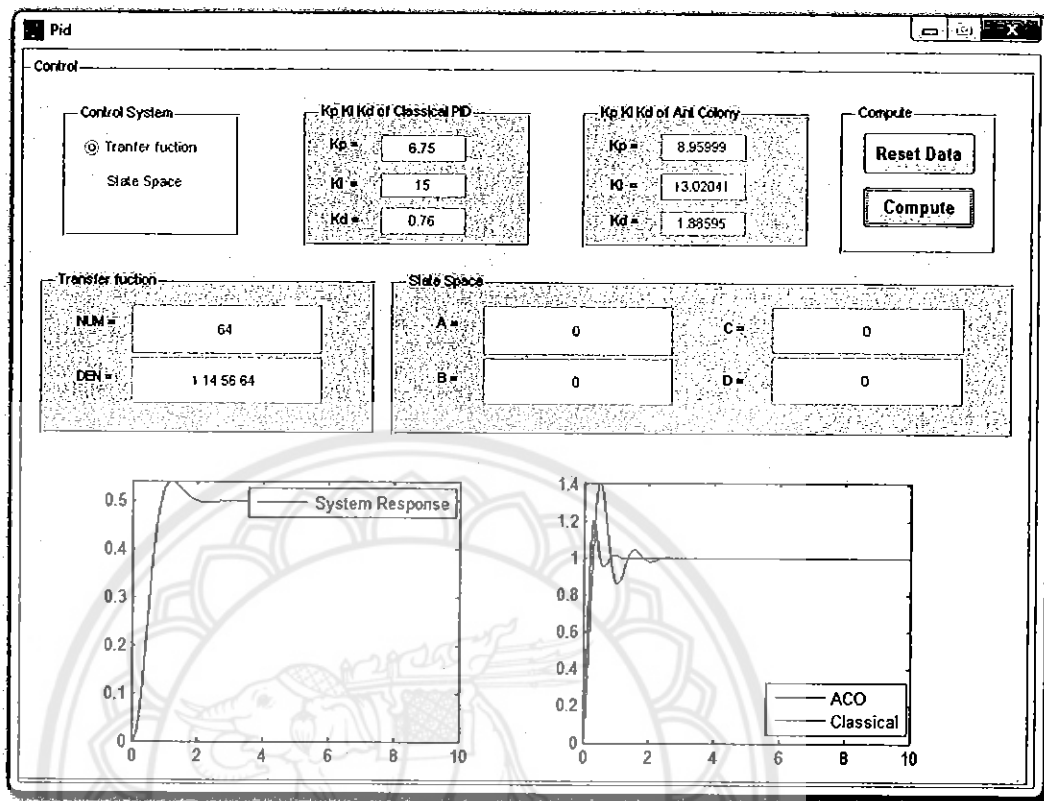
1. Control System ในหัวข้อนี้คือการเลือกระบบควบคุมที่ต้องการป้อนให้อยู่ในรูปแบบฟังก์ชันถ่ายโอน หรือปริภูมิสถานะ
2. K_p , K_i , K_d of Classical PID ในหัวข้อนี้คือการป้อนค่า K_p , K_i และ K_d ให้กับโปรแกรมซึ่งคำนวณได้จากการออกแบบระบบควบคุมแบบ PID

3. K_p, K_i, K_d of Ant Colony ในหัวข้อนี้คือค่า K_p, K_i และ K_d หลังจากใช้ ACS
4. Compute คือ หัวข้อที่จะเลือกจะทำ การล้างข้อมูลเก่า หรือจะทำการประมวลผล
5. Transfer function คือ หัวข้อในการป้อนสัญญาณเข้าเป็นแบบฟังก์ชันถ่ายโอน
6. State Space คือ หัวข้อในการป้อนสัญญาณเข้าเป็นแบบปริภูมิสถานะ
8. ทดลองป้อนข้อมูลลงใน โปรแกรมจากนั้นเลือก Compute ดังรูปที่ ก.9



รูปที่ ก.9 ทำการป้อนข้อมูล

9. หลังจากทำการเลือกปุ่ม "Compute" แล้วจะได้ผลดังรูปที่ ก.10



รูปที่ ก.10 ผลการคำนวณของ โปรแกรม

10. ถ้าผู้ใช้ต้องการป้อนข้อมูลของระบบใหม่จะต้องทำการล้างข้อมูลทั้งหมดก่อนซึ่งทำได้โดยการเลือกที่ปุ่ม Reset Data

ประวัติผู้เขียนโครงการ



ชื่อ นายณัฐพงศ์ พงศ์ไพจิตรวงศ์
ภูมิลำเนา จ.81/13 ต.ปากน้ำโพ อ.เมือง จ.นครสวรรค์ 60000
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

E-mail: toonzbra_net@hotmail.com



ชื่อ นายวีระพล บัวแก้ว
ภูมิลำเนา 7/2 หมู่ 5 ต.บึงเสนาท อ.เมือง จ.นครสวรรค์ 60000
ประวัติการศึกษา

- จบระดับมัธยมศึกษาจาก โรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยนเรศวร

E-mail: sexyangle_jv@hotmail.com