

การประยุกต์ใช้จีเนติกอัลกอริทึม เพื่อการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบ PID Application of Genetic Algorithm for Tuning of PID-Controller

นายณัฏฐวัฒน์ วีระยุทธ^{1*}, นายรัฐพงศ์ ปฏิภาณัง², นายกฤตยา ไชยยศ³, นายสว่างทิตย์ ศรีกิจสุวรรณ⁴

^{1, 2, 3} ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอุบลราชธานี ถ.สถลมารค อ.วารินชำราบ จ.อุบลราชธานี 34190

⁴ กลุ่มวิจัยเฉพาะทางวิศวกรรมอวกาศและทะเล มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ 10800

*ติดต่อ: weerayuth_gm@hotmail.com, 045-353363, Fax : 045-353333

บทคัดย่อ

ในปัจจุบันมีการพัฒนาทางด้าน Soft Computing โดยเฉพาะ Computational Intelligence เช่น Genetic Algorithms (GA) รวมทั้ง Ant Colony Optimization (ACO) และ Particle Swarm Optimization (PSO) ซึ่งนำไปสู่การประยุกต์ใช้ในการควบคุมขั้นสูงของกระบวนการในทางอุตสาหกรรมอย่างแพร่หลาย งานวิจัยนี้จึงนำเสนอการใช้ Genetic Algorithm ร่วมกับโปรแกรม MATLAB / Simulink ซึ่งง่ายต่อการนำไปใช้และมีเสถียรภาพสูงในการเข้าสู่ของการหาคำตอบสูง รวมทั้งใช้เวลาในการคำนวณค่อนข้างต่ำ เพื่อประยุกต์ใช้ในการออกแบบและปรับแต่งพารามิเตอร์ของตัวควบคุมแบบ PID เพื่อให้ได้ผลตอบสนองของการควบคุมที่เหมาะสม โดยค่าเหล่านี้จะถูกปรับแต่งโดยอัตโนมัติ โดยระบบควบคุมจะถูกออกแบบและจำลองการทำงานในโปรแกรม MATLAB / Simulink เพื่อทดสอบประสิทธิภาพของตัวควบคุมแบบ GA-based PID ที่ได้รับการปรับแต่งแล้ว และนำมาเปรียบเทียบกับตัวควบคุมแบบ PID ที่ปรับแต่งค่าพารามิเตอร์ด้วยวิธีการดั้งเดิม เช่น Ziegler-Nichols Method และ Chien-Hrones-Reswick Tuning Method (C-H-R Method) เพื่อดูผลตอบสนองของระบบควบคุม ทั้งนี้เพื่อเป็นการยืนยันและแสดงให้เห็นว่า การปรับแต่งตัวควบคุมแบบ PID ด้วย Genetic Algorithm ร่วมกับ MATLAB และ Simulink สามารถทำได้โดยสะดวก มีประสิทธิภาพและรวดเร็วกว่าวิธีการแบบดั้งเดิม

คำหลัก: Genetic Algorithm, Ziegler-Nichole Method, Chien-Hrones-Reswick Tuning Method, PID Tuning

Abstract

In the recent Year Soft Computing, especially Computational Intelligence, such as Genetic Algorithms (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), has been extensively studied and developed, which led to the Application in advance Control of industrial Processes. This Research proposes the Use of Genetic Algorithm in Conjunction with MATLAB / Simulink, which is easy to use and highly reliable in the Convergence of Finding an Answer with relatively low computational Time, for Designing and Tuning Parameters of PID Controller to obtain the optimal Control. These Control Parameters will be adjusted automatically. The Control System is designed and simulated in the MATLAB / Simulink Environment to test the Performance of GA-based PID Controller. The Comparison between GA-based PID Controller and the traditional Parameter Tuning Methods, such as

DRC-9

Ziegler-Nichols Method and Chien-Hrones-Reswick, is performed to confirm and demonstrate the Advantage of Using GA-based Parameter Tuning Method over the conventional One.

Keywords: Genetic Algorithm, Ziegler-Nichols Method, Chien-Hrones-Reswick Tuning Method, PID Tuning

1. บทนำ

ตั้งแต่ปี 1930 เป็นต้นมา ตัวควบคุมแบบ PID ซึ่งประกอบด้วยองค์ประกอบหลักทั้ง 3 ส่วน นั่นคือ ส่วนควบคุมแบบสัดส่วน (Proportional Action) ส่วนควบคุมแบบปริพันธ์ (Integral Action) และส่วนควบคุมแบบอนุพันธ์ (Derivative Action) ถูกผลิตออกมาเชิงพาณิชย์และมีการนำมาใช้อย่างแพร่หลายในเชิงอุตสาหกรรมหลายประเภท ระเบียบวิธีต่างๆ ในการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบ PID ได้ถูกเสนอขึ้นและนำมาใช้มาตั้งแต่ปี 1942 เพื่อให้ได้ผลตอบสนองของการควบคุมเป็นตามวัตถุประสงค์เช่น ให้มีค่า เปอร์เซนต์ของโอเวอร์ชูท (Percent of Overshoot) ผลรวมค่าสัมบูรณ์ของความผิดพลาด (IAE) ผลรวมของความผิดพลาดยกกำลังสอง (ISE) ตลอดจนเวลาในการลู่เข้าสู่ค่าที่ต้องการ (Settling Time) เป็นไปตามที่ต้องการ [1] ในบรรดาระเบียบวิธีเหล่านั้นคือ Ziegler-Nichols Tuning Method [6] ซึ่งเป็นวิธีการที่ได้รับความนิยมอย่างแพร่หลาย โดยนำเอาคุณลักษณะของผลตอบสนองในเวลาชั่วขณะ (Transient Response Characteristics) มาใช้ในการปรับแต่งค่าพารามิเตอร์ ทั้งนี้การปรับแต่งสามารถทำได้ทั้งแบบวงเปิด (Open Loop Tuning) และแบบวงปิด (Closed Loop Tuning) นอกจากนี้ยังมีรูปแบบการปรับแต่งพารามิเตอร์ของตัวควบคุมแบบ PID คือระเบียบวิธีของ Chien-Hrones-Reswick หรือ C-H-R-Method ซึ่งถูกพัฒนาต่อยอดจาก Ziegler Nichol Method แบบวงเปิด [1] เพื่อให้สามารถกำหนดค่า Overshoot ที่ต้องการได้ 2 ค่าที่แตกต่างกันคือ 20% Overshoot และ Least Overshoot ทั้งนี้ระเบียบวิธีที่ได้กล่าวมาแล้วในข้างต้น ต้องอาศัยการปรับแต่งค่าพารามิเตอร์แบบ Trial & Error ในขั้นตอนการปรับตั้งละเอียด (Fine Tuning) เพื่อให้ระบบควบคุมมีผลตอบสนองใกล้เคียงกับที่ต้องการให้มาก

ที่สุด ซึ่งอาจจะใช้เวลามากน้อยแตกต่างกันขึ้นอยู่กับประสิทธิภาพของวิศวกรผู้ออกแบบหรือผู้ที่ทำหน้าที่ปรับแต่งค่าพารามิเตอร์ของตัวควบคุม

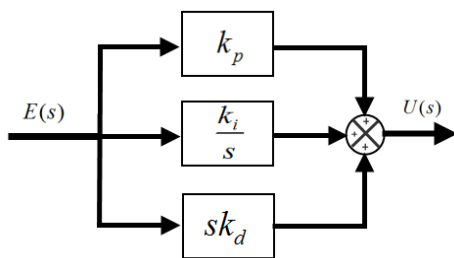
นอกจากนี้ยังมีเทคนิคการทำ PID-Tuning อีกหลายรูปแบบที่แตกต่างกัน โดยนำเอาขั้นตอนวิธีเชิงวิวัฒนาการ (Evolutionary Algorithms) ต่างๆ เข้ามาประยุกต์ใช้ เช่น ACO (Ant Colony Optimization) PSO (Particle Swarm Optimization) และ DE (Differential Evolution) ตลอดจน GA หรือ Genetic Algorithm [3] และมีการทบทวนวรรณกรรมการใช้ GA และ Soft Computing Techniques ใน PID [8] โดยจะสุ่มหาคำตอบซึ่งเป็นค่าพารามิเตอร์ (k_p, k_i, k_d) ภายใต้เงื่อนไขของฟังก์ชันเป้าหมาย (Objective Function) ที่ถูกกำหนดโดยผู้ออกแบบเพื่อให้ได้ผลตอบสนองของระบบควบคุมที่ต้องการการนำไปใช้งานจึงจำเป็นต้อง แปลงขั้นตอนวิธีเหล่านั้นให้อยู่ในรูปโปรแกรมต่างๆ ที่สามารถใช้คอมพิวเตอร์ช่วยในการคำนวณได้ ซึ่งเป็นการยากต่อการทำความเข้าใจ และยากสำหรับการต่อยอดงานวิจัยในด้านนี้

งานวิจัยนี้จึงนำเสนอการใช้ GA (Genetic Algorithm) ซึ่งทำงานร่วมกับโปรแกรม MATLAB / Simulink ที่ง่ายต่อการนำไปใช้และมีเสถียรภาพสูงในการลู่เข้าของการหาคำตอบสูง รวมทั้งใช้เวลาในการคำนวณค่อนข้างต่ำเมื่อเทียบกับวิธี Trial & Error เพื่อประยุกต์ใช้ในการออกแบบและปรับแต่งพารามิเตอร์ของตัวควบคุมแบบ PID เพื่อให้ได้ผลตอบสนองของการควบคุมที่เหมาะสม โดยเสนอเป็นลำดับขั้นตอนที่ง่ายต่อการทำความเข้าใจ รวมทั้งแสดงการเปรียบเทียบเพื่อให้เห็นข้อดีที่เหนือกว่าระเบียบวิธีการปรับแต่งค่าพารามิเตอร์แบบดั้งเดิมเพื่อเป็นกรณีศึกษา

2. ตัวควบคุมแบบพีไอดี (PID Controller)

DRC-9

ตัวควบคุมแบบ PID ประกอบด้วยองค์ประกอบ 3 ส่วนหลักคือ 1. ส่วนควบคุมแบบสัดส่วน (Proportional Action) 2. ส่วนควบคุมแบบปริพันธ์ (Integral Action) และ 3. ส่วนควบคุมแบบอนุพันธ์ (Derivative Action) โดยจะมีพารามิเตอร์ของตัวควบคุมที่สอดคล้องกับองค์ประกอบหลักทั้ง 3 คือ k_p , k_i และ k_d ตามลำดับ ความสำคัญของค่าเหล่านี้คือ เมื่อมีการปรับแต่งหรือใช้ค่าที่เหมาะสมจะทำให้ ค่า Overshoot ลดน้อยลง และทำให้ค่าความผิดพลาดที่สภาวะคงตัว (Steady State Error) เป็นศูนย์ รวมทั้งทำให้ระบบควบคุมมีเสถียรภาพมากขึ้น โครงสร้างของตัวควบคุมแบบ PID ในรูปแบบของแผนภาพบล็อก (Block Diagram) แสดงไว้ในรูปที่ 1

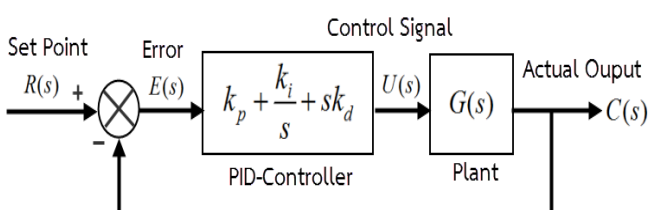


รูปที่ 1 โครงสร้างของตัวควบคุมแบบ PID

โดยตัวควบคุมแบบ PID มีฟังก์ชันถ่ายโอน (Transfer Function) ดังสมการที่ (1)

$$G_c(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \quad (1)$$

เมื่อ $U(s)$ คือสัญญาณควบคุม (Control Signal) และมี $E(s)$ เป็นสัญญาณของความผิดพลาด (Error Signal) หรือ ผลต่างระหว่างค่าอ้างอิงหรือ Set Point และค่าที่วัดได้จริงในระบบ (Actual Output) ดังจะเห็นได้จากรูปที่ 2



รูปที่ 2 Block Diagram ของระบบควบคุมแบบ PID

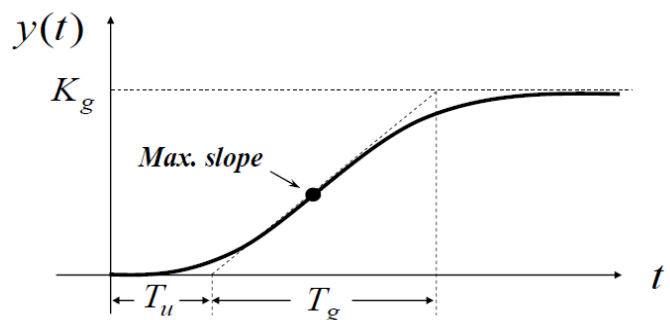
2.1 การปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบ PID (Conventional PID-Tuning Methods)

ค่าพารามิเตอร์ของตัวควบคุมแบบ PID สามารถหาได้จากการใช้ระเบียบวิธีที่แตกต่างกัน โดยทั่วไปแล้วระเบียบวิธีดังกล่าวจะถูกแบ่งออกเป็น 2 กลุ่มหลักคือ 1. ระเบียบวิธีแบบวงเปิด (Open Loop Tuning Method) และ 2. ระเบียบวิธีแบบวงปิด (Closed Loop Tuning Method) ซึ่งเป็นการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมในระหว่างที่ระบบทำงานแบบวงเปิด (Open Loop Operation) และระบบทำงานแบบวงปิด (Closed Loop Operation) ตามลำดับ

2.1.1 ระเบียบวิธีของ Ziegler-Nichols

แบ่งออกเป็นสองวิธีย่อย ทั้งนี้ขึ้นอยู่กับอันดับ (Order) ของระบบที่ต้องการควบคุม

1. ระเบียบวิธีแบบวงเปิด (Open Loop Method Method) - ใช้ในกรณีเมื่อผลตอบสนองของระบบที่ต้องการควบคุมไม่มีการแกว่ง (No Oscillation) กล่าวคือ $\zeta \geq 1$ หรือมีพฤติกรรมเป็นระบบที่มีอันดับหนึ่ง โดยจะใช้ Ziegler-Nichols Reaction Curve ดังแสดงในรูปที่ 3 ร่วมกับตารางที่ 1



รูปที่ 3 แสดง Ziegler-Nichols Reaction Curve

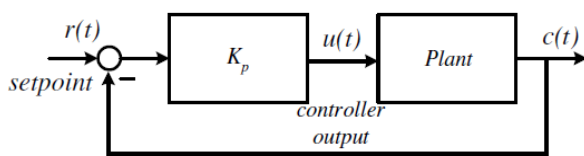
ตารางที่ 1 ค่าพารามิเตอร์สำหรับ Open Loop Ziegler-Nichols Method

Controller	k_p	T_i	T_d
------------	-------	-------	-------

DRC-9

P	$T_g/K_g T_u$	-	-
PI	$0.9 T_g/K_g T_u$	$T_u/0.3$	-
PID	$1.2 T_g/K_g T_u$	$2T_u$	$0.5T_u$

2. ระเบียบวิธีแบบวงปิด (Closed Loop Method Method) – ใช้ในกรณีที่ผลตอบสนองของระบบที่ต้องการควบคุมเกิดการแกว่ง ($0 < \zeta < 1$) การปรับแต่งค่าพารามิเตอร์ของ PID-Controller สามารถทำได้โดยทำการพิจารณาเบื้องต้นจากระบบควบคุม Proportional แบบวงปิด ดังแสดงในรูปที่ 4



รูปที่ 4 แสดงระบบควบคุม Proportional แบบวงปิด (Closed Loop Proportional Control System)

โดยการหาปรับค่า k_p แบบ Trial & Error ที่ทำให้ผลตอบสนองของระบบเริ่มแกว่ง หรือสามารถคำนวณได้โดยตรงจาก Routh-Hurwitz Criterion เมื่อทราบฟังก์ชันถ่ายโอนของระบบ จะได้ค่าเกณฑ์วิกฤติ k_{pu} และคาบเวลาวิกฤติ P_u เพื่อใช้หาพารามิเตอร์ตามตารางที่ 2 ในขั้นตอนต่อไป ทั้งนี้พารามิเตอร์ในตารางที่ 1 และตารางที่ 2 จะแตกต่างกันจากการที่ (1) โดยสามารถเขียนเป็นความสัมพันธ์ใหม่ดังสมการที่ (2)

$$G_c(s) = \frac{U(s)}{E(s)} = k_p \left(1 + \frac{T_i}{s} + T_d s\right) \quad (2)$$

ตารางที่ 2 ค่าพารามิเตอร์สำหรับ Closed Loop Ziegler-Nichols Method

Controller	k_p	T_i	T_d
P	$0.45k_{pu}$	-	-
PI	$0.45k_{pu}$	$P_u / 1.2$	-
PID	$0.45k_{pu}$	$P_u / 2$	$P_u / 8$

2.1.2 ระเบียบวิธีของ Chien-Hrones-Reswick

เป็นวิธีการที่พัฒนามาจาก Open Loop Ziegler-Nichols Method โดยสามารถเลือกใช้ชนิดของตัวควบคุมได้จากพารามิเตอร์ R ซึ่งได้มาจากการทดลองวัดผลตอบสนองวงเปิดของระบบ ดังรูปที่ 3 ร่วมกับตารางที่ 3 นอกจากนี้ยังสามารถเลือกขนาดของ Overshoot ที่ต้องการได้ 2 ค่า คือที่ 20% หรือให้มี Overshoot น้อยที่สุดเท่าที่จะเป็นไปได้ จากสูตรคำนวณในตารางที่ 4 และตารางที่ 5 ตามลำดับ

ตารางที่ 3 การเลือกชนิดของตัวควบคุมจากค่า R

Controller	$R = T_g / T_u$
P	$R > 10$
PI	$7.5 < R < 10$
PID	$3 < R < 7.5$
Higher Order	$R < 3$

ตารางที่ 4 ค่าพารามิเตอร์สำหรับ 20% Overshoot Chien-Hrones-Reswick Method

Controller	k_p	k_i	k_d
P	$0.7R / K_g$	-	-
PI	$0.6R / K_g$	K_p / T_g	-
PID	$0.95R / K_g$	$K_p / 1.35T_g$	$0.47K_p T_u$

ตารางที่ 5 ค่าพารามิเตอร์สำหรับ Least Overshoot Chien-Hrones-Reswick Method

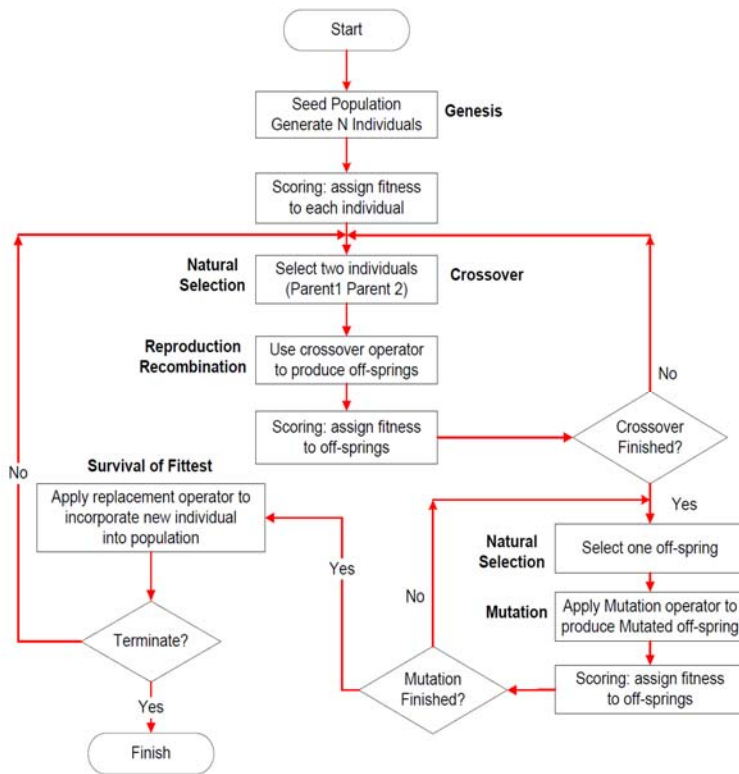
Controller	k_p	k_i	k_d
P	$0.3R / K_g$	-	-
PI	$0.35R / K_g$	$K_p / 1.2T_g$	-
PID	$0.6R / K_g$	K_p / T_g	$0.50K_p T_u$

3. ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithms)

จีเนติกอัลกอริทึมหรือขั้นตอนวิธีเชิงพันธุกรรมเป็นกระบวนการค้นหาคำตอบที่เหมาะสมเชิงฮิวริสติก (Heuristic Optimization Technique) ที่ถูกพัฒนาขึ้น

DRC-9

โดย John H. Holland และมีต้นแบบมาจากกฎการคัดเลือกทางธรรมชาติ [5] ขั้นตอนการดำเนินการค้นหาคำตอบของ GA สามารถสรุปไว้เป็นแผนผังการทำงานดังรูปที่ 5



รูปที่ 5 แสดงแผนผังการทำงานงานของ GA

โดยมีตัวดำเนินการ (Operators) หลัก 3 อย่างคือ 1.Selection 2.Crossover 3.Mutation ซึ่งในแต่ละรอบของการทำงานสามารถสรุปได้ดังนี้ [2]

• [Start] - สร้างกลุ่มประชากรเริ่มต้นจำนวน N ค่าในรูปแบบ Chromosomes โดยการสุ่ม (ภายในกรอบหรือช่วงของคำตอบที่เป็นไปได้)

• [Fitness] - คำนวณค่า Fitness ของประชากรแต่ละตัว หรือค่า Fitness (F(x)) ของ Chromosome ทุกตัว

• [New Population] - สร้างกลุ่มประชากรรุ่นใหม่ โดยทำตามขั้นตอนดังต่อไปนี้

- [Selection] :เลือกประชากรต้นแบบที่ละคู่ โดยสุ่มจากความน่าจะเป็นที่คำนวณจากค่า Fitness ของ

ประชากรแต่ละตัว (ค่า Fitness มีโอกาสได้รับการคัดเลือกสูง)

- [Crossover] :สุ่มเลือก Cross Site หรือตำแหน่งของการ Cross Over ระหว่างคู่ประชากรที่ถูกสุ่มเลือกมาแล้ว เพื่อสร้าง Off Spring หรือประชากรรุ่นใหม่

- [Mutation] :โดยการใช้ความน่าจะเป็นของการเกิดการผ่าเหล่า หรือ Mutation เพื่อเปลี่ยนค่าของตัวเลขบนตำแหน่งที่ถูกสุ่มมาจาก Chromosome

- [Accepting] :นำ Offspring ที่ได้ไปแทนที่ในกลุ่มประชากรเก่าเพื่อสร้างกลุ่มประชากรใหม่

- [Replace] :ประชากรกลุ่มใหม่ที่ได้ จะถูกนำไปใช้ในการค้นหาคำตอบในรอบต่อไป โดยมีเงื่อนไขให้การหยุดค้นหาคำตอบ เมื่อค้นพบคำตอบที่เหมาะสมแล้ว หรือเมื่อครบจำนวนรอบสูงสุดในการคำนวณ

4. การปรับค่าพารามิเตอร์ของตัวควบคุมแบบ PID ด้วยจีเนติกอัลกอริทึม

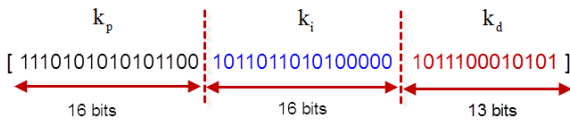
4.1 การกำหนดตัวแปรเริ่มต้นของ GA

ในเบื้องต้นจะทำการกำหนดจำนวนของประชากร N ตัว ที่จะใช้ในการค้นหาคำตอบ โดยสร้างจากช่วงของคำตอบที่เป็นไปได้ในรูปแบบของ Chromosome ในที่นี้เป็นการค้นหาค่า k_p , k_i , k_d ซึ่งแต่ละค่าจะถูกเข้ารหัส (Encoding) ให้แสดงผลในรูปแบบของ Binary String (เลขฐาน 2) โดยความยาวของ Binary String ของแต่ละค่าคำนวณได้จากสมการ (3)

$$2^{N_{bj}-1} < (U_{bj} - L_{bj}) \times 10^{SN_j} \leq 2^{N_{bj}} - 1 \quad (3)$$

เมื่อ U_{bj} , L_{bj} เป็นขอบเขตบน (Upper Bound) และขอบเขตล่าง (Lower Bound) ของตัวแปรที่ต้องการค้นหาตามลำดับ และ SN_j คือเลขนัยสำคัญแสดงตำแหน่งทศนิยมของตัวแปรที่ต้องการ N_{bj} คือจำนวนบิตที่แสดงความยาวของ Binary String [3]

DRC-9



รูปที่ 6 Chromosome แสดงค่า PID พารามิเตอร์ ตัวอย่างเช่น $k_p \in [0, 5]$, $k_i \in [0, 6]$, $k_d \in [0, 0.5]$ และต้องการเลขนัยสำคัญ $SN_j = 4$ ตำแหน่งจะได้ จำนวนบิตแสดง Binary String ของแต่ละตัวแปรคือ 16 บิต, 16 บิตและ 13 บิต ตามลำดับ นั่นคือ Chromosome จะมีความยาวรวมคือ 45 บิต ดังแสดงในรูปที่ 6

ทั้งนี้ตัวแปรตัดสินใจ (Decision Variable) $x_j (j=1-3)$ ในรูปที่ 6 จะถูกถอดรหัส (Decoding) ให้เป็นจำนวนจริงที่มีทศนิยม 4 ตำแหน่งตามสมการที่ (4)

$$x_j = L_{bj} + \text{Decimal}(\text{Substring}_j) \times \frac{(U_{bj} - L_{bj})}{2^{N_{bj}} - 1} \quad (4)$$

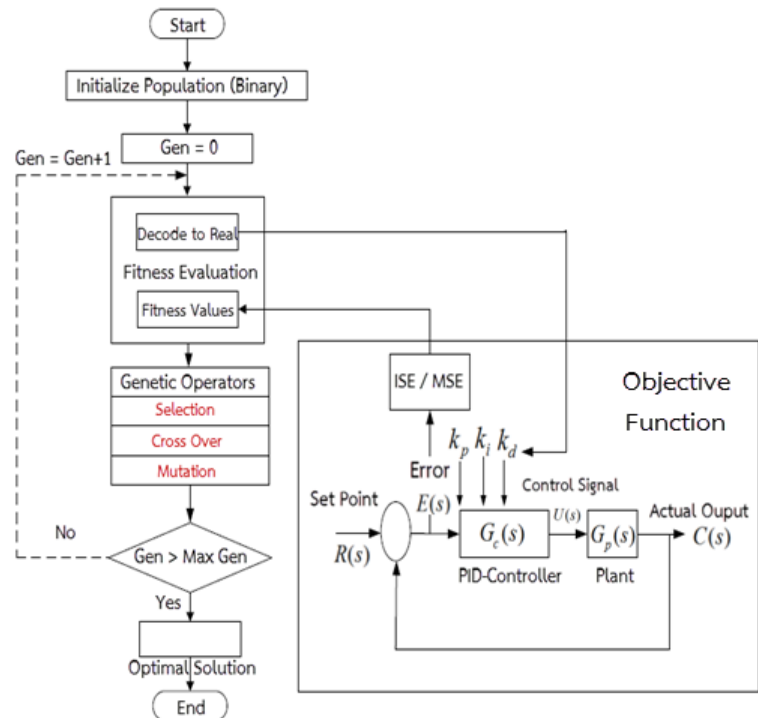
4.2 ฟังก์ชันเป้าหมาย (Objective Function)

โดยทั่วไปแล้วจะกำหนดสมรรถนะในการควบคุมอัตโนมัติผ่านผลตอบสนองต่อ Step Input ซึ่งอยู่ในรูปผลรวมค่าถ่วงน้ำหนัก (Weighted Combination) ของผลรวมของค่าความผิดพลาดยกกำลังสอง (ISE) ค่า Overshoot (OS) และ 5% Settling Time (t_s) [2] ดังสมการที่ 5

$$\text{Ob_Func} = \alpha(\text{ISE}) + \beta(\text{OS}) + \gamma(t_s) \quad (5)$$

เมื่อ α, β, γ เป็น Improvement Factors ใช้กำหนดผลตอบสนองของระบบควบคุมที่ต้องการ โดยที่ Chromosome ทุกตัวจะถูกถอดรหัสเป็นจำนวนจริงเพื่อนำไปใช้ในการคำนวณค่า Objective Function ตามสมการที่ (2) เพื่อนำค่า Fitness ที่ได้ไปเรียงลำดับตามความน่าจะเป็นของโอกาสที่ประชากรแต่ละตัวจะถูกคัดเลือก เพื่อนำไปสร้าง New

Offspring ในการค้นหาคำตอบในรอบการคำนวณถัดไป โดยมี Flow Chart แสดงการทำงานในการค้นหาค่าพารามิเตอร์ของ PID Controller ดังแสดงในรูปที่ 7



รูปที่ 7 Flow Chart แสดงการค้นหาค่าพารามิเตอร์ของ PID Controller

5.การจำลองการทำงาน (Simulation)

โดยการทดสอบกับระบบที่รู้แบบจำลองทางคณิตศาสตร์หรือฟังก์ชันถ่ายโอนของระบบ เช่น

$$G(s) = \frac{64}{s^3 + 14s^2 + 56s + 64} \quad (6)$$

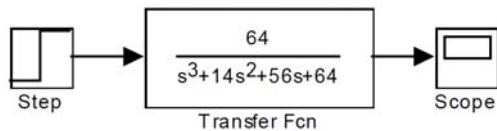
นำระบบในสมการที่ 6 ไปหาผลตอบสนองของระบบวงเปิดต่อ Unit Step Input โดย Simulink ดังรูปที่ 8 ซึ่งจะได้ Ziegler-Nichols Reaction Curve ในรูปแบบที่แสดงในรูปที่ 3 ซึ่งข้อมูลที่ได้จะถูกนำไปใช้ในการหาค่า Gain ของตัวควบคุม PID ต่อไป

DRC-9

5.1 PID-Tuning โดยใช้ระเบียบวิธีของ

Ziegler-Nichols & Chien-Hrones-Reswick

(CHR) : เบื้องต้นจะใช้ระเบียบวิธี ตามตารางที่ 1 ตารางที่ 4 และ 5 ตามลำดับ ซึ่งจะได้ค่าพารามิเตอร์ของตัวควบคุมแบบ PID ดังแสดงในตารางที่ 6

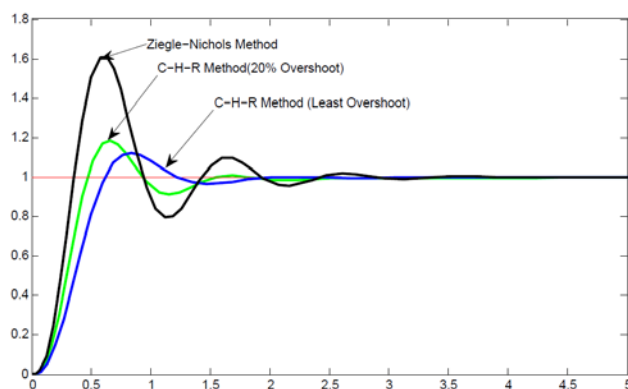
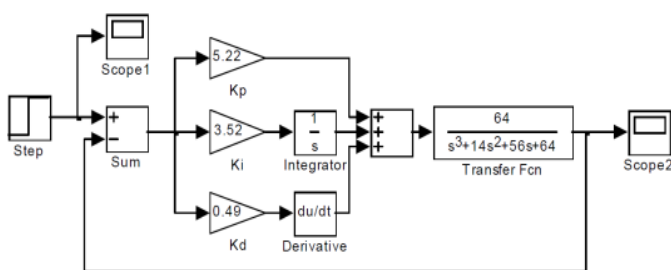


รูปที่ 8 การหาผลตอบสนองแบบ Open Loop

ตารางที่ 6 ค่าพารามิเตอร์ของ PID Controller

PID Gain	k_p	k_i	k_d
Ziegler-Nichols	6.600	16.500	0.660
CHR-20%	5.220	3.520	0.490
CHR-LeastOV	3.300	3.000	0.333

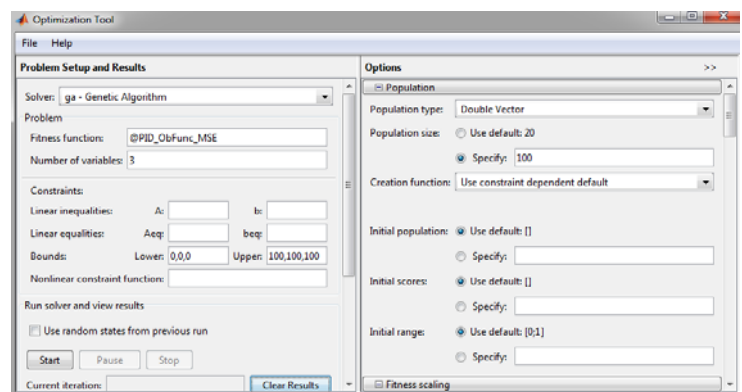
โดยใช้โปรแกรม MATLAB/Simulink จะได้ผลตอบสนองที่ต่างกันเมื่อใช้ PID Gain จากระเบียบวิธีที่ต่างกัน ดังรูปที่ 9



รูปที่ 9 ผลตอบสนองการควบคุมแบบ ZN /CHR-PID

5.2 PID-Tuning โดยใช้ Genetic Algorithm

ค่าพารามิเตอร์ในตารางที่ 7 ได้จากการใช้ GA Toolbox ในโปรแกรม MATLAB [7] ดังรูปที่ 10 ค้นหาค่าตอบ โดยมีฟังก์ชันเป้าหมายในสมการ (5) เป็นเกณฑ์ในการกำหนดทิศทางการหาค่าตอบที่เหมาะสม โดยกำหนดให้มีค่าขอบเขตล่างและขอบเขตบนที่ $(k_p, k_i, k_d) \in [0, 100]$

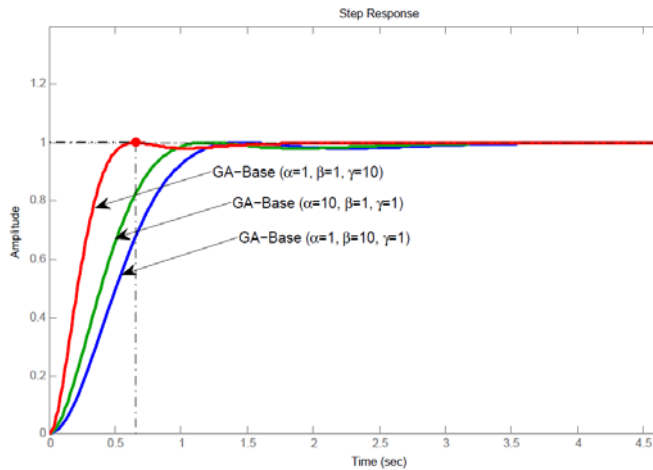


รูปที่ 10 แสดง GA Toolbox ในโปรแกรม MATLAB

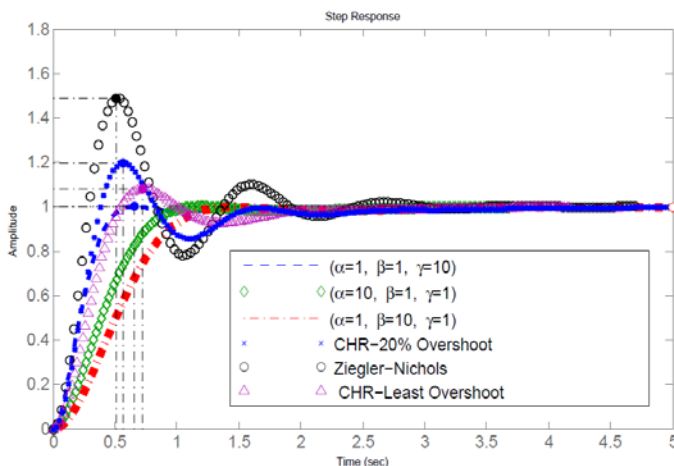
ตารางที่ 7 ค่าพารามิเตอร์ของ GA-based PID Controller

GA-based PID Gain	k_p	k_i	k_d
$\alpha = 1, \beta = 10, \gamma = 1$	1.347	1.761	0.099
$\alpha = 1, \beta = 1, \gamma = 10$	3.410	4.034	0.677
$\alpha = 10, \beta = 1, \gamma = 1$	1.827	2.228	0.216

DRC-9



รูปที่ 11 ผลตอบสนองการควบคุม GA-based-PID Controller



รูปที่ 12 ผลตอบสนองการควบคุม GA-based- vs. conventional Tuning Method –Controller ตารางที่ 8 พารามิเตอร์ของผลตอบสนองการควบคุม

PID-Controller	E_{ss}	%OS	t_s (s)
$\alpha = 1, \beta = 10, \gamma = 1$	0.00	0.00	1.189
$\alpha = 1, \beta = 1, \gamma = 10$	0.00	0.010	0.528
$\alpha = 10, \beta = 1, \gamma = 1$	0.000	0.000	0.965
Ziegler-Nichols	0.000	48.87	2.339
CHR-20% OV	0.000	19.835	2.557
CHR-Least OV	0.000	8.098	2.005

6 สรุปและข้อเสนอแนะ

แม้ว่าการปรับแต่งค่าพารามิเตอร์ของตัวควบคุมแบบ PID ด้วยวิธีการแบบดั้งเดิมเช่น Ziegler-Nichols และ Chien-Hrones-Reswick Tuning Method

สามารถทำได้อย่างสะดวกดังรูปที่ 12 แต่ยังต้องการขั้นตอนการทดสอบด้วยอินพุตแบบวงเปิด หรือแบบวงปิด เพื่อหาค่าตัวแปรที่จำเป็นต้องนำไปใช้กับตารางที่ 1-5 นอกจากนี้ยังต้องการขั้นตอน Fine Tuning เพื่อให้ได้ผลตอบสนองตามต้องการ ซึ่งต้องอาศัยทั้งประสบการณ์และวิธี Trial & Error จึงทำให้ใช้เวลานาน ในขณะที่เมื่อใช้ GA สามารถทำได้สะดวกและรวดเร็วและสามารถควบคุมพารามิเตอร์ต่างๆของผลตอบสนองดังแสดงในตารางที่ 8 ผ่าน Objective Function ที่ง่ายต่อการปรับแก้ค่าถ่วงน้ำหนัก (α, β, γ) เพื่อแก้ไขค่า ISE t_s หรือ Percent Overshoot ดังรูปที่ 11-12 ซึ่งให้ผลตอบสนองที่ดีกว่าวิธีการปรับแต่งค่าพารามิเตอร์แบบดั้งเดิม รวมทั้งง่ายต่อการพัฒนาเทคนิคนี้ เพื่อนำไปใช้งานกับระบบที่ไม่รู้แบบจำลองทางคณิตศาสตร์ที่แน่นอน ในทางปฏิบัติ

6. เอกสารอ้างอิง

- [1] Comparison of PID Controller Tuning Methods, Sharif University of Technology, Annual Report.
- [2] Fundamental of Genetic Algorithm : AI Course Lecture 39-40, Notes, Slide, URL: http://www.myreaders.info/html/artificial_intelligence.html, access on 12/06/2014.
- [3] Mohd S. Saad, Hishamuddin Jamaluddin, Intan Z. M. Darus (2012). PID Controller Tuning Using Evolutionary Algorithms, WSEAS Transaction on Systems and Control, Issue 4, vol. 7, October 2012, pp.139-149.
- [4] Arturo Y. Jaen-Cuellar¹, Rene de J. Romero-Troncoso, Luis Morales-Velazquez and Roque A. Osornio-Rios (2013). PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems, International Journal of Advanced Robotic Systems, Vol. 10, 324:2013, pp.1-14.
- [5] D.E.Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning. 1988 :Addison-Wesley Publishing Co., Inc.

DRC-9

[6] J.G.Ziegler and N.B.Nichols, (1942). Optimum Setting for Automatic Controllers, Transaction of American Society of Mechanical Engineers, vol.64,1942,pp.759-768.

[7] Subhadip Samanta, (2014). Genetic Algorithm: An Approach for Optimization (Using MATLAB), International Journal of Latest Trends in Engineering and Technology (IJLTET), vol. 3 Issue 3 January 2014,pp.261-267.

[8] Manoj Kushwah et al. (2014). Tuning PID Controller for Speed Control of DC Motor Using Soft Computing Techniques-A Review, Advance in Electronic and Electric Engineering. Volume 4, Number 2 (2014), pp. 141-148.