

Design of PID Controllers by Bat Algorithm for Benchmark Systems

การออกแบบตัวควบคุมพีไอดีด้วยขั้นตอนวิธีแบบค้างคาวสำหรับระบบเกณฑ์มาตรฐาน

Received	20 Jun 20
Reviewed	22 Jun 20
Revised	22 Sep 20
Accepted	6 Nov 20

Danupon Kumpanya*

دنوپل คำปัญญา*

Department of Electrical Engineering, Faculty of Engineering and Architecture, Rajamangala University of Technology Suvarnabhumi

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และสถาปัตยกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

Sattarpoom Thaiparnit

สัตตารพภูมิ ไทยพานิช

Department of Information System and Business Computer, Faculty of Business Administration and Information Technology, Rajamangala University of Technology Suvarnabhumi

สาขาวิชาสารสนเทศศาสตร์และคอมพิวเตอร์ธุรกิจ คณะบริหารธุรกิจและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

*Corresponding Author's, Tel:+6681-850-3236, E-mail: kdanupon@yahoo.com

*ผู้เขียนประสานงาน โทรศัพท์. 081-850-3236 อีเมล: kdanupon@yahoo.com

Abstract

This paper proposes a design of optimal proportional-integral-derivative (PID) controllers for benchmark systems by bat algorithm (BA), one of the most efficient artificial intelligent search techniques. The BA is based on the echolocation behaviour of micro bats with varying pulse emission and loudness. In this work, the BA is conducted to optimally design PID controller for benchmark systems proposes by K. J. Åström and T. Hägglund. As simulation results, it was found that the proposed BA-based PID design giving very satisfied performance for various designing PID controllers.

Keywords: PID Controllers, Benchmark Systems, Bat Algorithm, Artificial Intelligent Search

บทคัดย่อ

บทความนี้นำเสนอการออกแบบตัวควบคุมแบบสัดส่วน แบบปริพันธ์ และแบบอนุพันธ์ (พีไอดี) อย่างเหมาะสมสำหรับระบบเกณฑ์มาตรฐานด้วยขั้นตอนวิธีแบบค้างคาว หนึ่งในเทคนิคการค้นหาแบบปัญญาประดิษฐ์ที่มีประสิทธิภาพมากที่สุด โดยได้แนวคิดจากการปล่อยคลื่นเสียงที่มีความถี่สูงของค้างคาว เพื่อใช้ในการนำทางและค้นหาอาหาร ในบทความนี้ ขั้นตอนวิธีแบบค้างคาวถูกนำไปออกแบบตัวควบคุมพีไอดีอย่างเหมาะสมสำหรับระบบเกณฑ์มาตรฐานที่ได้รับการนำเสนอ โดย K. J. Åström และ T. Hägglund ผลการจำลองที่ได้รับพบว่า การออกแบบตัวควบคุมพีไอดีอิงขั้นตอนวิธีแบบค้างคาวให้สมรรถนะเป็นที่น่าพอใจมากสำหรับการออกแบบตัวควบคุมพีไอดีแบบต่าง ๆ

คำสำคัญ: ตัวควบคุมพีไอดี ระบบเกณฑ์มาตรฐาน ขั้นตอนวิธีแบบค้างคาว การค้นหาแบบปัญญาประดิษฐ์

1. บทนำ

การใช้ตัวควบคุมแบบสัดส่วน แบบปริพันธ์ และแบบอนุพันธ์ หรือ พีไอดี (Proportional-Integral-

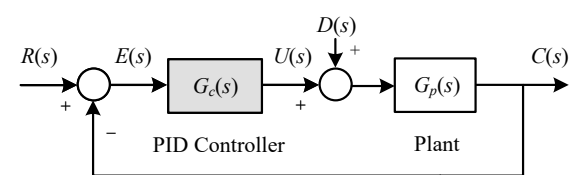
Derivative หรือ PID) ในงานอุตสาหกรรมได้รับการนำเสนอครั้งแรกเมื่อปี ค.ศ. 1939 [1] เนื่องจากใช้สะดวกและทำให้เป็นจริงได้ง่าย ทศวรรษที่ผ่านมา ตัวควบคุม PID ถูกใช้งานมากขึ้น ในระบบควบคุมแบบป้อนกลับ เพื่อให้ได้ค่าพารามิเตอร์ของตัวควบคุม PID ที่เหมาะสม ผู้ออกแบบสามารถใช้วิธีการออกแบบเชิงวิเคราะห์หรือกฎการปรับแต่ง ส่วนใหญ่วิธีการออกแบบเชิงวิเคราะห์จะสมมติว่าทราบแบบจำลองของพลานต์ (Plant) [2] ในขณะที่กฎการปรับแต่งจะสมมติว่าทราบผลตอบสนองของกระบวนการ [3] และทราบแบบจำลองของพลานต์ [4] อย่างไรก็ตามวิธีการออกแบบเชิงวิเคราะห์และกฎการปรับแต่งในทุกครั้งจะมีเงื่อนไขเกี่ยวกับแบบจำลองของพลานต์ อาทิเช่น เวลาไร้ผลสนอง (Dead Time) ซีโร (Zero) และโพล (Pole) ที่เป็นค่าจริง (Real) และค่าสังยุคเชิงซ้อน (Complex Conjugate) โพลเร็วและช้า ตลอดจนโพลไม่เสถียร (Unstable Pole) อื่นๆ เงื่อนไขเหล่านี้ทำให้วิธีการออกแบบและกฎการปรับแต่งไม่ใช่ในแบบทั่วไป โดยในปี ค.ศ. 2000 K. J. Åström และ T. Hägglund [5] ได้นำเสนอการรวบรวมระบบแบบต่างๆ ซึ่งเหมาะสมสำหรับใช้ทดสอบตัวควบคุม PID ระบบที่รวบรวมเหล่านี้ถือเป็นปัญหาเกณฑ์มาตรฐานหรือชุดตัวอย่างทดสอบสำหรับการประเมินวิธีการออกแบบตัวควบคุม PID

จนถึงปัจจุบันเทคนิคของปัญญาประดิษฐ์ (Artificial Intelligence หรือ AI) ได้รับการยอมรับและถูกใช้สำหรับการออกแบบตัวควบคุมในการใช้ควบคุมทางอุตสาหกรรม ตัวอย่างเช่น การออกแบบตัวควบคุม PID แบบปรับตัวด้วยขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm หรือ GA) [6] การออกแบบตัวควบคุม PID แบบปรับแต่งตัวเองด้วย GA [7] การออกแบบตัวควบคุม PID ที่มีความเที่ยงจำกัดด้วย GA [8] ถึงแม้ว่าการออกแบบตัวควบคุม PID ดังกล่าวจะมีประสิทธิภาพในการค้นหาค่าต่ำสุดวงกว้าง (Global Minimum) ของปริภูมิการค้นหาแต่ใช้เวลาการค้นหานานเกินไป ขั้นตอนวิธีแบบค้างคาว (Bat Algorithm หรือ BA) [9] เป็นทางเลือกหนึ่ง ในการแก้ปัญหาทางด้านวิศวกรรม เช่น การออกแบบคานเชื่อมประกอบ (Welded Beam Design) [10] การออกแบบภาชนะรับแรงดัน (Pressure Vessel Design) การออกแบบสปริง (Spring Design) [11] อื่นๆ ดังนั้น ขั้นตอนวิธีของ BA จึงคาดว่าจะขั้นตอนวิธีที่เป็นไปได้ เพื่อให้ได้ตัวควบคุม PID ที่เหมาะสม ในบทความนี้ วิธีของ BA จะถูก

นำไปออกแบบตัวควบคุม PID สำหรับระบบเกณฑ์มาตรฐานที่ได้รับการนำเสนอ โดย K. J. Åström และ T. Hägglund [5] บทความนี้ประกอบไปด้วย 5 ส่วน การออกแบบตัวควบคุม PID ในส่วนที่ 2 ขั้นตอนวิธีแบบค้างคาวในส่วนที่ 3 การออกแบบตัวควบคุม PID ด้วย BA ในส่วนที่ 4 และข้อสรุปในส่วนที่ 5 ตามลำดับ

2. วัสดุ อุปกรณ์ และวิธีการวิจัย

ระบบควบคุมแบบ PID มีโครงสร้างแสดงได้ดังรูปที่ 1 ตัวควบคุม $G_c(s)$ ที่เป็นตัวควบคุม PID จะได้รับสัญญาณค่าความผิดพลาด $E(s)$ และสร้างเป็นสัญญาณควบคุม $U(s)$ ส่งไปควบคุมทางเอาต์พุต $C(s)$ ของพลานต์ $G_p(s)$ เมื่อป้อนสัญญาณอินพุตอ้างอิง $R(s)$ และ $D(s)$ คือ สัญญาณรบกวนจากภายนอก



รูปที่ 1 โครงสร้างของระบบควบคุม PID

ฟังก์ชันถ่ายโอนของตัวควบคุม PID จะแสดงได้ดังสมการที่ (1) โดยที่ K_p เป็นอัตราขยายเชิงสัดส่วน (Proportional Gain) K_i เป็นอัตราขยายเชิงปริพันธ์ (Integral Gain) และ K_d เป็นอัตราขยายเชิงอนุพันธ์ (Derivative Gain) ฟังก์ชันถ่ายโอนวงปิด $C(s)/R(s)$ ของระบบควบคุม PID แสดงได้ดังสมการที่ (2)

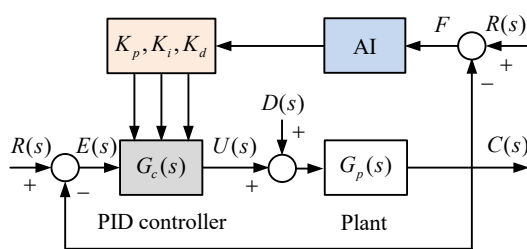
$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \quad (1)$$

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \quad (2)$$

การใช้การค้นหาแบบ AI ในการออกแบบตัวควบคุม PID จะมีโครงสร้างแสดงได้ดังรูปที่ 2 โดยที่ ค่าฟังก์ชันวัตถุประสงค์ F ถูกกำหนดให้เป็นผลรวมของความคลาดเคลื่อนยกกำลังสอง (Sum Squared Error หรือ SSE) ระหว่างอินพุตอ้างอิง $R(s)$ และเอาต์พุต

$C(s)$ เพื่อทำให้น้อยที่สุด แสดงได้ดังสมการที่ (3) การป้อนกลับไปยัง AI ค่าฟังก์ชันวัตถุประสงค์ F จะถูกทำให้ลดค่าลง เพื่อหาค่าพารามิเตอร์ของตัวควบคุม PID อย่างเหมาะสม ดังนั้น ในงานวิจัยนี้ การปรับแต่งแบบ AI จะใช้วิธีของ BA ซึ่งจะอธิบายไว้ในหัวข้อถัดไป

$$F = \sum_{i=1}^N [r(i) - c(i)]^2 \quad (3)$$



รูปที่ 2 โครงสร้างของระบบควบคุม PID บนพื้นฐาน AI

2.1. ขั้นตอนวิธีแบบค้างคาว

ขั้นตอนวิธีแบบค้างคาว (BA) [9] จะอาศัยแนวคิดจากการปล่อยคลื่นเสียงที่มีความถี่สูง (Sonar) ของค้างคาวเพื่อใช้ในการนำทางและค้นหาอาหาร (Echolocation) ค้างคาวเป็นสัตว์เลื้อยคืบด้วยนมขนาดเล็ก ที่มีลักษณะคล้ายหนู แต่มีปีกบินได้เหมือนนก เรียกว่า “นกมีหูหนูมีปีก” ค้างคาวมีดวงตาที่เล็กมาก จมูกและปากของค้างคาวกินแมลงจะมีหน้าที่ช่วยบังคับความถี่เสียงที่มีความถี่สูง เพื่อใช้ในการนำทางและหาอาหาร ค้างคาวไม่ได้อาศัยดวงตาในการมองเห็นขณะบินในที่มืด แต่จะเคลื่อนที่ขณะบินเข้าหาอาหารตามเสียงสะท้อนที่ถูกปล่อยออกมาด้วยคลื่นเสียงที่มีความถี่สูงอย่างเป็นจังหวะและสม่ำเสมอ

คลื่นเสียงที่มีความถี่สูง (คลื่นเหนือเสียง) ดังกล่าวหูของมนุษย์ไม่สามารถได้ยิน แต่หูของค้างคาวซึ่งมีความไวสูงมากจะสามารถตรวจจับคลื่นเหนือเสียงที่ส่งออกไปกระทบกับวัตถุใดๆ แล้วสะท้อนกลับมายังหูของค้างคาวได้ ทำให้ค้างคาวสามารถรับรู้ถึงตำแหน่งและระยะห่างของสิ่งกีดขวางจากเสียงที่สะท้อนกลับมาได้ จึงทำให้ค้างคาวบินได้อย่างคล่องแคล่วว่องไวแม้อยู่ในที่มืดโดยไม่ชนสิ่งกีดขวางรวมทั้งสามารถตรวจสอบตำแหน่งสิ่งที่เป็นอาหารได้อีกด้วย ขั้นตอนวิธีแบบค้างคาวจะอาศัยกลไกที่สำคัญ 3 ข้อ ดังต่อไปนี้ คือ

1) ค้างคาวสามารถค้นหาตำแหน่งของวัตถุโดยใช้เสียงสะท้อนกลับ เพื่อรับรู้ถึงระยะทางความแตกต่างระหว่างอาหาร (เหยื่อ) และสิ่งกีดขวาง กำหนดให้ปริภูมิการค้นหาเป็น d มิติ และค้างคาวตัวที่ i สามารถแทนด้วย x_i ความเร็วของค้างคาวตัวที่ i สามารถแทนด้วย v_i ความถี่พัลส์ของค้างคาวตัวที่ i สามารถแทนด้วย f_i ตำแหน่ง (ผลเฉลย) ใหม่ x'_i และความเร็วใหม่ v'_i ในช่วงเวลา t การปรับปรุงความถี่พัลส์จะอาศัยสมการที่ (4) ถึงสมการที่ (6) เมื่อ $\beta \in [0,1]$ คือ เวกเตอร์จากการสุ่ม (Random Vector) หาได้จากการแจกแจงแบบปกติ (Uniform Distribution) และ x_* คือ ผลเฉลยวงกว้าง (Global Solution) ในรอบปัจจุบัน เมื่อเทียบกับผลเฉลยทั้งหมดในจำนวนค้างคาว n ตัว และ t คือ รอบการค้นหาปัจจุบัน โดยที่การค้นหาเฉพาะที่ (Local Search) ซึ่งผลเฉลยใหม่จะถูกสร้างขึ้นจากผลเฉลยวงกว้างที่ดีที่สุด ในรอบปัจจุบัน ดังสมการที่ (7) เมื่อ $\varepsilon \in [-1,1]$ คือ จำนวนการสุ่ม (Random Number) โดยที่ $A' = (A'_i)$ คือ ความดังของเสียงเฉลี่ยของค้างคาวทั้งหมดในช่วงเวลานั้น

$$f_i = f_{\min} + \beta(f_{\max} - f_{\min}) \quad (4)$$

$$v'_i = v_i^{t-1} + f_i(x_i^{t-1} - x_*) \quad (5)$$

$$x'_i = x_i^{t-1} + v'_i \quad (6)$$

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A' \quad (7)$$

2) ค้างคาวมีทิศทางการบินอย่างไร้แบบแผน (Randomly) ด้วยความเร็ว v_i ในตำแหน่ง x_i ด้วยความถี่ที่ $f_{\min}$ โดยเปลี่ยนแปลงความยาวคลื่น λ และความดังของเสียง A_0 เพื่อค้นหาอาหารและสามารถปรับเปลี่ยนความยาวคลื่นของพัลส์ที่ปล่อยออกมาและปรับอัตราการปล่อยพัลส์ (Pulse Emission หรือ r) โดยที่ $r \in [0,1]$ ปกติขึ้นอยู่กับความระยะห่างของเป้าหมาย การปล่อยพัลส์จะเริ่มจากศูนย์และเพิ่มขึ้นเรื่อย ๆ เมื่อ

ค้ำคาวเข้าใกล้อาหาร (เหยื่อ) เพิ่มมากขึ้น อัตราการปล่อยพัลส์ของค้ำคาวจะถูกปรับเปลี่ยนไปตามสมการที่ (8)

$$r_i^{t+1} = r_i^0 (1 - e^{(-r^t)}) \quad (8)$$

3) ความดังของเสียงสามารถเปลี่ยนแปลงได้หลายรูปแบบ ความดังของเสียงจะเปลี่ยนแปลงจากความดังของเสียงสูงสุด A_0 ไปยังค่าคงที่ต่ำที่สุด $A_{\min}$ นั่นคือความดังจะลดเรื่อย ๆ เมื่อค้ำคาวเข้าใกล้อาหารเพิ่มมากขึ้น ความดังของเสียงจะถูกปรับเปลี่ยนตามสมการที่ (9)

$$A_i^{t+1} = \alpha A_i^t \quad (9)$$

ขั้นตอนวิธีแบบค้ำคาว (BA) [9] สามารถสรุปได้ดังนี้

ขั้นตอนที่ 1 กำหนดจำนวนประชากรค้ำคาว i และจำนวนรอบการค้นหาสูงสุด $t_{\max}$

ขั้นตอนที่ 2 กำหนดประชากรค้ำคาวเริ่มต้น x_i พร้อมกับความเร็ว v_i ความถี่พัลส์ f_i อัตราการปล่อยพัลส์ r_i และความดังของเสียง A_i ให้กับค้ำคาวแต่ละตัวโดยวิธีการสุ่มเพื่อใช้เป็นค่าเริ่มต้น

ขั้นตอนที่ 3 สร้างผลเฉลยใหม่โดยการปรับค่าความถี่พัลส์ f_i และปรับปรุงค่าความเร็ว v_i และตำแหน่ง x_i ของค้ำคาวแต่ละตัว โดยใช้สมการที่ (4) ถึงสมการที่ (6) และคำนวณหาค่าฟังก์ชันวัตถุประสงค์ของค้ำคาวทุกตัว

ขั้นตอนที่ 4 กำหนดเลขสุ่ม ถ้า $rand < r_i$ ให้เลือกผลเฉลยที่ดีที่สุดตามฟังก์ชันวัตถุประสงค์ และสร้างผลเฉลยวงแคบเฉพาะที่ (Local Solution) รอบๆ ผลเฉลยที่ดีที่สุดที่เลือกโดยใช้สมการที่ (7)

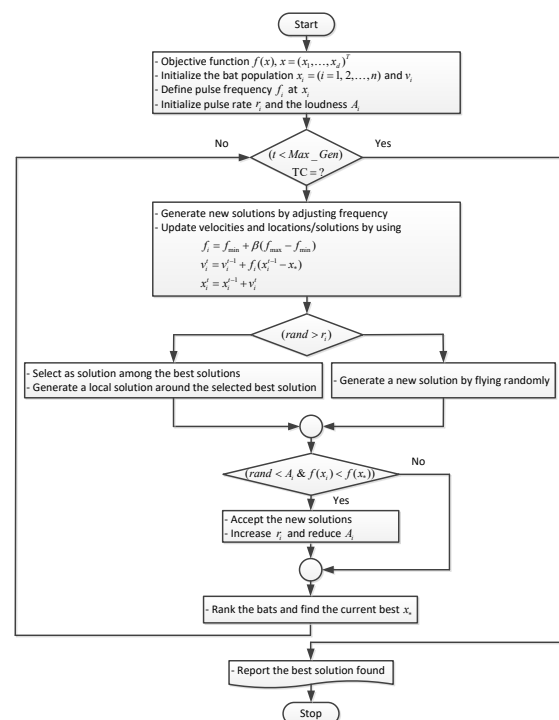
ขั้นตอนที่ 5 สร้างผลเฉลยใหม่ด้วยการสุ่มค้ำคาวใหม่

ขั้นตอนที่ 6 กำหนดเลขสุ่ม ถ้า $rand < A_i$ และ $f(x_i) < f(x_*)$ ให้ยอมรับผลเฉลยใหม่และเพิ่มอัตราการปล่อยพัลส์ของค้ำคาวแต่ละตัว r_i และลดความดังของเสียงของค้ำคาวแต่ละตัว A_i โดยใช้สมการที่ (8) และสมการที่ (9) ตามลำดับ

ขั้นตอนที่ 7 จัดเรียงลำดับค้ำคาวตามฟังก์ชันวัตถุประสงค์และหาผลเฉลยวงกว้างที่ดีที่สุดในรอบปัจจุบัน

ขั้นตอนที่ 8 ทำซ้ำขั้นตอนที่ 3 ถึงขั้นตอนที่ 7 จนครบรอบการทำงานซ้ำ

ดังนั้น ค่าพารามิเตอร์ของ BA สำหรับการประยุกต์ใช้งานทั่วไป กำหนดให้ n คือ จำนวนของค้ำคาว β คือ เวกเตอร์จากการสุ่ม ε คือ จำนวนการสุ่ม และ r คือ อัตราการปล่อยพัลส์ สามารถแสดงขั้นตอนวิธีของ BA ได้ด้วยแผนภูมิ (Flowchart) ของ BA ดังแสดงในรูปที่ 3 เมื่อ TC คือ เกณฑ์ยุติการค้นหา (Termination Criteria)



รูปที่ 3 แผนภูมิของ BA [9]

2.2. การออกแบบตัวควบคุม PID ด้วย BA

จากรูปที่ 2 การปรับแต่งค่าของ AI จะใช้วิธีของ BA เพื่อที่จะลดค่าฟังก์ชันวัตถุประสงค์ F ในสมการที่ (3) จนกระทั่งยุติการค้นหา (Termination Criteria หรือ TC) จำนวนรอบสูงสุด (Max_Gen) จะถูกกำหนดให้เป็น TC เพื่อประเมินสมรรถนะของวิธีการที่ออกแบบตัวควบคุม PID ด้วย BA จะถูกนำไปใช้กับระบบเกณฑ์มาตรฐานที่ถูกเลือกจากการนำเสนอโดย K. J. Åström และ T. Hägglund [5] ซึ่งระบบเหล่านี้จะยากต่อการออกแบบตัวควบคุม PID โดยวิธีการออกแบบชนิดดั้งเดิมหรือกฎการปรับแต่งค่าพารามิเตอร์ K_p , K_i และ K_d ของตัวควบคุม PID จะถูกกำหนดจากกรอบการทำงานได้ดังสมการที่ (10)

ตารางที่ 1 แบบจำลองของพลาเน็ตทั้ง 9 ชนิด กับข้อกำหนดในการออกแบบ

ลำดับที่	แบบจำลองของพลาเน็ต	ข้อกำหนดในการออกแบบ			
		t_{r_max} (sec.)	$P.O._{max}$ (%)	t_{s_max} (sec.)	E_{ss_max}
1.	$G_{p1}(s) = \frac{1}{(s+1)(\alpha s+1)(\alpha^2 s+1)(\alpha^3 s+1)}, \alpha = 0.5$	1.00	10.00	2.50	0.00
2.	$G_{p2}(s) = \frac{1}{(s+1)^4}$	3.50	10.00	10.00	0.00
3.	$G_{p3}(s) = \frac{1-\alpha s}{(s+1)^3}, \alpha = 0.5$	3.50	10.00	10.00	0.00
4.	$G_{p4}(s) = \frac{1}{(Ts+1)} e^{-s}, T = 10$	3.50	10.00	10.00	0.00
5.	$G_{p5}(s) = \frac{1}{(Ts+1)^2} e^{-s}, T = 10$	7.00	10.00	20.00	0.00
6.	$G_{p6}(s) = \frac{100}{(s+10)^2} \left(\frac{1}{s+1} + \frac{0.5}{s+0.05} \right)$	0.25	10.00	1.50	0.00
7.	$G_{p7}(s) = \frac{(s+6)^2}{s(s+1)^2(s+36)}$	0.50	10.00	1.50	0.00
8.	$G_{p8}(s) = \frac{\omega_0^2}{(s+1)(s^2+2\zeta\omega_0 s+\omega_0^2)}, \omega_0 = 1, \zeta = 0.1$	2.50	20.00	10.00	0.00
9.	$G_{p9}(s) = \frac{1}{(s^2-1)}$	0.10	15.00	0.50	0.00

Min F

subject to $t_r \leq t_{r_max}, P.O. \leq P.O._{max},$ (10)

$t_s \leq t_{s_max}, E_{ss} \leq E_{ss_max}$

ข้อกำหนดของสมรรถนะที่พึงประสงค์ในสมการที่ (10) เมื่อ t_r คือ ช่วงเวลาขึ้น (Rise Time) $P.O.$ คือ เปอร์เซ็นต์ค่าพุ่งเกิน (Percent Overshoot) t_s คือ ช่วงเวลาเข้าที่ (Setting Time) E_{ss} คือ ค่าผิดพลาดที่สถานะอยู่ตัว (Steady-state Error) t_{r_max} คือ ช่วงเวลาขึ้นสูงสุด (Maximum Rise Time) $P.O._{max}$ คือ เปอร์เซ็นต์ค่าพุ่งเกินสูงสุด (Maximum Percent

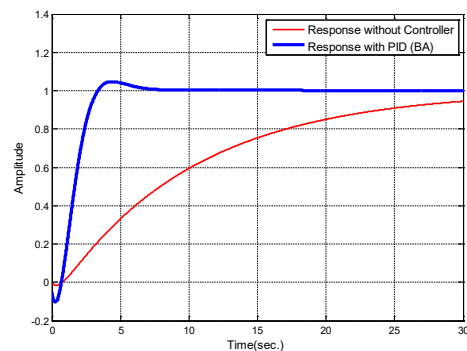
Overshoot) t_{s_max} คือ ช่วงเวลาเข้าที่สูงสุด (Maximum

Setting Time) และ E_{ss_max} คือ ค่าผิดพลาดที่สถานะ

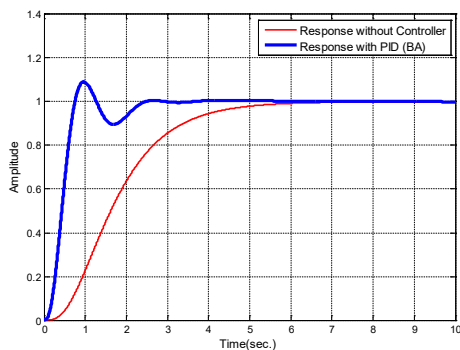
อยู่ตัวสูงสุด (Maximum Steady-state Error) ซึ่งถูกกำหนดเป็นเงื่อนไขอสมการ (Inequality Constraints) ของปัญหาการหาค่าเหมาะที่สุด ค่าตัวเลขของข้อกำหนดในการออกแบบจะต้องสอดคล้องกับระบบมาตรฐานที่กำหนด

จากตารางที่ 1 พบว่าแบบจำลองของพลาเน็ตทั้ง 9 ชนิด กับข้อกำหนดในการออกแบบที่สอดคล้องกัน ถูกนำไปใช้ในการออกแบบตัวควบคุม PID โดยใช้ BA สำหรับระบบเกณฑ์มาตรฐานซึ่งประกอบไปด้วย 1) ระบบอันดับสี่ (Fourth Order System) 2) ระบบโพลพหุคูณ (Multiple poles System) 3) ระบบซีโรครึ่งขวา (Right Half Zero System) 4) ระบบอันดับหนึ่งกับเวลาไร้ผลสนอง (First

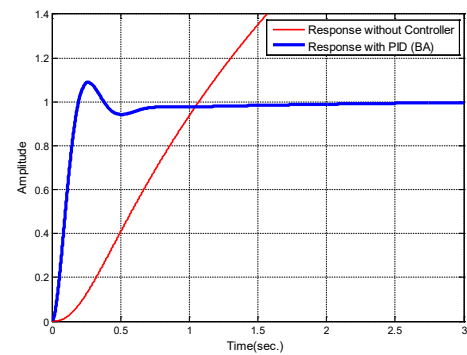
Order System with Dead Time) 5) ระบบอันดับสองกับเวลาไร้ผลสนอง (Second Order System with Dead Time) 6) แบบเร็วและช้า (Fast and Slow Modes) 7) ระบบเสถียรมีเงื่อนไข (Conditionally Stable System) 8) ระบบแกว่ง (Oscillatory System) 9) ระบบไม่เสถียร (Unstable System) ตามลำดับ เมื่อใช้วิธีของ BA จะต้องระบุพิสัยของค่าพารามิเตอร์ที่จะค้นหาในปริภูมิการค้นหา (Search Space) และการระบุพิสัยของค่าพารามิเตอร์ของตัวควบคุม PID ที่จะค้นหาแสดงได้ดังตารางที่ 2



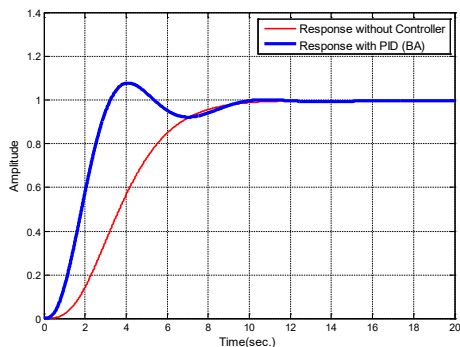
รูปที่ 8 ผลตอบสนองของ $G_{p4}(s)$



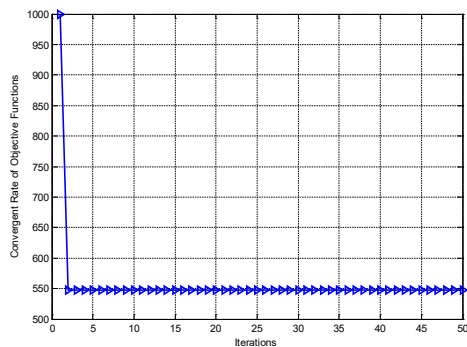
รูปที่ 4 ผลตอบสนองของ $G_{p1}(s)$



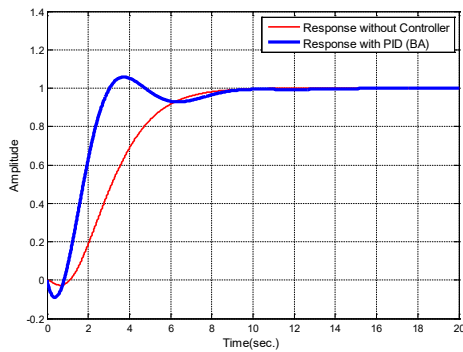
รูปที่ 10 ผลตอบสนองของ $G_{p6}(s)$



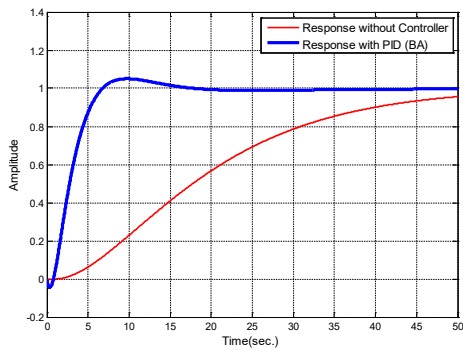
รูปที่ 6 ผลตอบสนองของ $G_{p2}(s)$



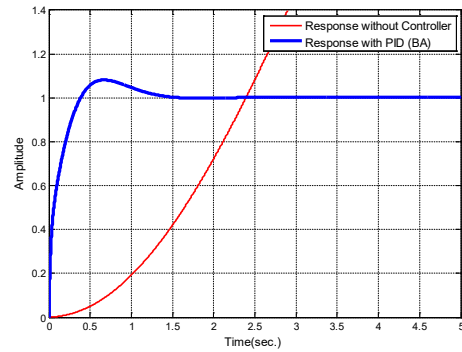
รูปที่ 5 การลู่เข้าหาผลเฉลยของ $G_{p1}(s)$



รูปที่ 7 ผลตอบสนองของ $G_{p3}(s)$



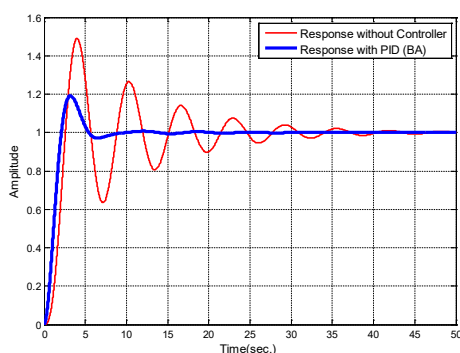
รูปที่ 9 ผลตอบสนองของ $G_{p5}(s)$



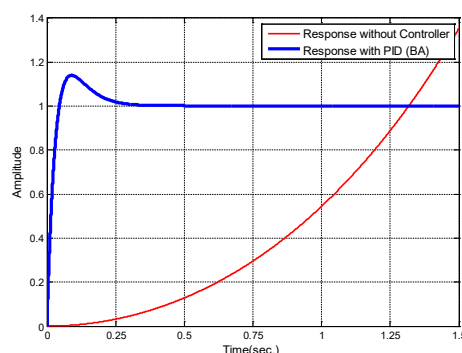
รูปที่ 11 ผลตอบสนองของ $G_{p7}(s)$

ตารางที่ 2 การระบุพิสัยของค่าพารามิเตอร์ของตัวควบคุม PID ที่ออกแบบด้วยวิธีของ BA

ลำดับที่	ปฏิกิริยาการค้นหาค่า			ลำดับที่	ปฏิกิริยาการค้นหาค่า		
	K_p	K_i	K_d		K_p	K_i	K_d
1.	[0, 10]	[0, 10]	[0, 10]	6.	[0, 10]	[0, 5]	[0, 5]
2.	[0, 5]	[0, 5]	[0, 5]	7.	[0, 20]	[0, 5]	[0, 50]
3.	[0, 5]	[0, 5]	[0, 5]	8.	[0, 5]	[0, 5]	[0, 5]
4.	[0, 10]	[0, 5]	[0, 5]	9.	[400, 500]	[0, 50]	[0, 50]
5.	[0, 10]	[0, 5]	[0, 50]				



รูปที่ 12 ผลตอบสนองของ $G_{p8}(s)$



รูปที่ 13 ผลตอบสนองของ $G_{n9}(s)$

ขั้นตอนวิธีของ BA จะได้รับการพัฒนาเป็นโปรแกรมการค้นหาค้นด้วยโปรแกรม MATLAB/SIMULINK [12] บนเครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลกลางขนาด 2 แกน ที่มีสัญญาณนาฬิกา 2 กิกะเฮิรตซ์ และมีหน่วยความจำหลักชนิด DDR3 ขนาดความจุ 3 กิกะไบต์ ที่คาบเวลาสุ่ม (Sampling Period) 1 มิลลิวินาที การหาค่าพารามิเตอร์ของ BA ในเบื้องต้นทำได้ดังนี้ จำนวนของค้างคาว $n = 20$ ความถี่ $f_{\min} = 0.0$ และ $f_{\max} = 2.0$ ความดังของเสียง $A_0 = 0.5$ อัตราการปล่อยพัลส์ $r_0 = 0.5$ และ $\alpha = \gamma = 0.9$ จำนวนรอบสูงสุด (Max_Gen) = 1000 กำหนดให้เป็น TC การออกแบบจะดำเนินการค้นหาทั้งหมด 40 ตัวอย่าง กับการสุ่มผลเฉลยเริ่มต้นแบบต่างๆ เพื่อได้รับผลเฉลยที่ดีที่สุด ภายใต้ข้อกำหนดของสมรรถนะที่พึงประสงค์ในสมการที่ (10) กับการระบุพิสัยของค่าพารามิเตอร์ของตัวควบคุม PID ให้สอดคล้องกับปฏิกิริยาการค้นหาค้นหา (ระบบจากกรอบการทำงานจริง) ดังตารางที่ 2

3.ผลการทดลอง

ผลตอบสนองของ $G_{p1}(s)$ แสดงในรูปที่ 4 ในขณะที่ การลู่เข้าหาผลเฉลย $G_{p1}(s)$ แสดงในรูปที่ 5 ค่าฟังก์ชันวัตถุประสงค์ F ทั้งหมด 40 ตัวอย่าง มีเพียงตัวอย่างเดียวที่ลู่เข้าหาผลเฉลยวงกว้างได้เร็วและมีค่าฟังก์ชันวัตถุประสงค์ F น้อยที่สุด ซึ่งยืนยันได้ว่าตัวควบคุม PID ที่ออกแบบด้วยวิธีของ BA สามารถหาค่าพารามิเตอร์ของตัวควบคุมได้อย่างเหมาะสม ในการเฝ้าติดตามอินพุต (Input Tracking) สำหรับผลตอบสนองของ $G_{p2}(s)$ ถึง $G_{p9}(s)$ แสดงในรูปที่ 6 ถึง รูปที่ 13 ตามลำดับ การลู่เข้าหาผลเฉลยวงกว้างในแบบต่างๆ จะถูกไล่เว้นไม่แสดงไว้ เนื่องจากมีรูปแบบคล้ายกันกับ $G_{p1}(s)$ ที่แสดงในรูปที่ 5 จากรูปที่ 6 จนถึง รูปที่ 13 จะสังเกตได้ว่าตัวควบคุม PID ที่ออกแบบด้วยวิธีของ BA สามารถหาค่าพารามิเตอร์ของตัวควบคุม PID ได้อย่างเหมาะสม ในการเฝ้าติดตามอินพุต ค่าพารามิเตอร์ของตัวควบคุมและสมรรถนะของระบบควบคุม PID สามารถสรุปได้ดังตารางที่ 3

ตารางที่ 3 ค่าพารามิเตอร์และสมรรถนะของระบบควบคุม PID ที่ออกแบบด้วยวิธีของ BA

ลำดับที่	ตัวควบคุมพีไอดี			สมรรถนะของระบบควบคุม				ฟังก์ชัน วัตถุประสงค์ F
	K_p	K_i	K_d	t_r (sec.)	$P.O.$ (%)	t_s (sec.)	E_{ss}	
1.	3.7480	2.1983	2.2256	0.7518	9.13	2.2702	0.00	547.3181
2.	1.8020	0.5142	1.7980	3.3967	5.42	8.8003	0.00	2076.1448
3.	1.5570	0.4944	0.9850	3.0379	5.78	8.6226	0.00	2110.4472
4.	5.8140	0.6129	0.5166	3.3517	4.75	6.1992	0.00	1929.2262
5.	7.4273	0.3322	28.3031	6.7995	5.10	14.4950	0.00	3112.0972
6.	8.3050	1.4986	1.1975	0.1940	8.94	1.2152	0.00	183.9233
7.	17.007	0.8781	22.267	0.3847	8.20	1.2350	0.00	169.9037
8.	0.2366	1.2087	1.2161	2.0980	19.20	7.5265	0.00	1776.5384
9.	494.1433	45.1189	43.7620	0.0452	14.00	0.2490	0.00	60.1943

4.อภิปรายผลและสรุปผลการศึกษา

4.1. สรุปผล

บทความนี้นำเสนอการออกแบบตัวควบคุม PID ด้วย BA สำหรับระบบเกณฑ์มาตรฐานที่ได้รับการนำเสนอ โดย K. J. Åström และ T. Häggglund [5] ระบบเหล่านี้ถือเป็นปัญหาเกณฑ์มาตรฐานสำหรับการประเมินวิธีการออกแบบตัวควบคุม PID วิธีของ BA เป็นวิธีการออกแบบที่ยืดหยุ่นมากกว่าวิธีการออกแบบเชิงวิเคราะห์หรือกฎการปรับแต่ง ผลการจำลองพบว่า วิธีของ BA บนพื้นฐานวิธีการออกแบบตัวควบคุม PID ที่นำเสนอให้สมรรถนะเป็นที่น่าพอใจสำหรับการออกแบบตัวควบคุมพีไอดีแบบต่าง ๆ

5.องค์ความรู้ใหม่ที่ได้จากงานวิจัย

งานวิจัยนี้ นำเสนอการประยุกต์ขั้นตอนวิธีแบบค้างควา หนึ่งในเทคนิคการค้นหาแบบ AI ที่มีประสิทธิภาพมากที่สุด ในการออกแบบตัวควบคุม PID โดยทดสอบกับแบบจำลองของพลาตันต์ทั้ง 9 ชนิด ที่นำเสนอโดย K. J. Åström และ T. Häggglund [5] เพื่อให้ได้ค่าพารามิเตอร์ของตัวควบคุม PID ที่เหมาะสม ซึ่งเป็นวิธีการออกแบบที่มีความยืดหยุ่นและมีประสิทธิภาพมากกว่าการออกแบบเชิงวิเคราะห์หรือกฎการปรับแต่ง

6.เอกสารอ้างอิง

- [1] S. Bennett, "Development of the PID controller," IEEE Control System Magazine, pp.58–65, 1994.
- [2] R. C. Dorf, "Modern Control Systems," 10th edn. New Jersey Prentice-Hall, 2005.
- [3] J. G. Ziegler and N. B. Nichols, "Optimum Settings for Automatic Controllers," Trans. ASME, vol. 64, pp. 759-768, 1942.
- [4] A. O. Dwyer, Handbook of PI and PID Controller Tuning Rules, Imperial College Press, 2003.
- [5] K. J. Åström and T. Häggglund, "Benchmark Systems for PID Control," IFAC Digital Control: Past, Present and Future of PID Control, Terrassa, Spain, pp.165–166, 2000.
- [6] S. Mehrdad and C. Greg, "An Adaptive PID Controller based on Genetic Algorithm Processor," Proc. IEE Conf. on Genetic

- Algorithm in Engineering System, pp. 88–93, 1995.
- [7] Y. Mitsukura, T. Yamamoto and M. Kaneda, “A Design of Self-tuning PID Controller using a Genetic Algorithm,” Proc. American Control Conference, pp. 1361–1365, 1999.
- [8] J. F. Whidborne, “A Genetic Algorithm Approach to Design Finite–Precision PID ControllerStructures,” Proc. American Control Conference, pp. 4338–4342, 1999.
- [9] X. S. Yang, “A new metaheuristic bat-inspired an algorithm,” in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, 284, Springer, pp. 65–74, 2010.
- [10] X. S. Yang, “Bat algorithm for multi-objective optimization,” Int. J. Bio-Inspired Computation, vol. 3 no. 5, pp. 267–274, 2011.
- [11] P. W. Tsai, J. S. Pan, B. Y. Liao, M. J. Tsai, and V. Istanda, “Bat algorithm inspired algorithm for solving numerical optimization problems,” Applied Mechanics and Materials, vol. 148–149, pp. 134–137, 2011.
- [12] MATLAB/SIMULINK User’s Guide, The Math Works Inc., Natick, MA, 1998..