

การควบคุมความเร็วมอเตอร์ BLDC โดยใช้เซนเซอร์ฮอลล์ร่วมกับคอนโทรลเลอร์

$PI^\lambda D^\mu$ เชิงเศษส่วน ที่ออกแบบด้วยอัลกอริทึมค่างคาวแบบปรับปรุง (MBA)

Hall Sensor-Based Speed Control of a BLDC Motor Using a Fractional-Order

$PI^\lambda D^\mu$ Controller Optimized by the Modified Bat Algorithm (MBA)

ชูเกียรติ ศิริ

ภาควิชาเทคโนโลยีไฟฟ้า สถาบันการอาชีวศึกษาภาคกลาง 1 chookiat9999@gmail.com

บทคัดย่อ

งานวิจัยนี้นำเสนอการพัฒนาตัวควบคุมความเร็วแบบวงปิดสำหรับมอเตอร์ BLDC โดยประยุกต์ใช้ตัวควบคุมแบบอนุพันธ์-อินทิกรัลลำดับเศษส่วน (Fractional Order Integral-Derivative Controller: $PI^\lambda D^\mu$) ซึ่งสามารถปรับการทำงานของมอเตอร์ให้อยู่ใกล้ค่าความเร็วอ้างอิงได้อย่างแม่นยำ และให้ผลลัพธ์ที่มีเสถียรภาพและความทนทานสูงกว่าการควบคุมด้วยตัวควบคุม PID แบบดั้งเดิม ในขั้นตอนการวิจัย มอเตอร์ BLDC ได้รับการจำลองการทำงานด้วยโปรแกรม MATLAB/Simulink เพื่อนำไปสู่การหาค่าพารามิเตอร์ของตัวควบคุม $PI^\lambda D^\mu$ ได้แก่ $K_p, K_i, K_d, \lambda, \mu$ โดยอาศัยการปรับค่าที่เหมาะสมผ่านอัลกอริทึมค่างคาวแบบปรับปรุง (Improved Bat Algorithm) การประเมินผลประสิทธิภาพของกลยุทธ์การควบคุมที่นำเสนอได้ดำเนินการผ่านการจำลองด้วยคอมพิวเตอร์ รวมถึงการทดสอบเพิ่มเติมภายใต้สภาวะที่มีการเปลี่ยนแปลงค่าพารามิเตอร์บางประการของมอเตอร์ เพื่อวิเคราะห์ความทนทานของระบบ ผลการทดลองเชิงจำลองแสดงให้เห็นว่า กลยุทธ์การควบคุมที่นำเสนอมีความยืดหยุ่นและมีเสถียรภาพสูงกว่าการควบคุมด้วย PID แบบดั้งเดิมอย่างมีนัยสำคัญ ทั้งนี้ชี้ให้เห็นถึงศักยภาพของการประยุกต์ใช้งานจริงในระบบขับเคลื่อนที่ต้องการความแม่นยำและความน่าเชื่อถือสูง

คำสำคัญ: ตัวควบคุม $PI^\lambda D^\mu$ ลำดับเศษส่วน มอเตอร์ BLDC การหาค่าที่เหมาะสมที่สุดโดยการดัดแปลงอัลกอริทึมค่างคาว

Abstract

This study presents the development of a closed-loop speed controller for BLDC motors by implementing a Fractional Order Integral-Derivative Controller ($PI^\lambda D^\mu$). The proposed controller can regulate the motor operation to closely track the reference speed with high accuracy, providing superior stability and robustness compared to conventional PID controllers. In this study, the BLDC motor was modeled and simulated using MATLAB/Simulink to determine the parameters of the $PI^\lambda D^\mu$ controller, including $K_p, K_i, K_d, \lambda, \mu$. The optimal values were obtained using an Improved Bat Algorithm. The performance of the proposed control strategy was evaluated through computer simulations, as well as additional tests under conditions where

certain motor parameters were varied, in order to assess the robustness of the system. The simulation results demonstrate that the proposed control strategy exhibits significantly higher flexibility and stability compared to conventional PID control. These findings highlight the potential for practical applications in drive systems that require high precision and reliability.

Keywords: Fractional-Order $PI^\lambda D^\mu$, BLDC Motor, Metaheuristic Search, Modified Bat Algorithm

1. บทนำ

มอเตอร์แบบไม่มีแปรงถ่าน (BLDC) มีการนำมาใช้งานอย่างแพร่หลาย เช่น ระบบอัตโนมัติในอุตสาหกรรม, การแพทย์, การขับเคลื่อนไฟฟ้า, เครื่องใช้ไฟฟ้าภายในบ้าน, อากาศยาน, ยานพาหนะบนถนน, เครื่องบิน, อุปกรณ์ทางการแพทย์, ฮาร์ดดิสก์ เป็นต้น มอเตอร์ชนิดนี้มีข้อดี ได้แก่ ความเชื่อถือได้สูง, ประสิทธิภาพดี, ความหนาแน่นของกำลังสูง, น้ำหนักเบา, ต้องการการบำรุงรักษาน้อย และรองรับช่วงความเร็วที่กว้าง ทำให้มอเตอร์ BLDC ได้รับความนิยมเพิ่มมากขึ้น [1] มอเตอร์ BLDC จะทำงานโดยในแต่ละช่วงเวลาจะมีเพียงสองเฟส ที่มีกระแสไหลอยู่ในขณะที่เฟสที่สามจะเป็นสถานะลอย ซึ่งจะสัมพันธ์กับกระแสและแรงเคลื่อนไฟฟ้าย้อนกลับ (Back EMF) [2] โดยมีการตรวจจับตำแหน่งของโรเตอร์สองวิธี ได้แก่ ใช้เซนเซอร์ (hall sensor) และวิธีที่ไม่ใช้เซนเซอร์ (sensorless) มอเตอร์ BLDC เป็นระบบที่มีหลายตัวแปรและไม่เป็นเชิงเส้น [3] ดังนั้นจึงเป็นเรื่องยากที่จะได้ผลการควบคุมที่แม่นยำและเชื่อถือได้โดยใช้วิธีการควบคุมแบบเชิงเส้นตามแบบดั้งเดิม ในงานวิจัยนี้ได้มีการใช้ตัวควบคุม PID อันดับเศษส่วน (Fractional Order PID Controller) เพื่อให้ได้ผลการควบคุมที่แม่นยำและเชื่อถือได้สูงกว่าวิธีการควบคุมแบบดั้งเดิม

2. การออกแบบระบบควบคุมลำดับเศษส่วน

2.1 คณิตศาสตร์ลำดับเศษส่วน

แคลคูลัสลำดับเศษส่วน (Fractional Order Calculus) เป็นแขนงหนึ่งของทฤษฎีการวิเคราะห์ทางคณิตศาสตร์ ซึ่งนักคณิตศาสตร์ศึกษาการหาอนุพันธ์และอินทิกรัลของฟังก์ชันในลำดับที่ไม่เป็นจำนวนเต็ม (ลำดับเศษส่วน) แคลคูลัสลำดับเศษส่วนทำการประเมิน $(\frac{dy^n}{dt^n})$ และ n อินทิกรัล

ซ้อนหลายชั้น โดยที่ n เป็นจำนวนเศษส่วน มีนิยามหลายแบบที่ใช้ในการอธิบายฟังก์ชันลำดับเศษส่วน [4] นิยามที่ได้รับการยอมรับอย่างแพร่หลายประกอบด้วย สูตรอินทิกรัลของโคชี (Cauchy integral formula), นิยามของกรุนวาลด์-เลตนิคอฟ (Grunwald-Letnikov) และนิยามของรีมันน์-ลีอูวิลล์ (Riemann-Liouville) นิยามของรีมันน์-ลีอูวิลล์ (Riemann-Liouville) เป็นนิยามที่มีการใช้งานมากที่สุดในแคลคูลัสลำดับเศษส่วน ซึ่งการอินทิกรัลลำดับเศษส่วนถูกนิยามไว้ดังนี้[4]

$$aDt^{-\varphi} f(t) = \frac{1}{\Gamma(\varphi)} \int_a^t (t-\tau)^{\varphi-1} f(\tau) d\tau \quad (1)$$

โดยที่ $aDt^{-\varphi}$ คือ โอเปอเรเตอร์ดิฟเฟอเรนเชียล-อินทิกรัล มันเป็นการผสมผสานระหว่างการดำเนินการอนุพันธ์และอินทิกรัล ซึ่งถูกใช้อย่างแพร่หลายในแคลคูลัสลำดับเศษส่วน นิยามของรีมันน์-ลีอูวิลล์ แสดงดังนี้

$$aDt^{-\varphi} = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \alpha > 0 \\ 1 & \alpha = 0 \\ \int_a^t (d\tau)^{-\alpha} & \alpha < 0 \end{cases} \quad (2)$$

โดยที่ φ แทนลำดับจริงของโอเปอเรเตอร์ดิฟเฟอเรนเชียล-อินทิกรัล ($0 < \varphi < 1$), a คือระยะเวลาริเริ่มต้น, $\Gamma(\varphi)$ คือฟังก์ชันแกมมาของออยเลอร์ และ t คือพารามิเตอร์ที่ใช้ในการดำเนินการโอเปอเรเตอร์ดิฟเฟอเรนเชียล-อินทิกรัล การแปลงลาปลาซของอนุพันธ์ลำดับเศษส่วนของ $f(t)$ นิยามได้ดังนี้[4]

$$L[Dt^\varphi f(t)] = s^\varphi L[f(t)] - \sum_{k=1}^n s^k [Dt^{\varphi-k-1} f(t)] t=0 \quad (3)$$

2.2 ระบบ PID ลำดับเศษส่วน

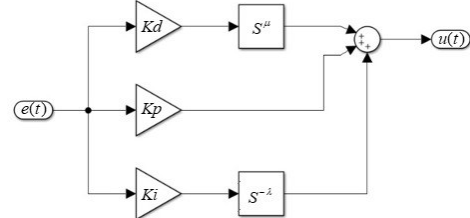
ระบบควบคุมลำดับเศษส่วนถูกอธิบายด้วยสมการเชิงอนุพันธ์ลำดับเศษส่วน แคลคูลัสลำดับเศษส่วนอนุญาตให้อนุพันธ์และอินทิกรัล มีลำดับเป็นจำนวนจริงใด ๆ ก็ได้ [5] ตัวควบคุม $FOPID$ หรือ $PI^\lambda D^\mu$ เป็นการขยายจากตัวควบคุม PID แบบดั้งเดิม โดยมีการเพิ่มพารามิเตอร์อินทิกรัล λ และอนุพันธ์ μ ซึ่งมีค่าเป็นลำดับเศษส่วน เพื่อเพิ่มความยืดหยุ่นและทำให้ระบบไวต่อการเปลี่ยนแปลงของพารามิเตอร์น้อยลงสมการเชิงอนุพันธ์ของตัวควบคุม $PI^\lambda D^\mu$ สามารถอธิบายได้ดังนี้[5]

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^\mu e(t) \quad (4)$$

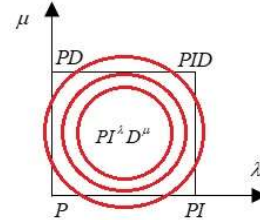
โดยที่ $e(t)$ คือค่าความคลาดเคลื่อนระหว่างค่าตัวแปรเอาต์พุตของกระบวนการที่วัดได้กับจุดที่ต้องการ (set point) และ $u(t)$ เป็นสัญญาณควบคุมเอาต์พุตของระบบ ดังนั้นฟังก์ชันถ่ายโอนของตัวควบคุมในโดเมนลาปลาซแสดงดังนี้

$$G(s) = K_p + K_i s^{-\lambda} + K_d s^\mu \quad (5)$$

รูปที่ 1 แสดงแผนภาพบล็อกของตัวควบคุม $FOPID$ โดยสามารถได้ตัวควบคุม PID แบบดั้งเดิม จากการกำหนดให้ $\lambda = \mu = 1$ นอกจากนี้ เมื่อ $\lambda = 1$ (หรือ 0) และ $\mu = 0$ (หรือ 1) จะได้ตัวควบคุม PI (หรือ PD) แบบปกติ ดังแสดงในรูปที่ 2 ตัวควบคุม $PI^\lambda D^\mu$ แบบลำดับเศษส่วนเป็นการทั่วไปของตัวควบคุม PID ลำดับจำนวนเต็ม และขยายจากจุดหนึ่งไปสู่ระนาบ[11]



รูปที่ 1 แสดงแผนภาพบล็อกของตัวควบคุม $FOPID$



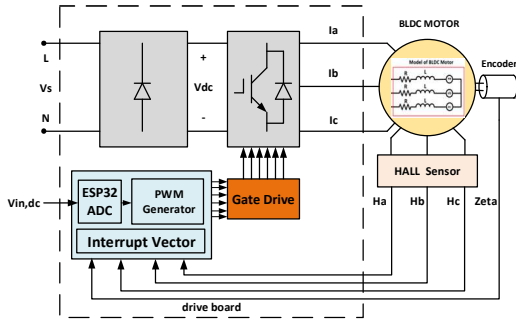
รูปที่ 2 ตัวควบคุม $PI^\lambda D^\mu$ แบบลำดับเศษส่วน

$$\left. \begin{aligned} K_p \text{ min} = 0, \text{ max} = 20 \\ K_i \text{ min} = 0, \text{ max} = 200 \\ \lambda \text{ min} = 0, \text{ max} = 2.9999 \\ K_d \text{ min} = 0, \text{ max} = 20 \\ \mu \text{ min} = 0, \text{ max} = 2.9999 \end{aligned} \right\} \quad (6)$$

เมื่อกำหนดให้ค่า $\mu = 1$ และ $\lambda = 1$ ฟังก์ชันวัตถุประสงค์ (Objective function) ของงานวิจัยนี้เป็นตัวควบคุม ลำดับจำนวนเต็ม ในกรณี μ อยู่ระหว่าง 0 ถึง 2.9999 และ λ อยู่ระหว่าง 0 ถึง 2.9999 ฟังก์ชันวัตถุประสงค์ (Objective function) เป็นตัวควบคุมลำดับเศษส่วน ข้อกำหนดขอบเขตการค้นหาคำตอบงานวิจัยนี้แสดงในสมการที่ (6)

2.3 แบบจำลองลำดับเศษส่วนของมอเตอร์ BLDC

มอเตอร์ BLDC แบบสามเฟสมีขดลวดสามชุดอยู่ในสเตเตอร์ ซึ่งจะรับพลังงานจากอินเวอร์เตอร์ที่ทำหน้าที่แปลงแรงดันไฟฟ้า DC ให้เป็นแรงดันไฟฟ้า AC แบบสามเฟส โดยมีความถี่ที่สอดคล้องกับตำแหน่งการหมุนของโรเตอร์โดยที่โรเตอร์มีขั้วแม่เหล็กถาวรติดตั้งอยู่บนแกนหมุน (shaft) ค่าความเหนี่ยวนำร่วมระหว่างสเตเตอร์และโรเตอร์มีลักษณะเป็นรูปทรงสี่เหลี่ยมคางหมู ซึ่งทำให้เกิดแรงเคลื่อนไฟฟ้าย้อนกลับ (Back EMF) ที่มีรูปร่างเป็นสี่เหลี่ยมคางหมูในขดลวดของสเตเตอร์ [6] รูปที่ 3 แสดงวงจรสมมูลของมอเตอร์ BLDC



รูปที่ 3 แสดงวงจรขับและวงจรสมมุติของมอเตอร์ BLDC

สมการแรงดันของแต่ละเฟสในสเตเตอร์สามารถเขียนได้ดังนี้

$$v_a = R i_a + (L-M) \frac{d i_a}{dt} + e_a \quad (7)$$

$$v_b = R i_b + (L-M) \frac{d i_b}{dt} + e_b \quad (8)$$

$$v_c = R i_c + (L-M) \frac{d i_c}{dt} + e_c \quad (9)$$

โดยที่ V_{abc} คือ แรงดันเฟส, I_{abc} คือ กระแสในเฟสของสเตเตอร์, R คือ ความต้านทานของสเตเตอร์ในแต่ละเฟส, L คือ ค่าความเหนี่ยวนำตัวเองของสเตเตอร์ในแต่ละเฟส และ M คือ ค่าความเหนี่ยวนำรวมแรงเคลื่อนไฟฟ้าย้อนกลับ (Back EMF) ที่เหนี่ยวนำขึ้นขึ้นขึ้นอยู่กับตำแหน่งของโรเตอร์ และสามารถแสดงได้ดังนี้[6]

$$e_a = \frac{k_e}{2} \omega_m F(\theta_e) \quad (10)$$

$$e_b = \frac{k_e}{2} \omega_m F(\theta_e - \frac{2\pi}{3}) \quad (11)$$

$$e_c = \frac{k_e}{2} \omega_m F(\theta_e - \frac{4\pi}{3}) \quad (12)$$

โดยที่ k_e คือ ความเร็วของโรเตอร์, ω_m คือ ค่าคงที่ของแรงเคลื่อนไฟฟ้าย้อนกลับ (Back-EMF Constant) และ θ_e คือ มุมทางไฟฟ้าของโรเตอร์ มุมทางไฟฟ้าของโรเตอร์เท่ากับมุมทางกลของโรเตอร์ คูณด้วยจำนวนขั้วแม่เหล็ก

$$\theta_e = \frac{p}{2} \theta_m \quad (13)$$

ฟังก์ชัน $F(\theta_e)$ แสดงรูปคลื่นของแรงเคลื่อนไฟฟ้าย้อนกลับ (Back-EMF) ที่มีลักษณะเป็นรูปสี่เหลี่ยมคางหมูหนึ่งคาบของฟังก์ชันนี้สามารถเขียนได้ดังนี้

$$F(\theta_e) = \begin{cases} 1 & 0 < \theta_e < \frac{2\pi}{3} \\ 1 - \frac{6}{\pi} (\theta_e - \frac{2\pi}{3}) & \frac{2\pi}{3} < \theta_e < \pi \\ -1 & \pi < \theta_e < \frac{5\pi}{3} \\ -1 + \frac{6}{\pi} (\theta_e - \frac{2\pi}{3}) & \frac{5\pi}{3} < \theta_e < 2\pi \end{cases} \quad (14)$$

แรงบิดที่พัฒนาได้ T_e สามารถแสดงได้ดังนี้

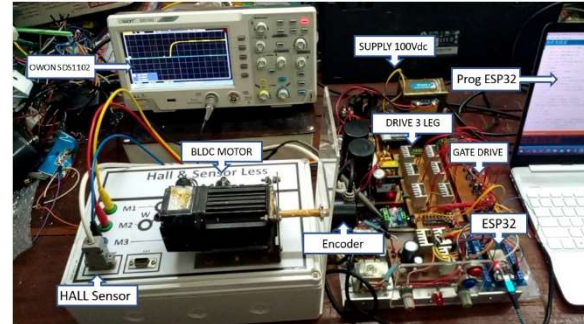
$$T_e = \frac{k_t}{2} \left[F(\theta_e) i_a + F(\theta_e - \frac{2\pi}{3}) i_b + F(\theta_e - \frac{4\pi}{3}) i_c \right] \quad (15)$$

โดยที่ k_t คือ ค่าคงที่ของแรงบิด (Torque Constant) สมการของส่วนทางกลสามารถแสดงได้ดังนี้

$$T_e - T_L = J \frac{d\omega_m}{dt} + \beta \omega_m \quad (16)$$

โดยที่ T_L คือ แรงบิดโหลด, J คือ โมเมนต์ความเฉื่อยของโรเตอร์ และ β คือ ค่าคงที่ของแรงเสียดทาน

ข้อมูลของมอเตอร์ BLDC พิจารณาฟังก์ชันถ่ายโอนเป็นระบบอันดับสองโดยเพิ่มวงจรขับที่แทนด้วยฟังก์ชันถ่ายโอนอันดับหนึ่งเข้าไป แสดงได้ดังสมการที่(17) ส่วนสมการที่(18),(19) แสดงแบบจำลองลำดับเศษส่วนโดยการดัดแปลงอัลกอริทึมคางคาว(MBA) ที่ความเร็ว 1800 รอบต่อนาที แสดงดัง สมการที่(20)[7] เป็นplant ของระบบที่นำมาใช้ในงานวิจัยนี้ได้จากการระบุเอกลักษณ์ของแบบจำลองมอเตอร์ BLDC แบบลำดับเศษส่วนมอเตอร์ BLDC ขนาด 100W,100 VDC,2A มีขั้วแม่เหล็ก 4 ขั้ว แสดงได้ในรูปที่ 4 [7]



รูปที่ 4 การทดลองในห้องปฏิบัติการจริง[7]

$$G_p(s) = \left(\frac{K_A}{\tau_A s + 1} \right) \left(\frac{K_t}{JL_a s^2 + (BL_a + JR_a) s + (BR_a + K_t K_b)} \right) \quad (17)$$

$$\left. \begin{aligned} a_0 &= K_A K_t \\ b_3 &= \tau_A J L_a \\ b_2 &= \tau_A B L_a + \tau_A J R_a + J L_a \\ b_1 &= \tau_A B R_a + \tau_A K_t K_b + B L_a + J R_a \\ b_0 &= B R_a + K_t K_b \end{aligned} \right\} \quad (18)$$

$$G_p(s) = \frac{a_0}{b_3 s^3 + b_2 s^2 + b_1 s + b_0} \quad (19)$$

$$G_p(s) = \frac{30.983212}{0.00010998s^{2.6072}} + 0.01994747s^{1.5098} + 1.00011994s^{0.9960} + 30.98231285 \quad (20)$$

$$G_p(s) = \frac{0.0338}{0.000000000808s^{2.6072}} + 0.0000018648s^{1.5098} + 0.0010955s^{0.9960} + 0.033788 \quad (21)$$

สมการที่ (17) – (20) เป็นสมการที่ได้จากงานวิจัยของผู้เขียน[7] ส่วนในสมการที่(21) ได้จากการนำสมการที่(20) มาทำการหาค่าพารามิเตอร์ของมอเตอร์ BLDC และพารามิเตอร์ของชุดขับด้วยเมตาฮิวริสติกด้วยการดัดแปลงอัลกอริทึมค้างคาวแสดงผลดังตารางที่1

ตารางที่1 แสดงค่าพารามิเตอร์ของ BLDC มอเตอร์และตัวขับ

Parameter	DRIVE & BLDC MOTOR
KA	0.1618
KT	0.2089
TA	0.0008
Kb	0.1363
J	0.0001
La	0.0101
Ra	10.6343
B	0.0005

ค่าในตารางนี้นำไปใช้ในการทดสอบในการควบคุมความเร็วมอเตอร์ BLDC และนำไปทดสอบความทนทานของระบบในงานวิจัยต่อไป

2.4 การดัดแปลงอัลกอริทึมค้างคาว

ประเด็นสำคัญประการหนึ่งของเมตาฮิวริสติกแบบประชากร คือการสร้างสมดุลระหว่างขั้นตอนการสำรวจและขั้นตอนการเพิ่มความเข้มข้นในกระบวนการค้นหาคำตอบ ในทางตรงกันข้าม การเพิ่มความเข้มข้นมีหน้าที่ในการค้นหาในระดับเฉพาะจุด (local search) ของอัลกอริทึมตามที่ได้กำหนด [8] อัลกอริทึมค้างคาวมีความสามารถในการเพิ่มความเข้มข้นในการหาคำตอบแต่ยังมีข้อบกพร่องบางประการในขั้นตอนการสำรวจ อัลกอริทึมค้างคาวยังประสบปัญหาขาดความสามารถในการจดจำคำตอบที่ดีที่สุดที่เคยพบระหว่างกระบวนการปรับให้เหมาะสม ซึ่งทำให้ค้างคาวมีแนวโน้มที่จะหลุดออกจากบริเวณที่มีคำตอบที่ดีในพื้นที่การค้นหา อัลกอริทึมค้างคาวยังมีแนวโน้มที่จะออกไปนอกขอบเขตของพื้นที่การ

ค้นหาคำตอบ Yang ใน [9] ได้เสนอให้ใช้ขีดจำกัดบนของเวกเตอร์ตำแหน่ง (upper bound limits on position vector) เพื่อแก้ไขข้อจำกัดเหล่านี้ อย่างไรก็ตาม ยังไม่มีวิธีการแก้ปัญหาอย่างเป็นระบบสำหรับปัญหาที่กล่าวถึงก่อนหน้านี้ ดังนั้นจึงได้มีการเสนอการปรับปรุงในงานวิจัยนี้โดยการปรับปรุงอัลกอริทึมค้างคาวได้ถูกนำเสนอในสองส่วน ได้แก่การปรับเปลี่ยนรูปแบบการยอมรับคำตอบใหม่ และการปรับสมการความเร็ว (velocity equation) เพื่อแก้ไขข้อจำกัดที่ได้กล่าวถึงก่อนหน้านี้ การปรับปรุงที่นำเสนอได้ถูกรวบรวมไว้ในรูปแบบรหัสเทียม (pseudo code) ซึ่งแสดงไว้ในรูปที่ 5 นอกจากนี้ยังได้มีการเพิ่มหน่วยความจำสำหรับเก็บคำตอบที่ดีที่สุดที่พบระหว่างกระบวนการปรับให้เหมาะสมโดยอัลกอริทึม

1:	Randomly initialize position x_i and velocity v_i of i -th bat in population
2:	Initialize pulsation frequency $Q_i \in [Q_{min}, Q_{max}]$, pulsation r_i and loudness A_i of i -th bat in population
3:	while not termination conditions are satisfied:
4:	for each bat in population:
5:	$v_i(t) = \alpha_i v_i(t-1) + Q_i(x^* - x_i(t-1)) + Q_i(x_{ever}^* - x_i(t-1))$
6:	$x_i(t) = x_i(t-1) + v_i(t)$
7:	if $\text{rand}(0,1) > r_i^t$: $x_i' \leftarrow$ generate new solution around current bat x_i
8:	if $f(x_i') < f(x_i)$ or $\text{rand}(0,1) < A_i^t$: Update values of pulsation and loudness, respectively r_i^t and A_i^t as: $A_i^{t+1} \leftarrow \alpha A_i^t$; $r_i^{t+1} \leftarrow r_i^t(1 - \exp(-\gamma t))$
9:	Evaluate bats population using fit function f
10:	Find best bat in population and mark him as x^*
11:	if $f(x^*) < f(x_{ever}^*)$: $x_{ever}^* \leftarrow x^*$

รูปที่ 5 การปรับเปลี่ยนอัลกอริทึมค้างคาว[10]

แนวทางคือทำการปรับสมการตำแหน่งและความเร็วของค้างคาวเมื่อเปรียบเทียบกับสมการที่นำเสนอใน [9] ผู้วิจัยได้ใช้ส่วนประกอบของคลังคำตอบ (archive component) เพื่อช่วยนำทางค้างคาวไปยังบริเวณที่เคยพบคำตอบที่ดี และยังได้นำแนวคิดของสัมประสิทธิ์การรับรู้ (cognition coefficients) มาใช้แทนการใช้ขีดจำกัดบนของตำแหน่งสมการ (22) และ (23) แสดงการปรับปรุงที่เสนอไว้ดังนี้

$$v_i(t) = \alpha_i v_i(t-1) + Q_i(x^* - x_i(t-1)) + Q_i(x_{ever}^* - x_i(t-1)) \quad (22)$$

$$v_i(t) = \alpha_i v_i(t-1) + Q_i(x^* - x_i(t-1)) + Q_i(x_{ever}^* - x_i(t-1)) \quad (23)$$

α_i = สัมประสิทธิ์การรับรู้ของค้างคาวตัวที่ i

$x^* - x_i(t-1)$ = องค์ประกอบทางสังคมของค้างคาว

$x_{ever}^* - x_i(t-1)$ = ส่วนประกอบหน่วยความจำเก็บข้อมูลถาวร

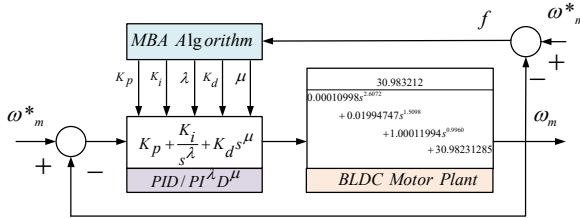
Q_i = สัญญาณความถี่ของค้างคาวตัวที่ i

การปรับปรุงยังรวมถึงการ เพิ่มรูปแบบใหม่ของหน่วยความจำที่เรียกว่า x_{ever}^* ซึ่งใช้เก็บคำตอบที่ดีที่สุดที่เคยพบมาตลอดกระบวนการค้นหา สิ่งที่สำคัญคือ การปรับปรุงที่ได้นำเสนอไม่ได้เปลี่ยนความซับซ้อน

เชิงคำนวณของอัลกอริทึม เนื่องจากการปรับปรุงเหล่านี้มีลักษณะเป็นเชิงเส้น (linear) และไม่ได้ต้องการคำนวณเพิ่มเติมหรือการประเมินฟังก์ชันความเหมาะสม (fitness function) เพิ่มเติมแต่อย่างใด

2.5 การออกแบบระบบควบคุมความเร็วลำดับเศษส่วน

ตัวควบคุม $PI^\lambda D^\mu$ ลำดับเศษส่วนได้นำการค้นหาค่าแบบอัลกอริทึม ค้างคาวแบบปรับปรุง มาใช้ในการออกแบบระบบควบคุมความเร็วรอบสำหรับมอเตอร์ BLDC โครงสร้างการค้นหาค่าแบบอัลกอริทึมค้างคาวแบบปรับปรุงในการออกแบบระบบควบคุมความเร็วรอบมอเตอร์ BLDC แสดงอยู่ในรูปที่ 6 อัลกอริทึมอัลกอริทึมค้างคาวแบบปรับปรุงถูกนำมาใช้เพื่อกำหนดค่า $K_p, K_i, K_d, \lambda, \mu$ ที่เหมาะสมที่สุดสำหรับพารามิเตอร์ โดยกำหนดขอบเขตการค้นหาดังสมการที่ (6) [4]



รูปที่ 6 โครงสร้างของระบบควบคุม $PI^\lambda D^\mu$ ที่ถูกออกแบบด้วย MBA

2.6 ความทนทาน

เพื่อทดสอบความทนทานของวิธีการที่นำเสนอได้มีการศึกษาผลของการเปลี่ยนแปลงฉับพลันสำหรับแรงบิดเมื่อใส่โหลด 1 หน่วย ต่อประสิทธิภาพของการควบคุมแรงบิดและการเปลี่ยนแปลงความเร็วจากความเร็ว 1800 รอบต่อนาที และ 800 รอบต่อนาที และการเปลี่ยนแปลงความต้านทาน R_a และความเฉื่อย J ต่อความเร็วแสดงดังตารางที่ 2

ตารางที่ 2 แสดงค่า $R_a = R_a + 20\%R_a$ และ $J = J + 20\%J$

Parameter	R, J normal	$R = R + 20\%R, J = J + 20\%J$
KA	0.1618	0.1618
KT	0.2089	0.2089
TA	0.0008	0.0008
Kb	0.1363	0.1363
J	0.0001	0.00012
La	0.0101	0.0101
Ra	10.6343	12.7612
B	0.0005	0.0005

$$G_p(s) = \frac{0.0338}{0.000000009696s^{2.6072} + 0.000024411s^{1.5098} + 0.0015643s^{0.9960} + 0.034854} \quad (24)$$

สมการที่(24) ได้จากตารางที่ 2 โดยปรับความต้านทาน $R_a = R_a + 20\%R_a$ และปรับค่า $J = J + 20\%J$ ในสมการนี้จะถูกนำไปทดสอบระบบเพื่อนำไปทดสอบความคงทนแสดงได้ดังรูปที่ 12-13

3. ผลการทดสอบและการอภิปรายผล

ในการวิจัยนี้ ตัวควบคุม PID และ $PI^\lambda D^\mu$ ถูกนำมาเปรียบเทียบกับกันโดยใช้เทคนิคการค้นหาค่าที่เหมาะสมที่สุดแบบ MBA ค่าพารามิเตอร์เริ่มต้นการค้นหานี้ จำนวนตัวแปรที่ต้องการค้นหาเป็น $d = 5$ จำนวนของค้างคาว $n = 20$ ความถี่ $Q_{min} = 0$ และ $Q_{max} = 2$ ความดังของเสียง $A = 0.5$ อัตราการปล่อยพัลส์ = 0.5 และจำนวนรอบสูงสุด (Max_Gen) = 400 กำหนดให้เป็นเกณฑ์ยุติการค้นหา(TC) อยู่ภายใต้ขอบเขตสูงสุดและต่ำสุดดังสมการที่ (6) ผลที่ได้แสดงในตารางที่ 3

ตารางที่ 3 ค่าพารามิเตอร์ของตัวควบคุม PID/ $PI^\lambda D^\mu$

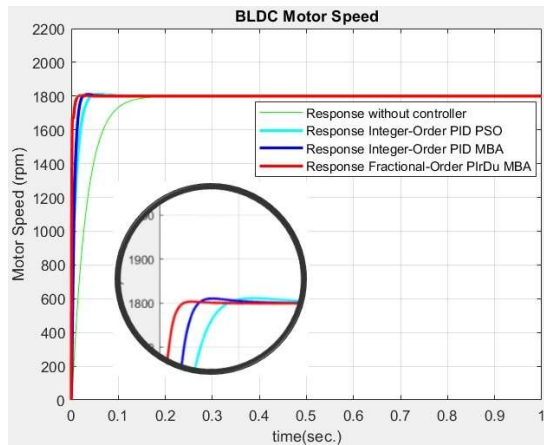
Parameter	PID (PSO)	PID (MBA)	$PI^\lambda D^\mu$ (MBA)
K_p	4.1935	5.0780	25.8571
K_i	136.7900	170.6344	782.5988
λ	1	1	0.9979
K_d	0.0163	0.0038	0.0503
μ	1	1	1.0525

ค่าพารามิเตอร์ในตารางที่ 3 สำหรับระบบควบคุมความเร็ว มอเตอร์ BLDC ขึ้นอยู่ภายใต้การป้อนอินพุตสัญญาณแบบขั้นบันได นอกจากนี้ผู้วิจัยได้มีการหาค่าพารามิเตอร์ของตัวควบคุมเพื่อเปรียบเทียบประสิทธิภาพของ MBA กับเมตาฮิวริสติก PSO และ $PI^\lambda D^\mu$ เพื่อยืนยันข้อดีที่เหนือกว่า

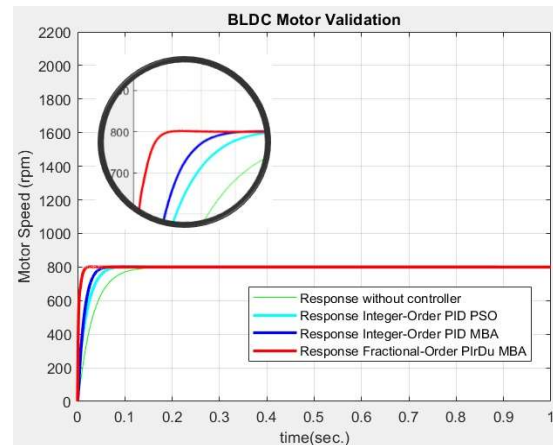
ตารางที่ 4 ค่าประสิทธิภาพของตัวควบคุม PID/ $PI^\lambda D^\mu$

Controller	Performance Parameter						
	t_r	t_p	t_s	t_{reg}	MO%	Obj(f)	t_{Seach}
PID (PSO)	0.0431	0.0578	0.1458	0.2378	0.6399	1.7628	1223.55
PID (MBA)	0.0249	0.0328	0.0765	0.2528	0.6043	1.5990	724.100
$PI^\lambda D^\mu$ (MBA)	0.0151	0.0216	0.0348	0.1191	0.1991	1.1288	756.801

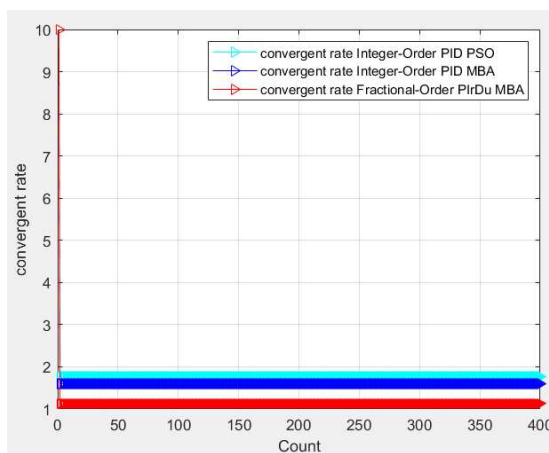
ผลที่ได้ในตารางที่ 4 พบว่าตัวควบคุมแบบ $PI^\lambda D^\mu$ มีความเหมาะสมกว่าเมื่อเทียบกับตัวควบคุม PID(PSO) และตัวควบคุม PID(MBA) ผลตอบสนองแสดงได้ดังรูปที่ 7 ส่วนรูปที่ 8 แสดงเส้นโค้งลู่เข้าปัจจุบันของคำตอบสำหรับตัวควบคุม PID/ $PI^\lambda D^\mu$ โดยกำหนดพารามิเตอร์เริ่มต้นของการค้นหาในสมการที่(6) รูปที่ 9 แสดงผลการตอบสนองของระบบเมื่อเพิ่มโหลด 1 หน่วย และรูปที่ 11 แสดงผลตอบสนองของสัญญาณควบคุมขณะ No load และ On load ขนาด 1 หน่วย ที่เวลา 0.25 วินาที รูป 12-13 แสดงความคงทนของระบบเมื่อ $R = R + 20\%R$ และ $J = J + 20\%J$ ตามลำดับ



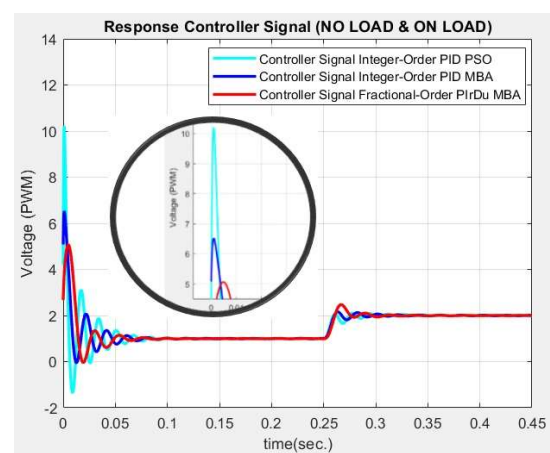
รูปที่ 7 ผลตอบสนองของระบบควบคุม $PID/PI^\lambda D^\mu$ ความเร็วรอบ 1800 rpm



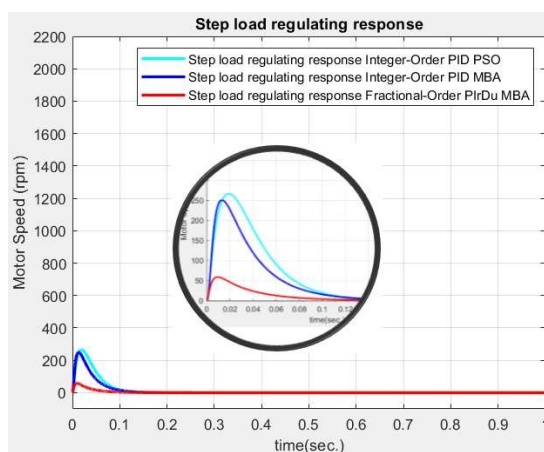
รูปที่ 10 ผลตอบสนองของ $PID/PI^\lambda D^\mu$ เมื่อทดสอบที่ 800 rpm



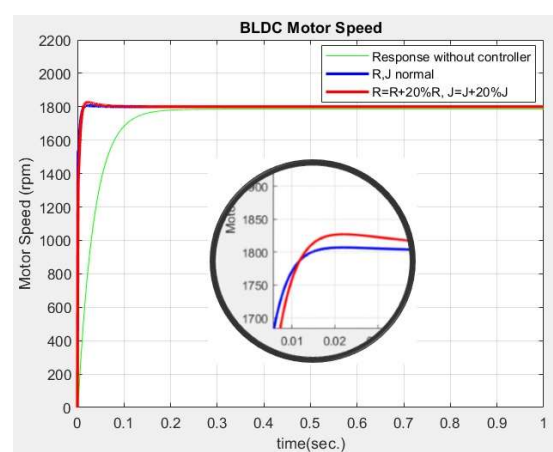
รูปที่ 8 การลู่อเข้าหาผลเฉลยของตัวควบคุม $PID/PI^\lambda D^\mu$



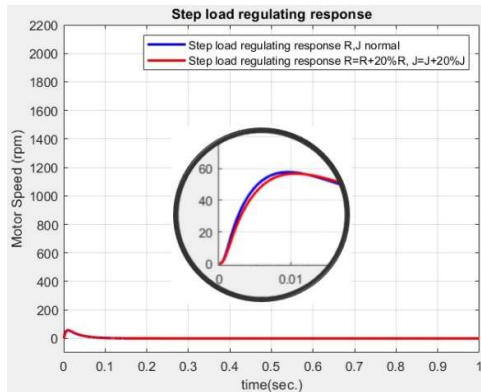
รูปที่ 11 ผลตอบสนองของสัญญาณควบคุมเมื่อเพิ่มโหลด 1 หน่วย



รูปที่ 9 ผลตอบสนองของตัวควบคุมเมื่อเพิ่มโหลด 1 หน่วย



รูปที่ 12 ตัวควบคุม $PI^\lambda D^\mu$ เมื่อ $R=R+20\%R$, $J=J+20\%J$



รูปที่ 13 ตัวควบคุม $PI^\lambda D^\mu$ เมื่อเพิ่มโหลด 1 หน่วย
เมื่อ $R=R+20\%R$, $J=J+20\%J$

4. สรุป

ตัวควบคุม $PI^\lambda D^\mu$ ลำดับเศษส่วนอย่างเหมาะสมสำหรับควบคุมความเร็วมอเตอร์ BLDC ที่นำเสนอในบทความนี้ ผลจากการตรวจสอบได้เปรียบเทียบกับสมรรถนะของระบบควบคุมที่ใช้ตัวควบคุม $PI^\lambda D^\mu$ ลำดับเศษส่วนกับตัวควบคุม PID ลำดับจำนวนเต็ม พบว่าตัวควบคุม $PI^\lambda D^\mu$ มีสมรรถนะดีกว่าตัวควบคุม PID ลำดับจำนวนเต็ม พิจารณาจากผลที่ได้ในตารางที่ 3 ค่าพารามิเตอร์ของตัวควบคุม $K_p, K_i, K_d, \lambda, \mu$ ที่ได้พบว่ามีค่าแตกต่างกันเนื่องจากฟังก์ชันวัตถุประสงค์ของ $PI^\lambda D^\mu$ มีค่าน้อยกว่า PID จึงทำให้ค่าพารามิเตอร์ของตัวควบคุมถูกต้องมากกว่าจึงให้ผลตอบสนองของระบบดีกว่าแสดงได้ดัง ตารางที่ 4 แสดงได้ดังรูปที่ 7 ผลตอบสนองของระบบควบคุม จึงทำให้ การตอบสนองของระบบควบคุมเมื่อเพิ่มโหลด 1 หน่วย ดังรูปที่ 9 และเมื่อพิจารณาผลตอบสนองของสัญญาณควบคุมพบว่า $PI^\lambda D^\mu$ จะตอบสนองสัญญาณควบคุมขณะ On load และ No load ได้เร็วกว่า PID แสดงดัง รูปที่ 11 ส่วนรูปที่ 12-13 แสดงถึงความคงทนของระบบถึงแม้ว่าค่า R , และ J มีค่าเพิ่มขึ้นระบบยังคงมีเสถียรภาพดังผลแสดงในงานวิจัยนี้

เอกสารอ้างอิง

- [1] Viond KvSingh patel, A. K. Pandey, " Modeling and Simulation of Brushless Dc motor using Pwm control Technique" International Journal of Engineering Research and Applications, Vol.3, Issue 3, May-Jun 2013, PP.612-620.
- [2] Padmaraja Yedamale, " Brushless DC (BLDC) Motor Fundamentals", AN885, 2003 Microchip Technology Inc.
- [3] José Carlos Gamazo - Real*, Ernesto Vázquez - Sánchez and Jaime Gómez - Gil, " Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends " Department of Signal Theory, Communications and Telematic

Engineering, University of Valladolid, Sensors, PP.6901-6947, June-July 2010.

- [4] Ammar A. Aldair and Weiji J. Wang , " Design of Fractional Order Controller Based on Evolutionary Algorithm for a Full Vehicle Nonlinear Active Suspension Systems ", International Journal of Control and Automation, Vol. 3, No.4, December 2010.
- [5] Ankit Rastogi, Pratibha Tiwari, " Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control Using Particle Swarm Optimization ", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume 3, Issue 2, May 201.
- [6] Stefan Baldursson, " BLDC motor modeling and control- a MATLAB/ SIMULINK implementation ", Master Thesis in Electrical Power Engineering, Chalmers University of Technology, May 2005, Gothenburg, Sweden.
- [7] C. Kiree, P. Somranmark, S. Karejinda, " Fractional-Order Model Identification of BLDC Motor by Modified Bat Algorithm" The 47th Electrical Engineering Conference (EECON-47), Udon Thani, Thailand, 2024, pp. 380-385.
- [8] G. Wang and L. Guo, " A Novel Hybrid Bat Algorithm with Harmony Search for Global Numerical Optimization", Journal of Applied Mathematics, vol. 2013, pp. 21, 2013
- [9] X. S. Yang, " A new metaheuristic bat-inspired algorithm", Nature Inspired Cooperative Strategies for Optimization, 2010.
- [10] A. Baziar, A. A. Kavousi-Fard, J. Zare, " A novel Self Adoptive Modification Approach Based on Bat Algorithm for Optimal Management of Renewable MG", Journal of Intelligent Learning System and Application, 2013.
- [11] "การสังเคราะห์ระบบควบคุม" ของ รองศาสตราจารย์ เดชา พวงดาวเรือง ISBN978-616-7659-35-0 หน้า 348-352



อาจารย์ ดร.ชูเกียรติ ตีรี

งานวิจัยที่สนใจ

Machine Design, Power Electronic, Embedded System and IoT, Control System, Search Optimization