

# LQR Controller Based on BAT Algorithm for Rotary Double Parallel Inverted Pendulum

Thanh-Tri-Dai Le<sup>1</sup>, Ngoc-Kien Nguyen<sup>2</sup>, Phuc-Truong Le<sup>3</sup>, Minh-Nguyen-Bao Bui<sup>4</sup>, Trong-Tin Nguyen<sup>5</sup>, Chi-Anh Tran<sup>6</sup>,  
Phuong-Tu Doan<sup>7</sup>, Duc-Nhan Dao<sup>8</sup>, Van-Dong-Hai Nguyen<sup>9</sup>, Thanh-Tung Nguyen<sup>10,\*</sup>  
<sup>1, 2, 3, 4, 5, 6, 7, 8, 9, 10</sup> Ho Chi Minh City University of Technology and Education (HCMUTE),  
Ho Chi Minh City (HCMC), Vietnam

<sup>10</sup> Ho Chi Minh City Bach Nghe College, Ho Chi Minh City (HCMC), Vietnam

Email: <sup>1</sup> 19151051@student.hcmute.edu.vn, <sup>2</sup> 21146477@student.hcmute.edu.vn, <sup>3</sup> 21146528@student.hcmute.edu.vn,  
<sup>4</sup> 20151013@student.hcmute.edu.vn, <sup>5</sup> 21151174@student.hcmute.edu.vn, <sup>6</sup> 20151334@student.hcmute.edu.vn,  
<sup>7</sup> 21146531@student.hcmute.edu.vn, <sup>8</sup> 21145224@student.hcmute.edu.vn, <sup>9</sup> hainvd@hcmute.edu.vn,  
<sup>10</sup> tungnguyen98ac@gmail.com

\*Corresponding Author

**Abstract**—This paper presents an enhanced approach to stabilizing the Rotary Double Parallel Inverted Pendulum (RDPIP) through a combination of the LQR method and the BAT algorithm. Traditionally, selecting appropriate  $Q$  and  $R$  matrices relies on designers' intuitions or trial-and-error processes, often resulting in suboptimal performance. By leveraging the BAT algorithm's swarm intelligence, the proposed method automatically optimizes the cost function to yield improved control performance. Key improvements include shorter stabilization time, reduced overshoot, and minimized oscillations. Simulation results show that the BAT-enhanced LQR controller significantly outperforms traditional design in terms of convergence speed and system damping. These findings underscore the potential of metaheuristic algorithms in refining classical control strategies for complex, nonlinear systems.

**Keywords**—LQR Controller; BAT Algorithm; Rotary Double Parallel Inverted Pendulum; Swarm Intelligence

## I. INTRODUCTION

Inverted pendulum (IP)—characterized by its under-actuated, nonlinear, and strongly coupled nature—is a classic benchmark in control theory. It effectively exemplifies key attributes such as stability, robustness, and tracking performance, making it an ideal platform for comparing different control strategies and assessing their respective strengths and weaknesses. Consequently, extensive research on this system has significantly advanced the field of control theory [1]-[3].

Fig. 1 illustrates fundamental structure of a novel benchmark model: the first-order RPDIP. With its intricate mechanical design, higher degrees of freedom, and pronounced nonlinear dynamics, RPDIP poses a considerably greater challenge in terms of control. Studies on RPDIP are expected to provide valuable insights that will guide the development of control methods for similar systems. LQR and nonlinear controllers have been applied to Rotary IP (RIP) [4][5] and other kinds of pendulums [6]. PRDIP is a developed form from RIP in mechanical design by adding parallel link to existing arm [7]. Based on that research, a fuzzy controller is designed to operate this model well from imitating LQR traditional controller [8]. Thence, it is proved that intelligent control can be applied for this model from a

linear controller. However, a direction of optimizing LQR control has not been mentioned yet for this model. In this research, we utilize results from previously tested LQR control [7] but we develop a searching method to optimize the traditional LQR control.

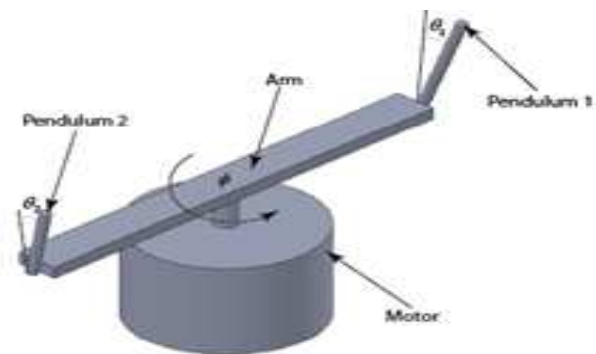


Fig. 1. PRDIP model [7]

LQR is a classical control that is applied widely in IP [8]-[12]. This method is designed by modeling the system structure. Its accuracy and efficiency have been verified. But, weight matrices  $Q$  and  $R$  are still chosen through the trial-and-error test. In [13]-[17], methods of choosing those matrices are shown, but difficulties in simultaneously ensuring satisfaction of the objective function and satisfying the criteria still exist. In this paper, we use the BAT algorithm for multivariate problems. This algorithm learns from the movement of bats in finding prey and avoiding obstacles in order to find solutions for the objective function [18]. LQR control is designed from on swarm optimization. The BAT algorithm is used to optimize  $Q$  and  $R$ . By updating matrix parameter values, we use weighting techniques to incorporate designers' experiences in algorithm's search. Simulation proves the ability of BAT algorithm. Optimized control law is shown to be effective when compared with conventional LQR control rules.

Mathematical model of system forms the foundation for designing the LQR controller, which is enhanced through the application of swarm optimization techniques. Specifically, the BAT algorithm is employed to optimize the  $Q$  and  $R$

matrices, leveraging its robust search capabilities. To further refine optimization process, a weighting technique is integrated into parameter update mechanism, allowing the incorporation of designer's expertise into the algorithm's search process. This approach aims to achieve a globally optimal solution for the matrices, enabling LQR controller to generate an optimal state feedback control matrix. By doing so, it addresses the limitations associated with traditional methods that rely heavily on experience, trial-and-error for selecting  $Q$  and  $R$  matrices. Effectiveness of the proposed controller is demonstrated through simulation, where its performance is compared against conventional LQR control.

Novelty of BAT algorithm in comparison to other-based LQR designs, such as PSO or genetic algorithm (GA), lies in its bio-inspired echolocation mechanism, which enables a dynamic balance between exploration and exploitation through adaptive adjustments of frequency, loudness, and pulse emission rates. Unlike PSO, which relies on position and velocity updates influenced by social learning, or GA, which depends on stochastic crossover and mutation operators, the BAT algorithm offers a more structured and directed search process. It facilitates efficient local exploitation near the best-known solutions while preserving global search capabilities, thereby improving the likelihood of escaping local optimization in high-dimensional and nonlinear optimization problems. The BAT algorithm requires fewer control parameters, simplifies tuning efforts, and generally exhibits faster convergence. These features make it particularly well-suited for optimizing the  $Q$  and  $R$  weighting matrices, especially for complex and under-actuated systems such as RDPIP.

## II. MATHEMATICAL MODEL OF PRDIP

### A. The fundamentals of PRDIP

System parameters are listed in Table 1 [7], Table 2 presents key parameters relevant to PRDIP [7].

Table 1. System parameters

Parameters	Definition	Unit
$m_i$	First pendulum's mass	kg
$l_{hi}$	First pendulum's length	m
$\theta_i$	Angle of $i^{\text{th}}$ pendulum	rad
$J_i$	The inertia of " $i$ -th" pendulum	kgm <sup>2</sup>
$J_i = J_i \sin^2 \theta_i$	Effective moment of inertia of pendulum	kgm <sup>2</sup>
$\phi$	Angle of arm	rad
$L$	Length of arm	m
$J_0$	Arm's inertia	kgm <sup>2</sup>
$\tau$	Torque of DC motor	Nm
$g$	Gravitational constant	m/s <sup>2</sup>
$C_i$	Coefficient of viscosity for $i$ -th pendulum	Nms
$C_0$	Coefficient of viscosity of arm	Nms

Table 2. Values of system parameters

Parameter	Pendulum 1	Pendulum 2	Arm
$m_i$	0.059	0.038	Na
$l_{hi}$	0.0127	0.082	Na
$L$	Na	Na	0.51
$J_i$	0.0001526	0.082	Na
$J_0$	Na	Na	0.75
$C_i$	$1.526 \times 10^{-4}$	$4.0693 \times 10^{-4}$	Na
$C_0$	Na	Na	4.978

Lagrange operator is

$$L = V - W \quad (1)$$

The following is the Lagrange equation.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial W}{\partial \dot{q}_i} = F_i \quad (2)$$

Kinetic energy is

$$K = \frac{1}{2} J_0 \dot{\phi}^2 + \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} J_2 \dot{\theta}_2^2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (3)$$

where  $v_1, v_2$ , respectively, are the velocities of the first and second pendulums.

Kinetic energy is rewritten as.

$$\begin{aligned} K = & \frac{1}{2} J_0 \dot{\phi}^2 + \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} J_2 \dot{\theta}_2^2 \\ & + \frac{1}{2} m_1 (l_{h1} \sin \theta_1 \dot{\phi})^2 + \frac{1}{2} m_1 (L \dot{\phi})^2 \\ & + \frac{1}{2} m_1 (l_{h1} \dot{\theta}_1)^2 - m_1 l_{h1} L \cos \theta_1 \dot{\theta}_1 \dot{\phi} \\ & + \frac{1}{2} m_2 (l_{h2} \sin \theta_2 \dot{\phi})^2 + \frac{1}{2} m_2 (L \dot{\phi})^2 \\ & + \frac{1}{2} m_2 (l_{h2} \dot{\theta}_2)^2 - m_2 l_{h2} L \cos \theta_2 \dot{\theta}_2 \dot{\phi} \end{aligned} \quad (4)$$

The system's potential energy is

$$U = m_1 g l_{h1} \cos \theta_1 + m_2 g l_{h2} \cos \theta_2 \quad (5)$$

Energy dissipation of PRDIP is

$$W = \frac{1}{2} c_0 \dot{\phi}^2 + \frac{1}{2} c_1 \dot{\theta}_1^2 + \frac{1}{2} c_2 \dot{\theta}_2^2 \quad (6)$$

The Lagrange equation is.

$$\begin{aligned} L = V - W = & \frac{1}{2} J_0 \dot{\phi}^2 + \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} J_2 \dot{\theta}_2^2 \\ & + \frac{1}{2} m_1 (l_{h1} \sin \theta_1 \dot{\phi})^2 + \frac{1}{2} m_1 (L \dot{\phi})^2 \\ & + \frac{1}{2} m_1 (l_{h1} \dot{\theta}_1)^2 - m_1 l_{h1} L \cos \theta_1 \dot{\theta}_1 \dot{\phi} \\ & + \frac{1}{2} m_2 (l_{h2} \sin \theta_2 \dot{\phi})^2 + \frac{1}{2} m_2 (L \dot{\phi})^2 \\ & + \frac{1}{2} m_2 (l_{h2} \dot{\theta}_2)^2 - m_2 l_{h2} L \cos \theta_2 \dot{\theta}_2 \dot{\phi} \\ & - m_1 g l_{h1} \cos \theta_1 - m_2 g l_{h2} \cos \theta_2 \end{aligned} \quad (7)$$

DC servo motor is used to control whole model. Relation between torque and voltage is

$$\tau = \frac{K_t}{R_a} V_I - \frac{K_t K_b}{R_a} \dot{\phi} \quad (8)$$

Parameters of DC motor are in Table 3 [7].

Table 3. Parameters of motor

Parameter	Unit	Value
$K_t$	Vs/rad	0.064944
$K_b$	Vs/rad	0.064944
$R_a$	$\Omega$	6.835271

Dynamic equations of PRDIP are

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} = \frac{K_t}{R_m} \begin{bmatrix} V_f \\ 0 \\ 0 \end{bmatrix} \quad (9)$$

Where:

$$Z_{11} = J_0 + m_1 l_{h1}^2 \sin^2 \theta_1 + m_1 L^2 + m_2 l_{h2}^2 \sin^2 \theta_2 + m_2 L^2 \quad (10)$$

$$Z_{12} = -m_1 L l_{h1} \cos \theta_1 \quad (11)$$

$$Z_{13} = -m_2 L l_{h2} \cos \theta_2 \quad (12)$$

$$K_1 = m_1 l_{h1}^2 \dot{\theta}_1 \dot{\phi} \sin(2\theta_1) + m_1 l_{h1} L \dot{\theta}_1^2 \sin \theta_1 + c_0 \dot{\phi} + m_2 l_{h2}^2 \dot{\theta}_2 \dot{\phi} \sin(2\theta_2) + (c_0 + \frac{K_t K_b}{R_a}) \dot{\phi} \quad (13)$$

$$Z_{21} = -m_1 L l_{h1} \cos \theta_1 \quad (14)$$

$$Z_{22} = J_1 + m_1 l_{h1}^2 \quad (15)$$

$$Z_{23} = 0 \quad (16)$$

$$K_2 = -m_1 l_{h1}^2 \dot{\phi}^2 \sin \theta_1 \cos \theta_1 - m_1 g l_{h1} \sin \theta_1 + c_1 \dot{\theta}_1 \quad (17)$$

$$Z_{31} = -m_2 L l_{h2} \cos \theta_2 \quad (18)$$

$$Z_{32} = 0 \quad (19)$$

$$Z_{33} = J_2 + m_2 l_{h2}^2 \quad (20)$$

$$K_3 = -m_2 l_{h2}^2 \dot{\phi}^2 \sin \theta_2 \cos \theta_2 - m_2 g l_{h2} \sin \theta_2 + c_2 \dot{\theta}_2 \quad (21)$$

### B. Linearized state equation

Equilibrium point (the upright position) is chosen as

$$\phi \approx 0; \theta_1 \approx 0; \theta_2 \approx \pi; \dot{\phi} \approx 0; \dot{\theta}_1 \approx 0; \dot{\theta}_2 \approx 0 \quad (22)$$

$$x = [\phi \quad \theta_1 \quad \theta_2 \quad \dot{\phi} \quad \dot{\theta}_1 \quad \dot{\theta}_2]^T \quad (23)$$

The linear model at the equilibrium can be obtained by

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (24)$$

Matrices A and B are calculated to be

$$A = N^{-1}S; B = N^{-1}J \quad (25)$$

Where:

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_0 + m_1 L^2 + m_2 L^2 & -m_1 L l_{h1} & -m_2 L l_{h2} \\ 0 & 0 & 0 & -m_1 L l_{h1} & J_1 + m_1 l_{h1}^2 & 0 \\ 0 & 0 & 0 & -m_2 L l_{h2} & 0 & J_2 + m_2 l_{h2}^2 \end{bmatrix} \quad (26)$$

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -(c_0 + \frac{K_t K_b}{R_a}) & 0 & 0 \\ 0 & m_1 g l_{h1} & 0 & 0 & -c_1 & 0 \\ 0 & 0 & m_2 g l_{h2} & 0 & 0 & -c_2 \end{bmatrix} \quad (27)$$

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{K_t}{R_a} & 0 \end{bmatrix}^T \quad (28)$$

Thence, it yields

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0.1008 & 0.1037 & -1.8760 & -0.0002 & -0.001 \\ 0 & 20.4029 & 0.1095 & -3.9575 & -0.0424 & -0.001 \\ 0 & 0.1360 & 26.0858 & -3.6667 & 0.0003 & -0.2478 \end{bmatrix} \quad (26)$$

$$B = [0 \quad 0 \quad 0 \quad 0.0123 \quad 0.0130 \quad 0.0167]^T \quad (27)$$

If using pole-placement, MATLAB can be obtained by using the command eig (A):

$$\begin{aligned} \lambda_1 = 0; \lambda_2 = 4.4922; \lambda_3 = 4.9816; \\ \lambda_4 = -4.5458; \lambda_5 = -5.2414; \lambda_6 = -1.8528 \end{aligned} \quad (28)$$

## III. SYNTHESIS CONTROLLER FOR PRDIP

### A. LQR Method

Nonlinear system is shown as

$$\dot{x} = f(x) + g(x)u \quad (29)$$

where  $x = [x_1 \quad x_2 \quad \dots \quad x_n]^T$  is state variable matrix; u is control signal.

Equilibrium point is chosen as:

$$x_1 = x_2 = \dots = 0; x_3 = -\pi(x_0 = x) \quad (30)$$

When u=0, system is balanced. We can approximate system in (29) to linear form as

$$\dot{x} = Ax + Bu \quad (31)$$

where

$$A = \left. \frac{\delta f}{\delta x} \right|_{\substack{x=x_0 \\ u=0}}; B = \left. \frac{\delta g}{\delta x} \right|_{\substack{x=x_0 \\ u=0}}$$

Around equilibrium point, we consider system approximately a linear system. Thence, LQR control structure at working point of a linear system in (31) can be shown in Fig. 2.

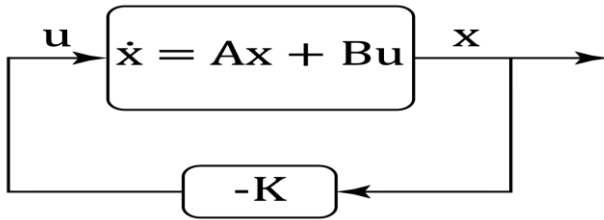


Fig. 2. LQR controller structure

Control signal is

$$u(t) = -Kx(t) \quad (32)$$

Command to compute K is executed as follows:

$$K_{LQR} = lqr(A, B, Q_{LQR}, R_{LQR}) \quad (33)$$

Where: A and B are calculated in (31), Q and R are weight matrices selected as follows.

$$Q_{LQR} = \begin{bmatrix} Q_0 & 0 & 0 & 0 \\ 0 & Q_1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Q_5 \end{bmatrix}; R_{LQR} = R_0 \quad (34)$$

Matrices Q and R impose mutual restrictions, where Q is directly proportional to the system's anti-interference capability. Increasing Q enhances this capability and shortens the system's adjustment time. However, it also amplifies system oscillations and raises energy consumption. Conversely, increasing R reduces power consumption but extends the adjustment time. Therefore, the key to effective design lies in determining the appropriate weight matrices Q and R. Once these matrices are established, the state feedback matrix K is determined. However, selecting Q and R largely relies on experience and a trial-and-error approach in the LQR controller design process. This subjectivity can lead to an imperfect controller design, ultimately affecting control efficiency.

$$K_{LQR} = [K_0 \quad K_1 \quad K_2 \quad K_3 \quad K_4 \quad K_5] \quad (35)$$

### B. Basics of BAT algorithm

BAT algorithm [19], is inspired by the echolocation behavior of microbats. A key distinguishing feature of this algorithm is its use of frequency tuning, making it the first of

its kind to integrate optimization with computational intelligence. In this approach, each bat is represented by a velocity  $v_i^t$  and a position  $x_i^t$  at iteration  $t$  within a  $d$ -dimensional search space. The position serves as a solution vector associated with a specific objective function. Among the  $n$  bats in the population, the best solution  $x^*$  found during the iterative search is retained for reference. The algorithm follows these fundamental assumptions [17]-[23]:

Bats utilize sound wave echolocation to assess distances and can distinguish between food sources, prey, and obstacles.

Each bat moves randomly with velocity  $v_i^t$  at position  $x_i^t$ , adjusting frequency  $Q$  (or wavelength) of emitted pulses as well as the pulse emission rate  $r \in [0, 1]$  based on the target's proximity. While echo intensity can vary in different ways, it is generally assumed to decrease from an initial maximum value  $A_0$  to a minimum threshold  $A_{min}$ .

Many studies, for simplicity, do not incorporate ray tracing into this algorithm. Instead, they leverage variations in frequency  $f$  or wavelength  $\lambda$  to adapt to different applications, depending on factors such as ease of implementation [18]. Regarding this problem, LQR controller will be utilized in conjunction with BAT algorithm to determine the optimal Q and R parameter set.

First, we determine the parameters of algorithm and generate an initial BAT population. The following key parameters are initialized in Table 4:

Table 4. BAT algorithm initialization parameters

Parameters	Definition
$n_{BAT}$	Number of BATs
$d$	Number of parameters to find
$N$	Maximum number of iterations
$A_0$	Initial loudness
$r_0$	Initial pulse emission rate
$L_b, U_b$	The search space limits
$Q_{max, min}$	The frequency limits

After defining the parameters of the BAT algorithm, first, we initialize the initial random position  $x_{i\_BAT}$ , velocity  $v_i$ , and frequency  $f_i$  of the bat.

$$x_{i\_BAT} = L_b + (U_b - L_b) \cdot \text{rand}(d) \quad (36)$$

$$v_i = 0 \quad (37)$$

$$f_i = Q_{minmax_{min}} \quad (38)$$

Next is to evaluate the quality of each initial solution in the overall scheme. If the solution violates constraints (i.e., any value  $\leq 0$ ), assign an infinite fitness value.

$$F(x_{i\_BAT}) = \infty, \text{ if } x_{i\_BAT} \text{ is invalid} \quad (39)$$

The weight matrices Q and R for LQR controller are constructed as follows to match the search process of BAT algorithm:

$$Q_{LQR_{BAT}} = \begin{bmatrix} x_{0_{BAT}} & 0 & 0 & 0 \\ 0 & x_{1_{BAT}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & x_{5_{BAT}} \end{bmatrix} \quad (40)$$

$$R_{LQR_{BAT}} = x_{6_{BAT}}$$

The optimal controller  $K$  is determined by solving the Riccati equation using the matrices  $A$ ,  $B$ ,  $Q$  and  $R$ .

$$K_{LQR_{BAT}} = lqr(A, B, Q_{LQR_{BAT}}, R_{LQR_{BAT}}) \quad (41)$$

Performance of controller is evaluated using the fitness function.

$$F(x_{i_{BAT}}) = \int_0^T \sum e_i + 1 \quad (42)$$

Where  $e_i$  is the error of  $x_i = [\phi \ \theta_1 \ \theta_2 \ \dot{\phi} \ \dot{\theta}_1 \ \dot{\theta}_2]^T$ . Updating of the best-known solution, best-known solution  $x^*$  is updated by selecting solution with minimum fitness value.

$$x^* = \underset{x_{i_{BAT}}}{\operatorname{argmin}} F(x_{i_{BAT}}) \quad (43)$$

Update bat positions  $x_{i_{BAT\_U}}^{(t-1)}$ , velocities  $v_{i_{BAT\_U}}^{(t-1)}$ , and frequencies  $f_{i_{BAT\_U}}$  iteratively.

$$f_{i_{BAT\_U}} = Q \min \max_{\min} \quad (44)$$

$$v_{i_{BAT\_U}}^{(t)} = v_{i_{BAT\_U}}^{(t-1)} + (x_{i_{BAT\_U}}^{(t-1)} - x^*) f_{i_{BAT\_U}} \quad (45)$$

$$x_{i_{BAT\_U}}^{(t)} = x_{i_{BAT\_U}}^{(t-1)} + v_{i_{BAT\_U}}^{(t-1)} \quad (46)$$

To ensure adherence to the defined search space, boundary constraints are enforced on the position.

$$x_{i_{BAT\_U}} = \max(\min(x_{i_{BAT\_U}}, U_b), L_b) \quad (47)$$

If a bat emits fewer pulses, it performs a local search near the best-known solution.

$$\text{If } \text{rand} > r_i, x_{i_{BAT\_U}} = x^* + \epsilon A_t \quad (48)$$

Where:  $\epsilon \sim U(-1,1)$ . Next, repeat the process from formula number (39) to (42). The solution is updated if an improvement is observed.

$$\text{If } F(x_{i_{BAT\_U}}) \leq F(x_i) \text{ and } \text{rand} < A_i, \text{ then } x_i = x_{i_{BAT\_U}} \quad (49)$$

The algorithm parameters are adjusted to decrease loudness and increase the pulse emission rate, thereby achieving a balance between exploration and exploitation. Decrease loudness over iterations:

$$A_i^{(t+1)} = \alpha A_i^{(t)} \quad (50)$$

Increase pulse emission rate:

$$r_i^{(t+1)} = r_i^{(0)} [1 - e^{-\gamma t}] \quad (51)$$

Where  $\alpha$  and  $\gamma$  are optional constants that can be adjusted to suit specific cases. The best solution found so far has been retained. If a new solution outperforms the current global best, it is updated accordingly.

$$x^* = x_{i_{BAT\_U}}, Fi_{BAT\_U_{min}} \quad (52)$$

The entire BAT algorithm is iteratively repeated until the termination condition of loop  $N$  is met.

Fig. 3 illustrates a block diagram of controller synthesis utilizing LQR method integrated with BAT algorithm.

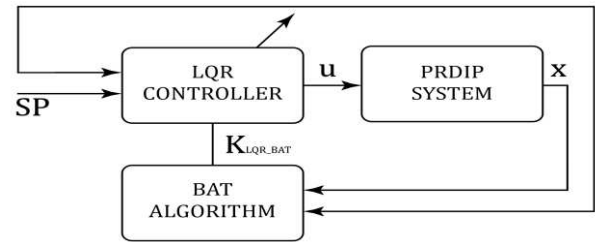


Fig. 3. Block diagram of the controller.

#### IV. RESULTS AND DISCUSSION

Parameter set obtained through the trial-and-error method as well as the parameter set derived from the BAT algorithm. The results from both approaches were then compared to evaluate their effectiveness. To synthesize LQR controller, we first select parameter determination scenario as follows: bring pendulums from initial position  $x_i = [0.02 \ 0.01 \ 0.01 \ 0.0001 \ 0.001 \ 0.008]^T$  to original equilibrium position  $x_i = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ .

##### A. Traditional LQR control

Parameter values for  $Q$  and  $R$  are specified as follows.

$$Q_{LQR} = \begin{bmatrix} 30^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^8 \end{bmatrix} \quad (53)$$

$$R_{LQR} = 10$$

Control parameters are determined by the usual trial and error method and the resulting  $K$  matrix is

$$K_{LQR} = 10^5 * [0.0005 \ 4.1749 \ -4.2925 \ 0.0021 \ 0.9224 \ -0.9575] \quad (54)$$

The simulation results of traditional LQR controller are shown in Fig. 4, Fig. 5, Fig. 6, and Fig. 7. Fig. 4 shows position of the crank arm. It shows initial oscillation amplitude of approximately 0.12 radians. The system exhibits a significant damping effect, with oscillations gradually decreasing over time; however, complete stabilization is achieved after approximately 30 seconds, indicating a



relatively slow convergence. In contrast, Fig. 5 shows the position of Pendulum 1, which demonstrates an initial oscillation amplitude of around  $\pm 0.01$  radians and stabilizes fully within approximately 7 seconds, reflecting a notably faster convergence compared to the crank arm. Similarly, Fig. 6 presents position of Pendulum 2, with initial oscillation amplitude of approximately  $\pm 0.01$  radians, comparable to Pendulum 1. However, Pendulum 2 requires slightly more time to stabilize, reaching full stability after approximately 8 seconds. These results highlight the differences in stabilization dynamics between the crank arm and the

pendulums, with the latter exhibiting significantly faster convergence and more efficient damping.

Fig. 7 shows control signal  $U$ , which initially exhibits significant oscillations, reaching amplitudes of approximately  $\pm 12V$ . Over time, the signal demonstrates a gradual stabilization, with oscillations diminishing significantly after approximately 10 seconds, ultimately maintaining a stable and reduced level. This behavior indicates the effective convergence of the control system toward a steady-state condition.

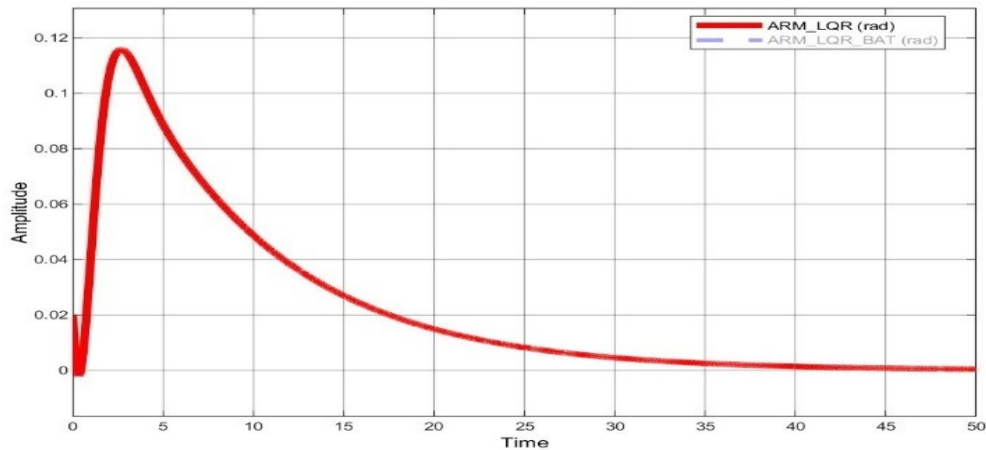


Fig. 4. Angle of arm

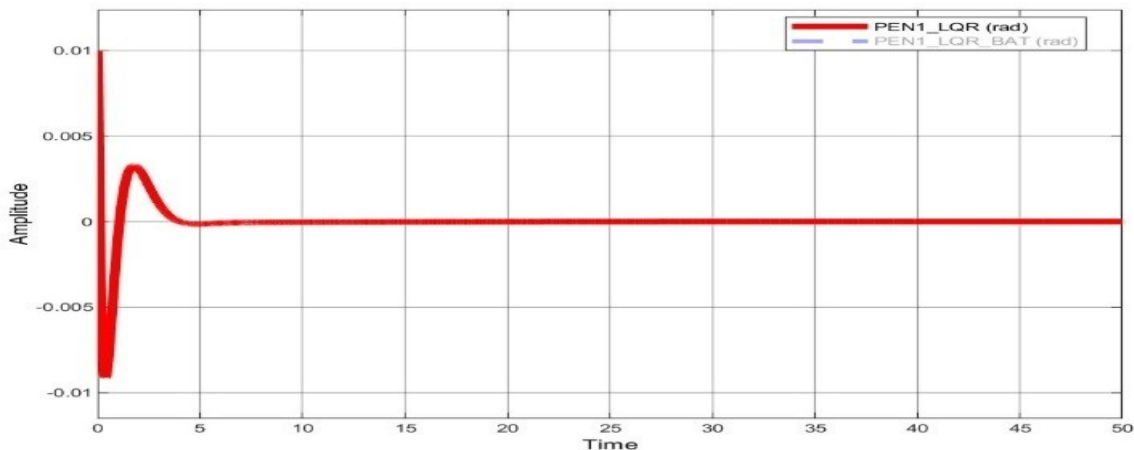


Fig. 5. Angle of pendulum 1

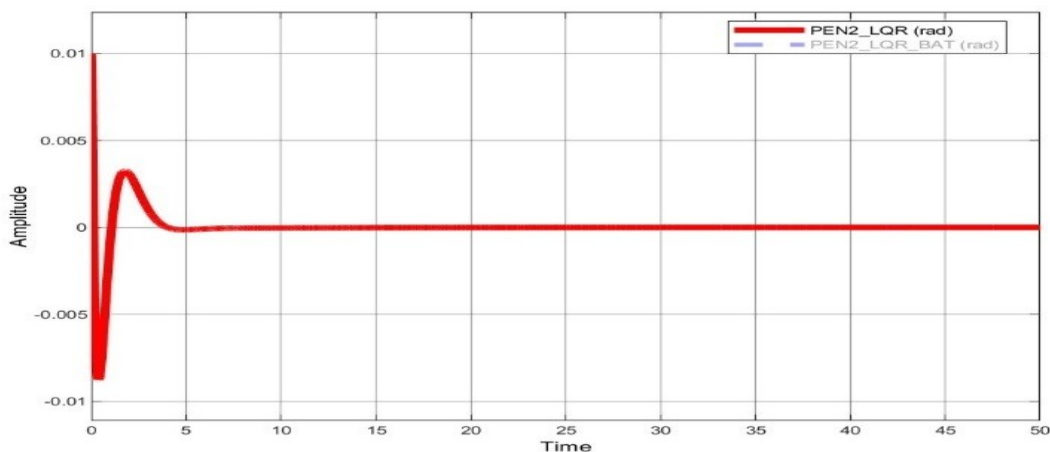


Fig. 6. Angle of pendulum 2

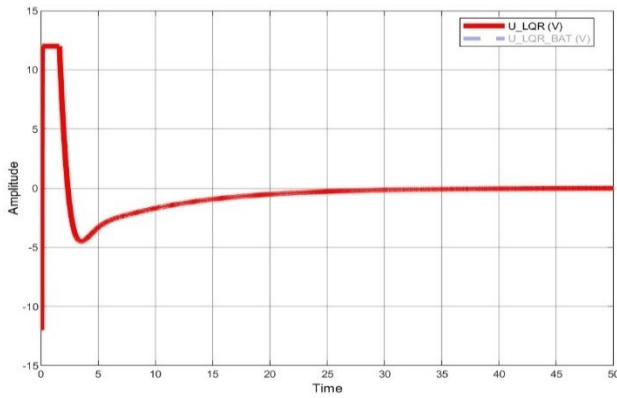


Fig. 7. Signal control U

**B. LQR Controller optimized by BAT algorithm**

In this section, we present results obtained from determining  $Q$  and  $R$  parameters of LQR controller using BAT algorithm can be seen in Table 5.

$$Q_{LQR\_BAT} = \begin{bmatrix} x_{0\_BAT} & 0 & 0 & 0 & 0 & 0 \\ 0 & x_{1\_BAT} & 0 & 0 & 0 & 0 \\ 0 & 0 & x_{2\_BAT} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{3\_BAT} & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{4\_BAT} & 0 \\ 0 & 0 & 0 & 0 & 0 & x_{5\_BAT} \end{bmatrix} \quad (54)$$

$$R_{LQR\_BAT} = x_{6\_BAT}$$

Table 5. BAT algorithm parameters table found.

Parameters	Values
$x_{0\_BAT}$	948874.140131
$x_{1\_BAT}$	40401826.002
$x_{2\_BAT}$	4103244.42137
$x_{3\_BAT}$	29382554.1244
$x_{4\_BAT}$	3193948.6724
$x_{5\_BAT}$	86451296.7993
$x_{6\_BAT}$	4.89291747511

Control parameters are determined using BAT algorithm and the resulting  $K$  matrix is calculated as

$$K_{LQR\_BAT} = 10^6 * [[0.0004 \ 0.9697 \ -1.0224 \ 0.0026 \ 0.2142] - 0.2280] \quad (54)$$

Simulation results of LQR controller combined with BAT algorithm are shown in Fig. 8 to Fig. 11. Fig. 8 shows the position of the crank arm, showing initial oscillation amplitude of approximately 0.04 rad. The system exhibits a gradual reduction in oscillations, stabilizing after approximately 10 sec, with a smooth decrease in amplitude and no excessive overshoot, indicating an effective damping effect. Similarly, Fig. 9 shows the position of Pendulum 1, which demonstrates initial oscillation amplitude of around  $\pm 0.005$  rad. The system converges rapidly, achieving full stabilization within 3 sec, reflecting highly effective control with minimal residual fluctuations. Fig. 10 presents the position of Pendulum 2, with an initial oscillation amplitude comparable to Pendulum 1 ( $\sim \pm 0.005$  rad). However, Pendulum 2 stabilizes fully after approximately 4 sec, slightly

later than Pendulum 1, yet still demonstrating quick damping and a well-balanced control strategy. These results collectively highlight the efficiency and robustness of the control system in achieving rapid stabilization across all components.

In Fig. 11, it shows control signal  $U$ , initially exhibits fluctuations. Amplitude ranges from  $\pm 12V$ . Signal gradually decreases and stabilizes within 6 seconds, demonstrating a smooth reduction in control effort. This behavior suggests efficient energy utilization while ensuring system stability. A comparison of convergence times across system components further highlights the effectiveness of the control strategy, with the crank arm stabilizing in approximately 10 sec, Pendulum 1 in 3 sec, and Pendulum 2 in 4 sec, collectively underscoring balanced and rapid stabilization achieved by the control approach.

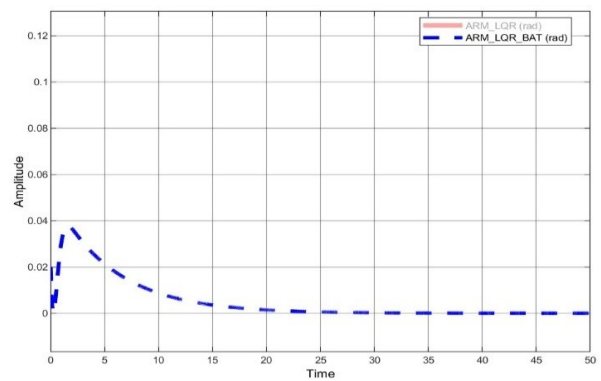


Fig. 8. Angle of arm

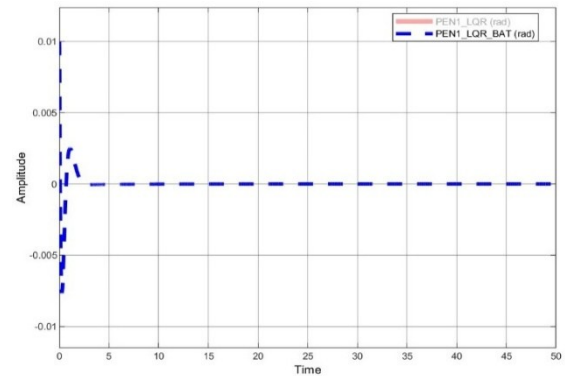


Fig. 9. Angle of pendulum 1

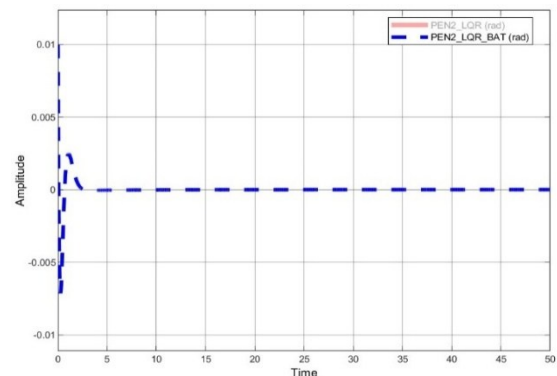


Fig. 10. Angle of pendulum 2

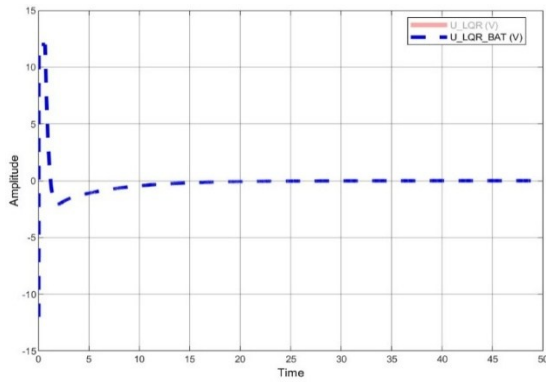


Fig. 11. Signal control U

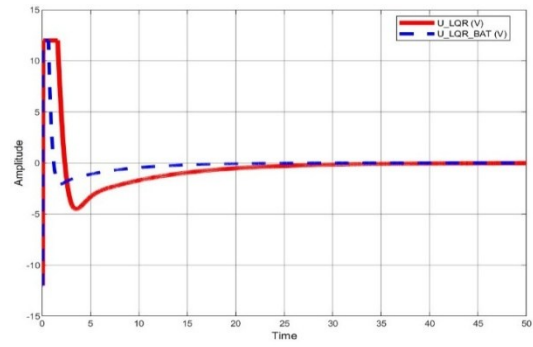


Fig. 15. Signal control u

C. Compare results

The simulation comparison of state variables of the system under traditional LQR and optimized LQR controllers is shown in Fig. 12 to Fig. 15. The details of these figures are listed in Table 6 to Table 8 for discussion.

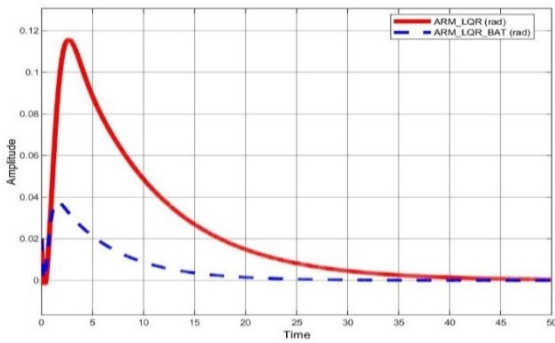


Fig. 12. Angle of arm

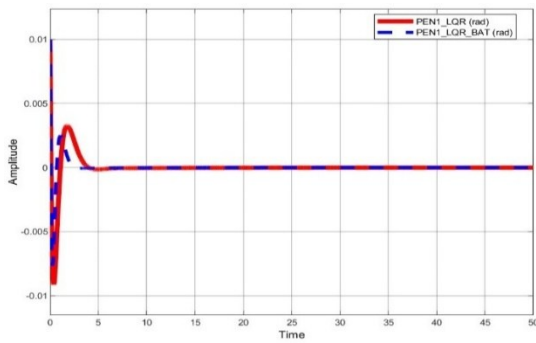


Fig. 13. Angle of pendulum 1

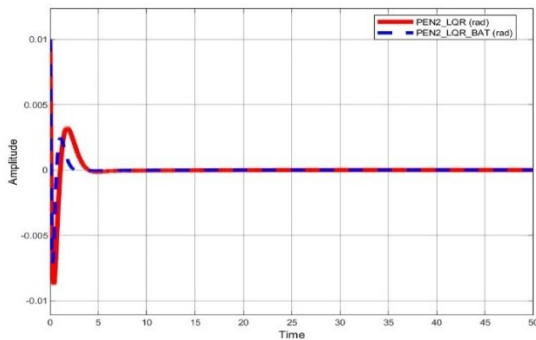


Fig. 14. Angle of pendulum 2

a) Stabilization Time:

In Table 6, the comparison of settling times of variables is shown for both traditional LQR and BAT-optimized LQR. In Table 6, LQR + BAT Algorithm significantly improves stabilization time across all components. The crank arm now stabilizes three times faster, while the pendulums converge in half the time compared to the traditional LQR.

Table 6. Comparison of Stabilization Time.

Component	LQR Traditional	LQR + BAT algorithm
Arm	~30 sec	~10 sec
Pendulum 1	~7 sec	~3 sec
Pendulum 2	~8 sec	~4 sec
Control Signal U	~10 sec	~6 sec

b) Oscillation Amplitude:

In Table 7, the vibrations of variables are compared in both traditional LQR and BAT-optimized LQR controllers. In Table 7, BAT algorithm significantly reduces the initial peak oscillations, resulting in a smoother and more controlled system response. Specifically, the crank arm's initial oscillation amplitude is reduced by 67%, demonstrating improved damping performance. Similarly, the pendulums exhibit a 50% reduction in oscillation amplitude, contributing to enhanced stability. While the control signal maintains the same peak amplitude, it stabilizes more rapidly, indicating both efficient energy utilization and faster convergence. These improvements collectively highlight the effectiveness of the BAT algorithm in optimizing system performance.

Table 7. Comparison of Initial Oscillation Amplitude.

Component	LQR Traditional	LQR + BAT algorithm
Arm	~0.12 rad	~0.04 rad
Pendulum 1	±0.01 rad	±0.005 rad
Pendulum 2	±0.01 rad	±0.005 rad
Control Signal U	±12V	±12V

c) Statistical Validation and Discussion of Limitations:

To further validate the performance improvement achieved by the BAT-optimized LQR controller, a statistical analysis was conducted based on multiple simulation runs. Specifically, we computed the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) of the angular positions of the crank arm and both pendulums. Table 8 presents the comparative results between the conventional LQR and the BAT-based LQR controllers.

In Table 8, statistical data further corroborate the visual observations. The BAT-based LQR consistently reduces both



the MAE and RMSE values across all components, indicating more precise tracking and better damping characteristics.

Table 8. Statistical performance comparison over 10 simulation runs

Component	Controller Type	MAE (Rad)	RMSE (Rad)
Arm	LQR	0.0385	0.0497
Arm	LQR + BAT	0.0127	0.0181
Pendulum 1	LQR	0.0096	0.0124
Pendulum 1	LQR + BAT	0.0042	0.0055
Pendulum 2	LQR	0.0103	0.0130
Pendulum 2	LQR + BAT	0.0049	0.0061

However, despite the improved performance, the proposed approach also introduces certain limitations that should be acknowledged. One primary concern is the computational cost associated with the BAT algorithm. Since the optimization requires multiple iterations of Riccati equation solving and dynamic simulations per candidate solution, the total computation time may become significant, especially for high-dimensional systems or real-time applications.

Furthermore, the current implementation assumes an offline optimization setting, where the optimal gain matrix is computed prior to deployment. Applying this approach to real-time systems would require additional considerations, such as real-time feasibility of matrix updates, computational load on embedded hardware, and stability under model uncertainties.

These aspects highlight the need for future work focused on hardware-in-the-loop testing, algorithmic simplification, or hybrid methods that combine fast-converging techniques with the global search capacity of BAT to balance performance and real-time capability.

## V. CONCLUSION

This study proposed a hybrid control design that combines LQR control with the BAT algorithm to stabilize RDPIP. By leveraging the global search capability of the BAT algorithm, the proposed method effectively optimized Q and R weighting matrices, overcoming limitations of conventional trial-and-error approaches. The simulation results demonstrated significant improvements in stabilization time, oscillation suppression, and control efficiency compared to the traditional LQR controller.

In addition to the simulation-based validation, statistical metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) confirmed the superior tracking accuracy of the BAT-optimized controller. These findings highlight the practical potential of integrating metaheuristic optimization with classical control strategies for under-actuated and nonlinear systems.

For future work, several concrete directions are recommended. First, real-time implementation of the proposed method on embedded hardware or robotic platforms should be investigated to evaluate its practical feasibility and robustness under physical uncertainties and disturbances. Second, comparative studies with other metaheuristic algorithms such as PSO, GA, or newer variants like Grey Wolf Optimizer (GWO) would further validate the effectiveness of BAT in this context. Additionally, developing a simplified or adaptive version of the BAT

algorithm with lower computational cost could enable online or real-time tuning of controller parameters.

Finally, the scalability and generalizability of this approach should be explored by applying it to other complex, high-order, or multi-input-multi-output (MIMO) control systems beyond the RDPIP, such as aerial vehicles, robotic manipulators, or balancing robots. Such extensions would demonstrate the broader applicability of the proposed control framework in diverse domains.

## ACKNOWLEDGMENT

This paper belongs to project SV2025-157 and it is funded by Ho Chi Minh City University of Technology and Education (HCMUTE). We, authors, want to give thanks for that support.

## REFERENCES

- [1] Fahmizal, Geonoky, and H. Maghfiroh, "Rotary Inverted Pendulum Control with Pole Placement," *J. Fuzzy Syst Control*, vol. 1, no. 3, pp. 90–96, 2023, <https://doi.org/10.59247/jfsc.v1i3.152>.
- [2] N.-C. Tran *et al.*, "LQR Control for Experimental Double Rotary Inverted Pendulum," *J. Fuzzy Syst Control*, vol. 2, no. 2, pp. 104–108, 2024, <https://doi.org/10.59247/jfsc.v2i2.212>.
- [3] P.-H. Huynh *et al.*, "Model Predictive Control for Rotary Inverted Pendulum: Simulation and Experiment," *J. Fuzzy Syst Control*, vol. 2, no. 3, pp. 215–222, 2024, <https://doi.org/10.59247/jfsc.v2i3.263>.
- [4] M. Abdullah *et al.*, "Swing up and stabilization control of rotary inverted pendulum based on energy balance, fuzzy logic, and LQR controllers," *Measurement and Control*, vol. 54, no. 9–10, pp. 1356–1370, 2021, <https://doi.org/10.1177/00202940211035406>.
- [5] I. Chawla and A. Singla, "Real-Time Control of a Rotary Inverted Pendulum using Robust LQR-based ANFIS Controller," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 19, no. 3–4, pp. 379–389, 2018, <https://doi.org/10.1515/ijnsns-2017-0139>.
- [6] Chawla *et al.*, "Robust LQR Based ANFIS Control of x-z Inverted Pendulum," *Amity International Conference on Artificial Intelligence*, pp. 818–823, 2019, <https://doi.org/10.1109/AICAI.2019.8701333>.
- [7] M. -T. Vo *et al.*, "Linear Quadratic Regulators Optimal Control of Rotary Double Parallel Inverted Pendulum," *International Conference on Control, Robotics and Informatics*, pp. 14–18, 2023, <https://doi.org/10.1109/ICCRI58865.2023.00009>.
- [8] C.-H. Nguyen *et al.*, "ANFIS-based LQR Control for Rotary Double Parallel Inverted Pendulum," *J. Fuzzy Syst Control*, vol. 2, no. 2, pp. 109–116, 2024, <https://doi.org/10.59247/jfsc.v2i2.214>.
- [9] Q. Yong *et al.*, "Balance control of two-wheeled self-balancing mobile robot based on TS fuzzy model," *Proceedings of 2011 6th International Forum on Strategic Technology*, pp. 406–409, 2011, <https://doi.org/10.1109/IFOST.2011.6021051>.
- [10] M. Park *et al.*, "Swing-up and LQR stabilization of a rotary inverted pendulum," *Artificial Life and Robotics*, vol. 16, pp. 94–97, 2011, <https://doi.org/10.1007/s10015-011-0897-9>.
- [11] R. Xiaogang *et al.*, "Design and LQ Control of a two-wheeled self-balancing robot," *2008 27th Chinese Control Conference*, pp. 275–279, 2008, <https://doi.org/10.1109/CHICC.2008.4605775>.
- [12] M. T. Vo *et al.*, "Combining Passivity Based Control and Linear Quadratic Regulator to Control a Rotary Inverted Pendulum," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 479–490, 2023, <https://doi.org/10.18196/jrc.v4i4.18498>.
- [13] N. Kawasaki and E. Shimemura, "Determining quadratic weighting matrices to locate poles in a specified region," *Automatica*, vol. 19, no. 5, pp. 557–560, 1983, [https://doi.org/10.1016/0005-1098\(83\)90011-0](https://doi.org/10.1016/0005-1098(83)90011-0).
- [14] N. Kawasaki *et al.*, "On the quadratic weights of an LQ-problem shifting only the specified poles," *Proceedings of the Society of Instrument and Control Engineers*, vol. 25, no. 11, pp. 1248–1250, 1989, <https://doi.org/10.9746/sicetr1965.25.1248>.
- [15] F. J. Kraus and V. Kučera, "Linear quadratic and pole placement iterative design," *1999 European Control Conference (ECC)*, pp. 653–658, 1999, <https://doi.org/10.23919/ECC.1999.7099379>.

- [16] F. J. Kraus and V. Kučera, "Linear quadratic and pole placement iterative design," *1999 European Control Conference (ECC)*, pp. 653-658, 1999, <https://doi.org/10.23919/ECC.1999.7099379>.
- [17] B. D. Anderson and J. B. Moore. *Optimal control: linear quadratic methods*. Courier Corporation. 2007. <https://books.google.co.id/books?hl=id&lr=&id=fW6TAwAAQBAJ>.
- [18] C. N. Xuan and T. Le Tran, "Design of State Feedback Controller with Optimal Parameters Using BAT algorithm for Reaction Wheel Pendulum," *2021 International Conference on Advanced Technologies for Communications (ATC)*, pp. 172-177, 2021, <https://doi.org/10.1109/ATC52653.2021.9598306>.
- [19] X. S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver press. 2010. [https://books.google.co.id/books?hl=id&lr=&id=iVB\\_ETlh4ogC](https://books.google.co.id/books?hl=id&lr=&id=iVB_ETlh4ogC).
- [20] X. S. Yang, "A new metaheuristic bat-inspired algorithm," In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp. 65-74, 2010, [https://doi.org/10.1007/978-3-642-12538-6\\_6](https://doi.org/10.1007/978-3-642-12538-6_6).
- [21] Thamir, Luay, "Bat Algorithm Based an Adaptive PID Controller Design for Buck Converter Model," *Journal of Engineering*, vol. 26, pp. 62-82, 2020, <https://doi.org/10.31026/j.eng.2020.07.05>.
- [22] G. Li, H. Xu, and Y. Lin, "Application of bat algorithm based time optimal control in multi-robots formation reconfiguration," *Journal of Bionic Engineering*, vol. 15, no. 1, pp. 126-138, 2018, <https://doi.org/10.1007/s42235-017-0010-8>.
- [23] O. B. Alzain *et al.*, "optimization of Sliding Mode Control based on BAT-Algorithm for the DFIG-WT," *International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-7, 2021, <https://doi.org/10.1109/ICCCEEE49695.2021.9429587>.
- [24] C. Abdelghani *et al.*, "Robust design of fractional order PID Sliding Mode based Power System Stabilizer in a power system via a new metaheuristic Bat algorithm," *International Workshop on Recent Advances in Sliding Modes (RASM)*, pp. 1-5, 2015, <https://doi.org/10.1109/RASM.2015.7154651>.