International Journal of Engineering &Technology, 13 (2) (2024) 294-300

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Optimal control for two wheeled self-balancing robot based on butterfly optimization algorithm

Ghaidaa Hadi Salih Elias *

College of Computer Science and Information Technology, Kerbala University, Kerbala, Iraq
**Corresponding author E-mail: gadahadi6@gmail.com*

Abstract

This paper examined the two-wheeled self-balancing robot mathematical model. A Linear Quadratic Regulator (LQR) controller was then successfully used by the author for this system. The trial and error method and two optimization algorithms, particle swarm optimization (PSO) and butterfly optimization algorithm (BOA), are suggested for tuning the LQR controller parameters. The comparison between the three LQR controller tuning techniques is performed to select the best one. With the help of the Python program, the performance of the control strategy is investigated and shown with regard to the tilt and heading angles. Controlling outcomes were tested through a simulation, and it was proved that the LQR control system based on the BOA succeeded in finding a better response in tracking tilt and heading angles with enhancement percentages of 61.111% and 78.348% than the LQR response based on PSO and the trial and error techniques, respectively.

Keywords: *Self-Balancing Robot; Control Gains; PSO Algorithm; Optimal LQR Control; Butterfly Optimization Algorithm.*

1. Introduction

A common multi input, multi output model in control system experiments is the two-wheeled self-balancing robot, often known as the inverted pendulum and cart models [1]. The two wheeled self-balancing robot is a common robot with potential uses in exploration and transportation, among other fields. Over the past few decades, two-wheeled self-balancing robot design and control have garnered a lot of interest from both industry and academics. The two-wheeled self-balancing robot is a high-order, multivariable, nonlinear, tightly coupled, and intrinsically unstable system that exemplifies an underactuated mechanical system. Underactuated mechanical systems have fewer control inputs than robot's degrees of freedom that need to be become stable. This makes it challenging to apply classical robotic approaches for system control. Therefore, a two wheeled self-balancing robot is an excellent platform for researchers to study the effectiveness of different controllers in control systems.

Numerous academics had suggested various controller designs and analytical techniques to manage the two wheeled self-balancing robot so that it could balance itself [1][2]. A fuzzy logic-based controller for an inverted pendulum model was developed and successfully tested in [3][4][5]. In [6], a linear state-space model for motion control of a two wheeled self-balancing robot was developed. A Newtonian method was used in [7] to calculate dynamics, and the control was designed using equations that were linearized around an operational point. In [8][9][10] and [11], a planar model was used to build a linearly stabilized LQR controller and Proportional Integral Derivative (PID) without taking the robot's heading angle into account. The aforementioned control law was created using a planar model without taking the robot's heading angle into account, so it cannot be applied to an actual system.

Control engineers have used a variety of optimization techniques to find the best values for the components of LQR matrices, including artificial bee colony [12], PSO algorithm [7][12], the combined simulated annealing and modified genetic algorithm [13], bacteria foraging optimization algorithm [14], genetic algorithm [12][14], differential evaluation [15], and ant colony optimization [16].

In this research, the two wheeled self-balancing robot system serves as the study object, and the dynamic equation is derived using the Newtonian mechanics equation method. By assuming that the system runs only around a single operational point and that the signals involved are tiny signals, a linear state-space model that roughly approximates the nonlinear system in the region of operation can be created. The LQR controller is employed the state-space model to regulate the system's tilt and heading angles, allowing the system to be directed to travel to a specific point. The trial and error method and two optimization algorithms, PSO and BOA, are suggested for tuning the diagonal of the LQR controller parameters (Q and R matrices). The comparison between three LQR tuning techniques is performed to select the best one. With the help of the Python program, the performance of the control strategy is investigated and shown with regard to the tilt angle ($\boldsymbol{\alpha}$) and heading angle (Θ).

The remaining paper is as follows: The method of the LQR control system is modeled in Section 2. The LQR control systems, with the suggested algorithms, are designed in Section 3. Experimental work results are discussed in Section 4, and the summarizing of the work in the paper is concluded in Section 5.

2. Model system

Fig. 1 illustrates the three main components of the robot: the platform, the pendulum, and the pendulum as the mass wheels. The system receives inputs in the form of torques, which are applied to the robots' two left and right-side wheels, respectively. The control system schemes' goals are to move the system's model to the required location while maintaining the robot's tilt angle in the vertical position [1].

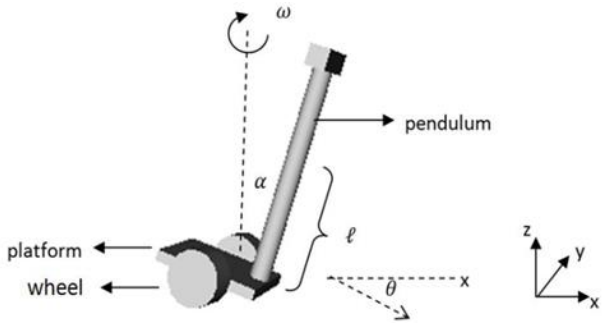

Fig. 1: A Two-Wheeled Self-Balancing Robot Model [1].

Where F_r, F_l are forces interacting between the platform and the right and left wheels (N), H_r, H_l are the friction forces applying on the right and left wheels (N), τ_r, τ_l are torques acting on the right and left wheels (N/m), Θ_l, Θ_r angles of rotation for the left and right wheels (rad), x_r, x_l are the movement of the left and right wheels relative to the x-axis (m), $\boldsymbol{\alpha}$ is the robot's tilt angle (rad), Θ is the robot's heading angle in relation to the z-axis, M, m are the wheel's mass and the pendulum's mass respectively (kg), I_w is the wheel's moment of inertia around the y-axis (kg.m^2), r indicates to the wheel's radius (m), g is the acceleration of gravity (m/s^2), l is the measurement along the z-axis from the robot's platform to the pendulum's center of mass (m), d represents the distance along the y-axis between the left and right wheels (m), I_M specifies platform's moment of inertia relative to the y-axis (kg.m^2), I_p denotes the platform and pendulum's moment of inertia with respect to the z-axis (kg.m^2), F_P interacting forces between the platform and the pendulum relative to the x-axis (N), M_P interaction moment about the y-axis between the platform and the pendulum (N/m), v is the robot's forward velocity (m/s), and $\boldsymbol{\omega}$ determines the robot's rotation velocity. According to the Newton law, forces and torques related to the right and left wheels are selected in Equations (1 to 4) [1]:

$$I_\omega \ddot{\Theta}_r = r(F_r - H_r) \tag{1}$$

$$M_w \ddot{x}_r = H_r - F_r \tag{2}$$

$$I_\omega \ddot{\Theta}_l = r(F_l - H_l) \tag{3}$$

$$M_w \ddot{x}_l = H_l - F_l \tag{4}$$

The pendulum's moments applied on the platform about the y-axis and forces operating on the pendulum in the direction of the x-axis are balanced to produce [1]:

$$M\ddot{x} = F_r + F_l + F_p \tag{5}$$

$$F_p = m(l\ddot{\alpha}\cos\alpha + \ddot{x} - l\dot{\alpha}^2 \sin\alpha) \tag{6}$$

$$M_p = ml(l\ddot{\alpha} + \cos\alpha\,\ddot{x} + g\sin\alpha \tag{7}$$

$$M_p = -I_m\ddot{\alpha} \tag{8}$$

The balancing between pendulums' moments and the moments acting on the platform around the z-axis, the result gives [1]:

$$I_p\ddot{\Theta} = (F_l - F_r)d \tag{9}$$

The dynamic model of the robot is then linearized around the point $\alpha = \dot{\alpha} = 0$, $\cos\alpha = 1$, $\sin\alpha = \alpha$, assuming that the robot only operates around a limited operating point. So, the state space equation of the balancing robot motion is selected in Equation (10) [1].

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{v} \\ \dot{\Theta} \\ \ddot{\Theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & 0 \\ a_{31} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ v \\ \Theta \\ \dot{\Theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ 0 & 0 \\ b_{51} & b_{52} \end{bmatrix} \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} \tag{10}$$

Here,

$$s = m^2 l^2 + ((M - m + 2((\frac{I_\omega}{r^2} + M_w)(ml^2 + I_M)) \tag{11}$$

$$a_{21} = -mlg(M - m + 2(\frac{I_\omega}{r^2} + M_w))/s) \tag{12}$$

$$a_{31} = -(m^2 l^2 g)/s \quad (13)$$

$$b_{21} = -ml/s,\ b_{22} = b_{21} \quad (14)$$

$$b_{31} = (ml^2 + I_M)/s,\ b_{32} = b_{31} \quad (15)$$

$$b_{51} = d/r(I_p + d^2(M_w + (I_w/r^2)),\ b_{52} = -b_{51} \quad (16)$$

The values of the systems' parameters are listed in Table 1 [1].

Table 1: Robots' Parameters

Parameter	Value	Unit
I_w	0.0313	kg.m^2
M_w	1	Kg
m	70	Kg
R	0.250	m
l	1	m
g	9.80	m/s^2
M	5	Kg
d	0.5	m
I_p	1.8569	kg.m^2
I_M	0.0385	kg.m^2

According to the data involved in Table 2, the robots' state space equations are selected in Equation (17) [1].

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \\ \dot{v} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 76.2755 & 0 & 0 & 0 & 0 \\ -86.1175 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \\ v \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -0.5021 & -0.5021 \\ 0.5024 & 0.5024 \\ 0 & 0 \\ 0.8961 & 0.8961 \end{bmatrix} \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} \quad (17)$$

3. LQR controller system design

The LQR approach is the most developed method for controller design in the evolution of modern control theory. It uses state-space methods to analyze a system has many output and input. Fig. 2 illustrates the schematic for this kind of two wheeled self-balancing robot control system [1].

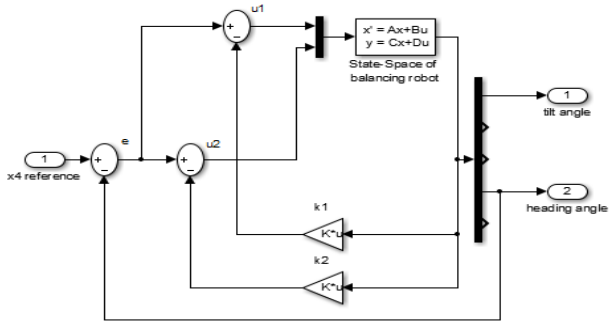

Fig. 2: Control Schematic for the Two-Wheeled Self-Balancing Robot.

Where,

$$ku(t) = -kx(t) \quad (18)$$

$$k = R^{-1}B^T P \quad (19)$$

Where u is the LQR control input, k is the feedback gain, P and R are square symmetric, positive definite matrices. The P matrix is selected by solution of the algebraic Riccarti equation matrix defined in Equation (20) [1][17].

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (20)$$

Q is the square symmetric positive semidefinite matrix. The Q and R matrices are tuned by the BOA, PSO, and trail and error techniques, and they are denotes in Equation (21) [17].

$$Q = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & d_3 & 0 & 0 \\ 0 & 0 & 0 & d_4 & 0 \\ 0 & 0 & 0 & 0 & d_5 \end{bmatrix}, R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \quad (21)$$

The Q and R matrices are properly tuned in order to reduce the integral absolute error (IAE) denoted in Equation (22) [18].

$$IAE = \int_0^\infty |e(t)| \, . \, dt \tag{22}$$

3.1. LQR design based on the PSO (LQR-PSO)

PSO is a new meta-heuristic optimization method that has been explored by the researchers Eberhart and Kennedy since 1995, and it is dependent on the swarm of individuals or particles. It uses an iterative process to explore random solutions until it finds the best one. Each individual in the swarm represents a particular solution, and it is given a position and speed, which they update at every generation to generate many solutions. These solutions are evaluated to select the global best and particle past. The global best is the best solution across all generations; on the other hand, the particle best stands for the best solution from each generation. The global best and the particle best are utilized to define the new position and speed in accordance with Equations (23 to 25) [17][18].

$$\omega = \omega_{max} - t \, (\omega_{max} - \omega_{min})/N \tag{23}$$

$$v_i^{t+1} = \omega \, v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g_i^t - x_i^t) \tag{24}$$

$$x_i^{t+1} = v_i^{t+1} + x_i^t \; i = 1, 2, \dots, n \tag{25}$$

Where, t, n, c_1, c_2, ω, r_1, r_2, p_i, g_i, N, ω_{min}, ω_{max}, x_i, and v, are iterations number, individuals' numbers, cognitive learning element, social learning element, inertia weight, random value inside the range (0,1), random value inside the range (0,1), the best position of the individuals, the most effective individual among group members, maximum iteration, the inertia weight's minimum value, the inertia weight's maximum value, particle's position, and the particle's velocity, respectively. This process is repeated until reaching the maximum bound of the repetition. The boundaries of the PSO technique are scheduled according to Table 2 [17][18].

Table 2: PSO Techniques' Boundaries

Description	Symbol	Value
Minimum and maximum bounds of inertia weights	ω_{min}, $\omega_{\max}$	0.4 and 0.7
Cognitive learning element	c_1	2
Social learning element	c_2	2
Velocity's maximum bound	v_{max}	6
Population size	n	20
Maximum bound of the repetition	N	300
Lower gains bounds	lb	[0, 0, 0]
Upper gains bounds	ub	[50, 50, 50]

3.2. LQR design based on the BOA (LQR-BOA)

The BOA is a modern meta-heuristic method that Singh and Arora developed in 2018. It is inspired by butterflies' natural behavior in the search for food. This technique is dependent on two matters, which are the fragrance function formulation and the difference in fragrance intensity. The formulation of the fragrance function can be represented according to Equation (26) [20].

$$f_i = cI^\tau \tag{26}$$

Where, f_i is the fragrance's perceived strength, c selects the sensory modality, I denotes the stimulus intensity, and τ signifies the power exponent based on how much of the smell is absorbed. The sensory modality c may be updated as in Equation (38) [20].

$$c_{t+1} = c_t + \frac{0.025}{c_t * N} \tag{27}$$

Where N states maximum iteration. As that, the butterfly's algorithm involves global search and local search. The worldwide search is the behavior of a butterfly when it detects the scent coming from another butterfly and moves toward it. Contrarily, the local search occurs when the butterfly transfers at random when it is unable to detect the fragrance coming from any other butterfly. The formulas for the global and local searches are explained in Equations (28) and (29), respectively [20].

$$X_i^{t+1} = X_i^t + (r^2 * X^* - X_i^t) * f_i \tag{28}$$

$$X_i^{t+1} = X_i^t + \left(r^2 * X_j^t - X_k^t\right) * f_i \tag{29}$$

Where $\boldsymbol{X}^*$is the optimal solution and r denotes random number between the interval 0 and 1. The mode to change between global and local search is named a switch probability and indicated the symbol P. The boundaries of the BOA that were used in that work are stated in Table 3 [20].

Table 3: BOA Techniques' Boundaries

BOA technique's parameter name	Symbol	Value
Switch probability	P	0.1
Absorbed smell amount	$\boldsymbol{\tau}$	Between 0.1 and 0.7
Sensory modality	c	0.01
Population size	n	20
Maximum bound of the repetition	N	300
Lower gains bounds	lb	[0, 0, 0]
Upper gains bounds	ub	[50, 50, 50]

Figs. 3 and 4 offer the flowchart and the anticipated block built chart for the optimal LQR controller, respectively.

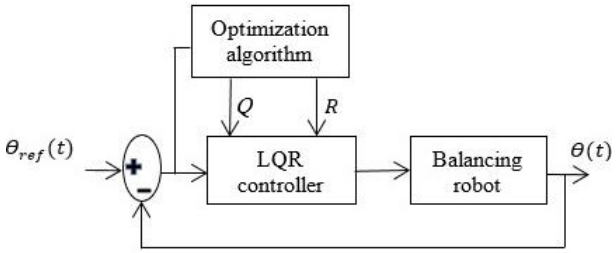

Fig. 3: Optimal LQR Chart.

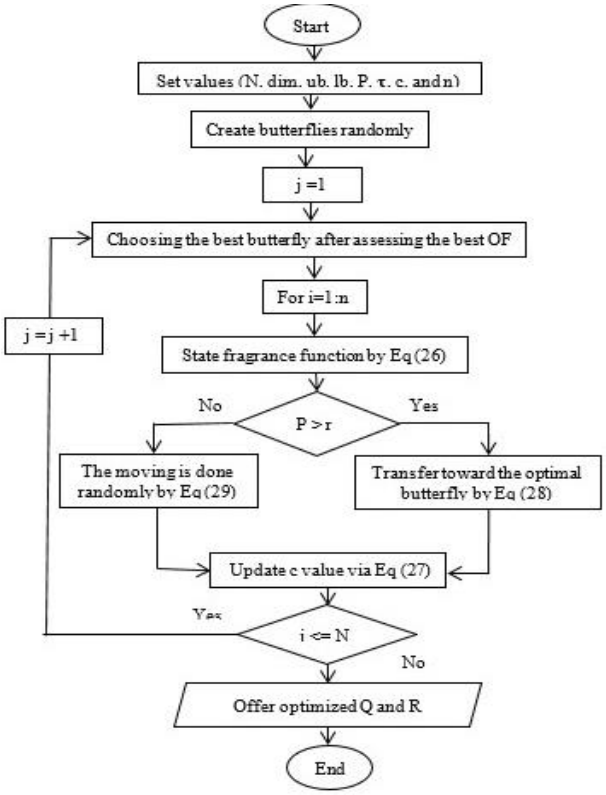

Fig. 4: LQR-BOA Flowchart.

4. Simulation results and discussion

In this part, the simulation outcomes of the suggested controllers, which are achieved on the model of a robot, are presented. The Python simulation program is used to stimulate and design the LQR, LQR-PSO, and LQR-BOA controllers for the system. Controllers' performance characteristics are also included in this section.

A two wheeled self-balancing robot system with each control system (LQR, LQR-PSO, or LQR-BOA) produced two output responses: the robot's heading angle ($\boldsymbol{\theta}$) and tilt angle ($\boldsymbol{\alpha}$). In order to examine the effectiveness of the controller, the initial tilt angle of the balancing robot in this study was set to -1 rad, and the initial state of the heading angle was set to 0 rad. On the other hand, the reference heading angle is given a unit-step signal. The results of Tables (4 to 6) comprise the best Q and R diagonal matrices in addition to the minimum performance index offered by the LQR, LQR-PSO, and LQR-BOA controllers.

Table 4: Optimal Diagonal for the Q Matrix

Optimal controller	q_1	q_2	q_3	q_4
LQR	0.1	0.1	0.1	0.001
LQR [1]	10	1	1	10
LQR-PSO	6.903	38.60	26.18	0
LQR-BOA	0.053	0.017	0	0

Table 5: Optimal Diagonal for the R Matrix

Optimal controller	r_1	r_2
LQR	0.8	0.4
LQR [1]	1	1
LQR-PSO	16.3308	13.9282
LQR-BOA	0.003	0.001

Table 6: Performance Index of the Optimal Controllers

Optimal controller	IAE
LQR	0.1045967
LQR [1]	1.01
LQR-PSO	0.0582339
LQR-BOA	0.0226465

As chaired in Table 6, the LQR-BOA method involves a better performance index in comparison to the LQR and LQR-PSO controllers. On the other hand, the feedback gains k for the LQR, LQR-PSO, and LQR-BOA controllers are listed in Table 7.

Table 7: Performance Index for the Optimal LQR Controllers

Optimal controller	K
LQR	-127 -143 0.019 0.003 4.077 -186 -20.9 0.418 -0.002 -3.786
LQR [1]	-153.462 -16.803 0.707 2.236 1.730 -153.462 -16.803 0.707 -2.236 -1.730
LQR-PSO	-1.20 -13.83 0 0 4.909 -207 -23.9 0 -0 -4.936
LQR-BOA	-1.44 -15.6 0.817 0 1.336 -163 -17.6 1.047 0 -1.223

The cost functions' convergence curve for the BOA and PSO algorithms is depicted in Fig. 5. The PSO method converged in this figure more slowly than the BOA method. That proves the suggested algorithm's strength.

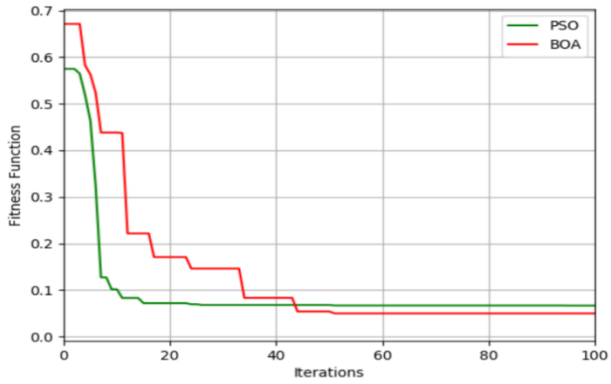

Fig. 5: Convergence Curve of the LQR-PSO and the LQR-BOA.

As shown in Figs 6 and 7, the optimal LQR controllers are capable of tracking reference signals for the tilt and heading angles. The LQR-BOA response for the tilt and heading angles is faster and smoother than the LQR and LQR-PSO controllers.

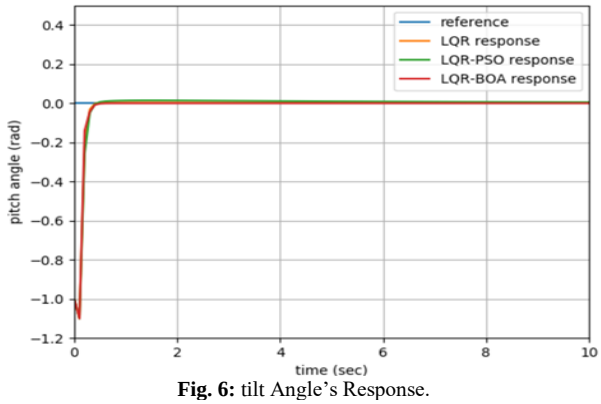

Fig. 6: tilt Angle's Response.

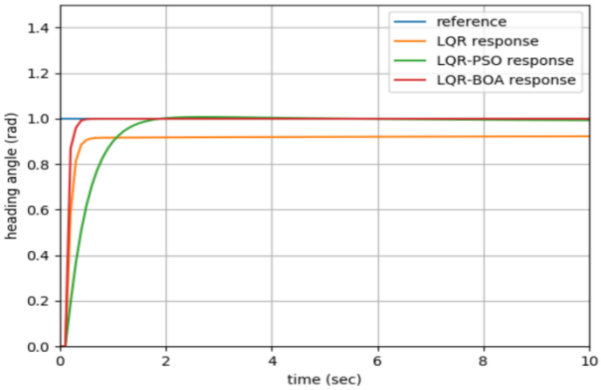

Fig. 7: Heading Angle's Response.

Table 8: Performance Standards of the Robot's Tilt Angle

Optimal controller	t_s (sec)	SSE (rad)
LQR	0.3427	0.0028
LQR [1]	2.23	0.0086
LQR-PSO	0.3629	0.0053
LQR-BOA	0.3528	0.0018

Table 9: Performance Standards of the Robot's Heading Angle

Optimal controller	t_s (sec)	SSE (rad)
LQR	0.494	0.0782
LQR [1]	2.76	0
LQR-PSO	1.4516	0.0069
LQR-BOA	0.373	0.0008

As chaired in Table 8, the LQR-BOA response has a smaller value of steady state error than the LQR, LQR [1], and LQR-PSO controllers, in addition to maintaining a small value of settling time. On the other hand, Table 9 content proves that the LQR-BOA response for the target heading angle has less settling time than the LQR, LQR [1], and LQR-PSO controllers in addition to maintaining a small value of steady state error than the LQR and LQR-PSO controllers.

5. Conclusion

This paper investigated the optimal LQR control system to stabilize the two wheel self-balancing robot. In order to design a LQR control system, the robot system dynamics must be modelled in state space. Trial and error method and two swarm optimization algorithms, PSO algorithm and BOA, are used for tuning the optimal LQR controller's parameters (diagonal Q and R matrices). Simulation outcomes of the tuned controllers are presented and then compared based on system stabilizing parameters. According to the comparison results, the LQR-BOA controller response for the tilt angle has a better value of steady state error than the LQR and LQR-PSO controllers, in addition to maintaining a small value of settling time. Besides, the heading angle response for the LQR-BOA controller has a better value of the steady state error and the settling time than the LQR and LQR-PSO controllers.

References

[1] M. O. Asali, F. Hadary, and B. W. Sanjaya, "Modeling , Simulation , and Optimal Control for Two-Wheeled Self-Balancing Robot," Int. J. Electr. Comput. Eng., vol. 7, no. 4, p. 2008, 2017, https://doi.org/10.11591/ijece.v7i4.pp2008-2017.

[2] L. Guo, S. A. A. Rizvi, and Z. Lin, "Optimal control of a two-wheeled self-balancing robot by reinforcement learning," Int. J. Robust Nonlinear Control, vol. 31, no. 6, pp. 1885–1904, 2021, https://doi.org/10.1002/rnc.5058.

[3] W. Wang, "Adaptive Fuzzy Sliding Mode Control for Inverted Pendulum," in Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCSCT '09), 2009, vol. 7, no. 4, pp. 231–234.

[4] T. Yong-chuan, L. Feng, Q. Qian, and Y. Yang, "Stabilizing Planar Inverted Pendulum System Based on Fuzzy Nine-point Controller," TELKOMNIKA Indones. J. Electr. Eng., vol. 12, no. 1, pp. 422–432, 2014, https://doi.org/10.11591/telkomnika.v12i1.4146.

[5] T. O. S. Hanafy and M. K. Metwally, "Simplifications the Rule Base for Stabilization of Inverted Pendulum System," TELKOMNIKA Indones. J. Electr. Eng., vol. 12, no. 7, pp. 5225–5234, 2014, https://doi.org/10.11591/telkomnika.v12i7.5358.

[6] M. ul Hasan, K. M. Hasan, M. U. Asad, U. Farooq, and J. Gu, "Design and Experimental Evaluation of a State Feedback Controller for Two Wheeled Balancing Robot," in 17th IEEE International Multi Topic Conference, 2014, pp. 366–371, https://doi.org/10.1109/INMIC.2014.7097367.

[7] J. Fang, "The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm," Math. Probl. Eng., vol. 2014, p. 6, 2014, https://doi.org/10.1155/2014/729095.

[8] K. Prakash and K. Thomas, "Study of Controllers for a Two Wheeled Self- Balancing Robot," 2016, https://doi.org/10.1109/ICNGIS.2016.7854009.

[9] H. S. Zad, A. Ulasyar, A. Z. Zohaib, and S. S. Hussain, "Optimal Controller Design for Self-balancing Two-wheeled Robot System," in 2016 International Conference on Frontiers of Information Technology Optimal, 2016, pp. 11–16, https://doi.org/10.1109/FIT.2016.011.

[10] C. Sun, T. Lu, and K. Yuan, "Balance control of two-wheeled self-balancing robot based on Linear Quadratic Regulator and Neural Network," in Proceedings of the 2013 International Conference on Intelligent Control and Information Processing, ICICIP 2013, 2013, vol. 1, pp. 862–867, https://doi.org/10.1109/ICICIP.2013.6568193.

[11] M. Majczak and P. Wawrzynski, "Comparison of two efficient control strategies for two-wheeled balancing robot," in 2015 20th International Conference on Methods and Models in Automation and Robotics, MMAR 2015, 2015, pp. 744–749, https://doi.org/10.1109/MMAR.2015.7283968.

[12] A. I. Abdulla, I. K. Mohammed, and A. M. Jasim, "Roll Control System Design Using Auto Tuning LQR Technique," Int. J. Eng. Innov. Technol., vol. 7, no. 1, pp. 10–17, 2017.

[13] "Determination of the Weighting Parameters of the LQR System for Reactor Power Control Using the Stochastic Searching Method," J. Korean Nucl. Soc., vol. 29, no. 1, pp. 68–77, 1997.

[14] I. K. Mohammed and A. I. Abdulla, "Balancing a Segway robot using LQR controller based on genetic and bacteria foraging optimization algorithms," TELKOMNIKA Telecommun. Comput. Electron. Control, vol. 18, no. 5, pp. 2642–2653, 2020, https://doi.org/10.12928/telkomnika.v18i5.14717.

[15] R. M. Storn and R. & S. Gmbh, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Differential Evolution - A simple and efficient adaptive scheme for global by Rainer Storn," J. Glob. Optim., vol. 11, no. May, pp. 341–359, 2019, https://doi.org/10.1023/A:1008202821328.

[16] D. Ali and H. Messaoud, "Optimized eigenstructure assignment by ant system and LQR approaches," Int. J. Comput. Sci. Appl., vol. 5, no. 4, pp. 45–56, 2008, doi: https://www.researchgate.net/publication/26621998.

[17] D. S. Naidu, OPTIMAL CONTROL SYSTEMS. Boca Raton,London,New York: CRC Press, 2003.

[18] G. Hadi, S. Elias, A. Al-moadhen, and H. Kamil, "Lateral Control of an Autonomous Vehicle Based on Salp Swarm Algorithm," 2023, vol. 030043, no. March, https://doi.org/10.1063/5.0120403.

[19] G. H. S. Elias, A. Al-Moadhen, and H. Kamil, "Optimizing the PID controller to control the longitudinal motion of autonomous vehicles," in AIP Conference Proceedings, 2023, vol. 2591, no. 1, p. 040045, https://doi.org/10.1063/5.0120396.

[20] G. Hadi, A. Al-moadhen, and H. G. Kamil, "Optimization of Path Tracking of Self-Acting Mobile Robotic System," University of Kerbala, 2022.