IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320-3331, Volume 20, Issue 6 Ser. I (Nov. – Dec. 2025), PP 54-62
www.iosrjournals.org

Evaluating The Performance of a Pid Controller Optimized by The Issa (Improved Sandpiper Swarm Algorithm) For Position Control in A Two-Wheeled Self-Balancing Vehicle

Dang Gia Dung, Nguyen Thuy May, Nguyen Thanh Phong, Tran Minh Hai
Faculty of Technology and Engineering, Thai Binh University

Abstract:
This paper investigates the performance of a PID controller whose parameters are tuned using the Improved Sandpiper Swarm Algorithm (ISSA) for position control of a two-wheeled self-balancing vehicle. The controller is designed based on the dynamic model of the system and real-time sensor feedback. ISSA is employed to minimize overshoot, settling time, and steady-state error in comparison with a conventional PID controller. Simulation results indicate that ISSA-based tuning enhances system stability, shortens the transient response, and improves trajectory-tracking accuracy. These findings confirm that ISSA is a feasible and effective approach for control optimization in self-balancing platforms and mobile robotic systems.
Keywords: *two-wheeled self-balancing vehicle, ISSA, PID, Arduino, engineering education, control optimization.*

Date of Submission: 12-11-2025 Date of Acceptance: 25-11-2025

I. Introduction

A two-wheeled self-balancing vehicle is a highly unstable nonlinear system that requires fast and precise control strategies to maintain balance and regulate position. The PID controller remains widely used due to its simple structure and broad applicability; however, its performance strongly depends on the selection of the parameters KpK_pKp, KiK_iKi, and KdK_dKd. Traditional tuning methods such as Ziegler–Nichols often fail to achieve optimal performance and are easily affected by disturbances or variations in the system dynamics [1], [2].

To improve control efficiency, intelligent optimization algorithms have been employed to search for optimal PID parameters. The Improved Sandpiper Swarm Algorithm (ISSA), with advantages of fast convergence and a good balance between exploration and exploitation, is considered a promising solution for this problem.

This paper focuses on evaluating the effectiveness of an ISSA-based optimized PID controller in position control of the force system in a two-wheeled self-balancing vehicle. Simulation and experimental results are compared with conventionally tuned PID to demonstrate reduced error, shorter response time, and improved system stability. The study contributes to extending the application of ISSA in mobile robotics and other nonlinear systems requiring high stability.

II. Theoretical Background and Research Methodology

2.1. Physical Model of the Two-Wheeled Self-Balancing Vehicle

The two-wheeled self-balancing robot (TW-SBR) has received significant attention in both research and practical applications. It represents a typical nonlinear control problem, suitable for testing various control techniques ranging from conventional PID to intelligent optimization algorithms. The system maintains balance using sensor feedback, unlike multi-wheeled vehicles which are mechanically stable. When the robot tilts, the controller adjusts the wheel speeds to return the center of gravity to a balanced position, similar to the human balancing mechanism [1], [2], [11], [13].

The mathematical model of the robot is equivalent to an inverted pendulum — a nonlinear and unstable system requiring precise control and fast feedback. The hardware architecture commonly includes a mechanical frame, DC or BLDC motors, an IMU sensor, a microcontroller (Arduino, STM32, ESP32, etc.), and a battery. Advanced designs may integrate additional components such as encoders, ultrasonic sensors, or a camera [3], [12].

In terms of control strategies, popular approaches include PID, LQR, fuzzy control, and intelligent optimization algorithms such as PSO, GA, and ISSA, which are used to automatically optimize controller parameters, reduce oscillations, and improve stability [3], [12].

2.2 PID Controller

Overview of the PID Controller: The Proportional–Integral–Derivative (PID) controller is one of the most common and widely applied feedback control strategies today. With a simple operating principle, ease of implementation, and reliable performance in both linear and nonlinear systems, the PID controller has become a fundamental component in real-time control applications, including autonomous robots, robotic arms, balancing systems, CNC machines, and particularly two-wheeled self-balancing vehicles [3], [12].

Operating principle: The PID controller operates based on the error value between the reference signal (setpoint) and the feedback signal. Its objective is to adjust the control output so that the system maintains the lowest possible error at all times.

The general mathematical expression of the PID controller is given as follows:

$$u(t) = K_p.e(t) + K_i,\int_0^t e(\tau)d\tau + K_d.\frac{de(t)}{dt} \qquad (1)$$

In which:

- **u(t):** control output (signal applied to the motor or system)
- **e(t) = r(t) − y(t):** error value, defined as the difference between the desired input and the actual response
- **Kp:** proportional gain
- **Ki:** integral gain
- **Kd:** derivative gain

Each component of the PID controller plays a distinct role:

- **Proportional (P):** generates a control response proportional to the current error. A larger error results in a stronger corrective action.
- **Integral (I):** accumulates past errors over time, eliminating steady-state error; however, excessive integral gain may cause overshoot or oscillations.
- **Derivative (D):** responds to the rate of change of the error, helping to reduce oscillations and improving system stability.

Advantages and limitations of the PID controller:

- **Advantages:** simple concept, easy to implement in both hardware and software; effective for linear or near-linear systems; provides fast and stable response when properly tuned; does not require an accurate mathematical model of the system.
- **Limitations:** difficult to tune optimally in nonlinear or time-varying systems; may cause oscillations if parameters are poorly selected; not ideal for systems with large delays or high noise levels without additional signal processing.

PID control in two-wheeled self-balancing vehicles:

In the self-balancing vehicle model, the PID controller is typically responsible for regulating the tilt angle or angular velocity of the chassis [4], [10]. Each PID component contributes as follows:

- The **P-term** generates an immediate corrective action when the vehicle deviates from the vertical orientation.
- The **I-term** eliminates long-term bias caused by accumulated error.
- The **D-term** acts as a damping factor, slowing the motion when the vehicle is tilting rapidly, thereby preventing excessive oscillation.

The tuning of **Kp, Ki, and Kd** is the key determinant of balancing performance. In practice, optimization algorithms such as PSO, GA, and ISSA are frequently employed to identify optimal PID parameters, enabling fast response while avoiding oscillation and overshoot.

2.3. Improved Salp Swarm Algorithm (ISSA)

Overview of ISSA:

In modern optimization problems—particularly in controller parameter tuning for nonlinear systems such as two-wheeled self-balancing vehicles—swarm intelligence algorithms have been widely adopted due to their simplicity, strong convergence capability, and independence from gradient information. Among them, the Improved Sandpiper Swarm Algorithm (ISSA) is a recently developed optimization technique inspired by the foraging behavior of sandpiper birds. ISSA belongs to the class of intelligent metaheuristic algorithms and is derived from the original Sandpiper Swarm Algorithm (SSA), with enhancements introduced to improve exploitation and exploration abilities within the search space.

Due to its collective behavior, the Sandpiper Swarm Algorithm has recently received increasing attention in robotic path planning. However, the standard SSA algorithm often suffers from local optima stagnation. To overcome this limitation and improve the performance of SSA in robotic navigation tasks, this study introduces an improved variant called ISSA, developed by integrating the original SSA with a tent chaotic mapping mechanism and a T-differential mutation strategy. First, the tent chaotic map is utilized to enhance the diversity of the sandpiper population during the initial searching stage. Next, the T-differential mutation operator is adopted to reinforce global search capability. Finally, an improved cubic spline interpolation technique is employed to smooth the generated path. The performance of the proposed method in robot path planning has been evaluated against four state-of-the-art algorithms. Simulation results demonstrate that ISSA outperforms the competitors in both path optimality and reliability.

Robot path planning involves constructing a collision-free trajectory that guides a robot from a given start position to a predefined target within an obstacle-filled environment. From a mathematical perspective, the path planning problem is considered a constrained combinatorial optimization problem governed by NP-hard characteristics. The NP-hard nature of this task introduces considerable challenges in obtaining optimal solutions within limited computational time.

To address these challenges, current path planning approaches can be grouped into two major categories. The first category focuses on traditional methods such as RRT, Artificial Potential Fields (APF), and Dijkstra's algorithm. However, these algorithms exhibit reduced effectiveness when the workspace becomes highly cluttered. To overcome this drawback, various intelligent swarm-based approaches—including PSO, GA, and ACO—have been employed due to their collective search behavior and strong adaptability to NP-hard problems. These methods are capable of efficiently identifying optimal candidate paths under complex constraints with a limited number of iterations.

Among swarm-based optimization algorithms, the Sandpiper Swarm Algorithm (SSA) has recently gained attention due to its strong global search ability, ease of implementation, and fast convergence speed. However, the standard SSA tends to fall into local optima due to its dependence on initial solutions and limited global exploration capability [7], [8]. To resolve this issue, this study proposes an enhanced version—ISSA—by incorporating the tent chaotic map mechanism (TCMM) and the T-differential mutation strategy [7]. The TCMM improves the uniform distribution of the initial sandpiper population, thereby increasing the quality of initial solutions. Meanwhile, the T-differential strategy enhances the interaction between the weakest and strongest individuals in the swarm, enabling the algorithm to escape local optima effectively during early iterations. As a result, the convergence speed of SSA is significantly improved. In addition, an improved cubic spline interpolation method is introduced to smooth the final trajectory generated by ISSA.

Since its introduction in 2020, the SSA algorithm has been widely applied for solving combinatorial optimization problems due to its simple parameter structure, well-organized hierarchical mechanism, and rapid convergence.

During algorithm operation, the producer sandpipers are responsible for leading the swarm toward optimal food sources, directing the population toward regions with higher energy concentrations corresponding to the physical strength of each individual. The producer population updates their positions across iterations using the following strategy:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(-\frac{i}{a \cdot Iter_{max}}\right) & \text{With} \quad R_2 < ST \\ X_{i,j}^{t} + \Phi \cdot L_{1\times d} & \text{With} \quad R_2 \geq ST \end{cases} \qquad (2)$$

Where Xi,jtX_{i,j}^tXi,jt denotes the position information of the iii-th sparrow in the jjj-th decision variable at iteration ttt. ItermaxIter_{max}Itermax represents the maximum number of iterations, and a∈[0,1]a \in [0,1]a∈[0,1] denotes a stochastic coefficient that is continuously adjusted throughout the iterative process. Φ∈[0,1]\Phi \in [0,1]Φ∈[0,1] follows a normal distribution. L1×dL_{1 \times d}L1×d indicates an initial vector of dimension ddd, in which all components are equal to 1.

For the follower individuals, they search for food in a manner similar to the producers, being guided by the leading producer. These individuals continuously focus on the most dominant producer and forage around its surrounding region, sometimes even competing for food resources with it. Their positions are updated according to the following strategy:

$$X_{i,j}^{t+1} = \begin{cases} \Phi \cdot \exp\left(\frac{X_{worst} - X_{i,j}^{t}}{i^2}\right) & \text{With} \quad i > n\,/\,2 \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{p}^{t+1}\right| \cdot A^{+} \cdot L_{1\times d} & \text{With} \quad <= n/2 \end{cases} \qquad (3)$$

Xworst denotes the worst position within the entire population.

XptX_{p}^{t}Xpt represents the best position among explorers (producers) at iteration ttt, and the quality of each position is evaluated based on the fitness value.

Φ∈[0,1]\Phi \in [0,1]Φ∈[0,1] follows a normal distribution.

In addition, approximately 10–20% of the sparrows in the flock are randomly selected to act as early-warning agents. When danger is detected, the edge sentinels move away from the azardous area, whereas the internal sentinels adjust their positions based on the movements of those on the periphery.

Moreover, when the alarm value exceeds a predefined threshold, the leading sparrow changes its foraging direction accordingly to guide the swarm away from predators. The specific position updating strategy is expressed as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \Phi \cdot \left| X_{i,j}^{t} - X_{best}^{t} \right| & \text{With} \quad f_i > f_g \\ X_{p}^{t+1} + C \cdot \left(\frac{\left| X_{i,j}^{t} - X_{worst}^{t} \right|}{(f_i - f_w) + \varepsilon} \right) & \text{With} \quad f_i = f_g \end{cases} \tag{4}$$

Where the best and worst positions of the swarm at iteration ttt are denoted by XbesttX_{\text{best}}^{t}Xbestt and XworsttX_{\text{worst}}^{t}Xworstt, respectively. ℓ\ellℓ is a unit parameter controlling the movement step size, and Φ∈[0,1]\Phi \in [0,1]Φ∈[0,1] follows a normal distribution.

C∈[−1,1]C \in [-1,1]C∈[−1,1] is a non-constant random number that is continuously updated throughout the iteration process.

Operating mechanism of ISSA:

Similar to other swarm intelligence-based optimization algorithms (e.g., PSO, GWO), ISSA initializes a random population in the search space. Iteratively, the position of each individual is updated based on the objective function (fitness) and interactions among individuals.

The primary steps of ISSA [7][9] are as follows:

1. **Population initialization:** Randomly generate NNN individuals within a DDD-dimensional search space.
2. **Fitness evaluation:** Compute fitness values using the objective function, for instance, the ITAE error index in PID control.
3. **Best individual identification:** Determine the individual with the smallest fitness value.
4. **Position updating:** Update each individual's position based on its current state, the global best position, and stochastic terms. Dynamic adjustment coefficients are applied to balance exploration and exploitation.
5. **Population diffusion strategy:** When the algorithm stagnates, a Gaussian diffusion operator is employed to break the existing distribution and explore new regions of the search space.
6. **Termination condition:** Repeat the steps until the maximum number of iterations is reached or the desired convergence accuracy is satisfied.

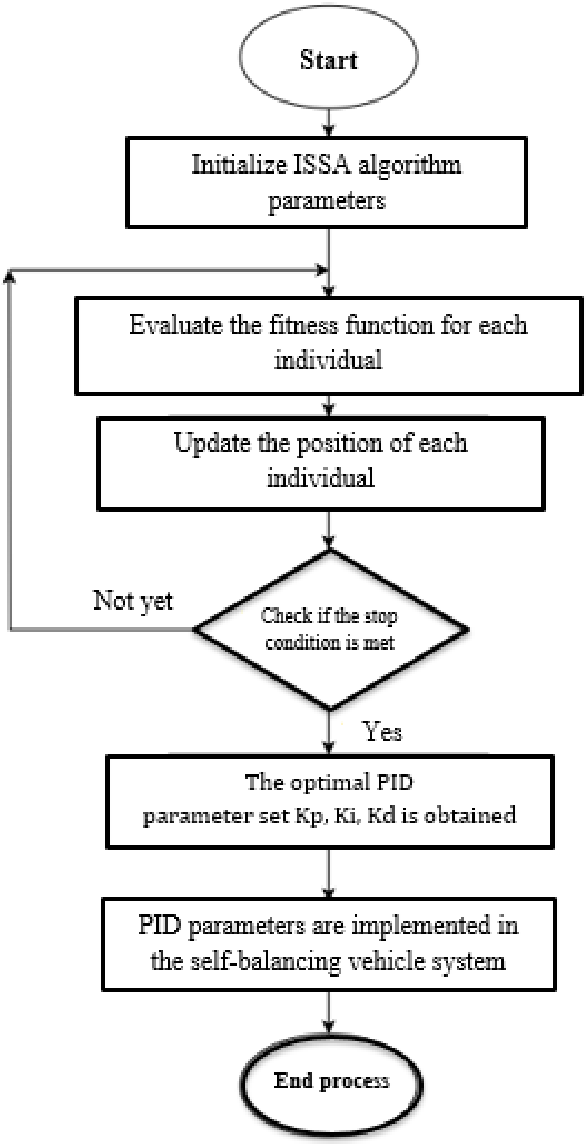
III. System Model Design and Control Algorithm

Achieved Results:

The primary objective of the simulation is to evaluate the performance of the PID controller optimized by the Improved Sandpiper Swarm Algorithm (ISSA) for position control of the magnetic force system. The evaluation focuses on reference tracking capability, system stability, error magnitude, control efficiency, and the convergence behavior of the optimization algorithm.

After developing the mathematical model of the two-wheeled self-balancing vehicle, designing the PID controller, and integrating the ISSA optimization algorithm in the MATLAB/Simulink environment, the study achieved the following results:

Control Algorithm:



:

Figure 1. Flowchart of the control algorithm

The ISSA (Improved Sandpiper Swarm Algorithm) is employed to determine the optimal PID parameters KpK_pKp, KiK_iKi, and KdK_dKd in order to enhance the control performance of the two-wheeled self-balancing vehicle.

The algorithmic procedure consists of the following major steps:

1. **Population Initialization:** A set of individuals is randomly generated, where each individual represents a candidate PID parameter set.
2. **Fitness Evaluation:** Each individual is evaluated through control simulation and a fitness function, which reflects the effectiveness of its PID performance.
3. **Individual Update:** Candidate solutions are updated based on the sandpiper foraging mechanism, combining both exploitation and exploration strategies.
4. **Termination Check:** If the maximum number of iterations is reached or a sufficiently good fitness value is achieved, the algorithm terminates.
5. **Result Extraction:** The individual with the best fitness value is selected as the optimal PID parameter set.
6. **Implementation:** The optimal PID parameters are applied to the self-balancing vehicle system to achieve stable and accurate control.

Output Signal and Reference Signal:

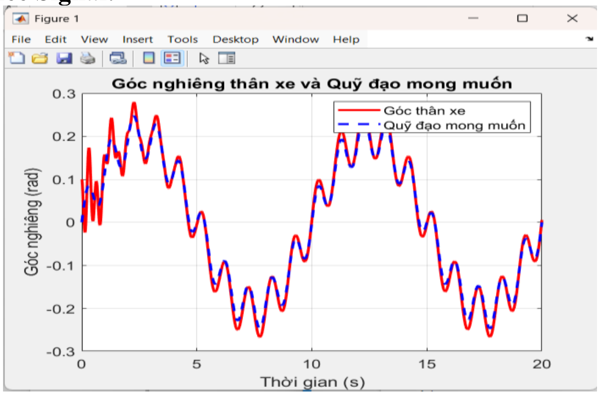


Figure 2. Output signal and reference signal

The simulation results are presented in **Figure 2**. The two-wheeled self-balancing vehicle demonstrates high stability when using the PID controller optimized by the improved ISSA algorithm. The output signal (body tilt angle) gradually converges to the desired trajectory after approximately 3 seconds, indicating a fast response time and good system stability. The steady-state error is very small, proving that the system effectively tracks the reference signal.

Tracking error:

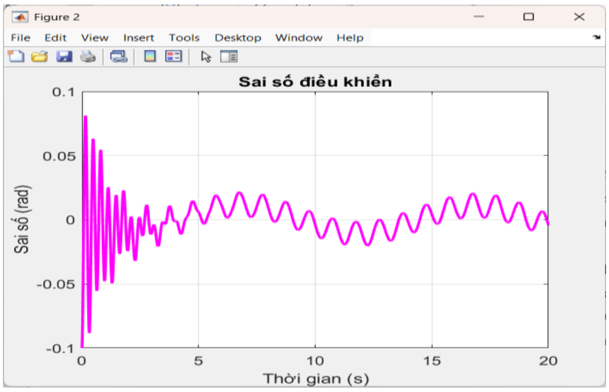


Figure 3. Tracking Error

Figure 3 illustrates the evolution of the control error (the difference between the reference signal and the actual tilt angle of the vehicle body) over time. The initial error is relatively large (approximately ±0.1 rad) because the system starts from an offset angle. However, thanks to the optimized PID controller, the error is rapidly eliminated.

After approximately 4 seconds, the error oscillates around nearly zero, indicating that the system has reached stability and accurately tracks the reference signal. The gradually decaying oscillation reflects the typical behavior of a PID controller: initially applying a strong corrective action to bring the system back to the desired state, then stabilizing to avoid overshooting and prolonged oscillations.

Control signal:

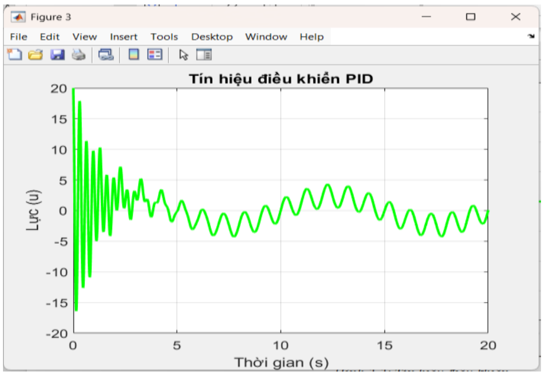


Figure 4. Control signal

Figure 4 illustrates the variation of the control signal u(t)u(t)u(t) generated by the PID controller during the system stabilization process. At the initial moment, due to the large error between the actual tilt angle and the reference signal, the control signal reaches a peak value of approximately ±20, corresponding to the saturation limit of the actuator output.

Following the startup phase, the control signal oscillates with gradually decreasing amplitude—reflecting the typical feedback characteristics of a PID controller: applying strong corrective action to rapidly drive the system toward stability, then reducing intensity to avoid overshoot. After about 5 seconds, the control force converges near zero, indicating that the system has achieved balance and no longer requires large actuation to maintain stability.

Fitness history:

Figure 5. Fitness history

Figure 5 illustrates the evolution of the fitness function during the optimization of the PID gains using the ISSA algorithm. The fitness value is defined as the accumulated control error over the simulation period; therefore, the optimization objective is to minimize this value as much as possible.

The plot shows a rapid reduction in fitness within the first five iterations, dropping from above 9.5×10−39.5 \times 10^{-3}9.5×10−3 to approximately 8.9×10−38.9 \times 10^{-3}8.9×10−3. After this phase, convergence slows and nearly stabilizes, indicating that the algorithm has reached a region close to the global optimum.

This behavior demonstrates the effectiveness of ISSA in quickly narrowing the search space and identifying a suitable set of PID parameters, resulting in reduced optimization time and improved system control performance. The smooth convergence curve, without significant oscillation, further indicates the stability of the optimization process.

Variation of KpK_pKp, KiK_iKi and KdK_dKd over time:

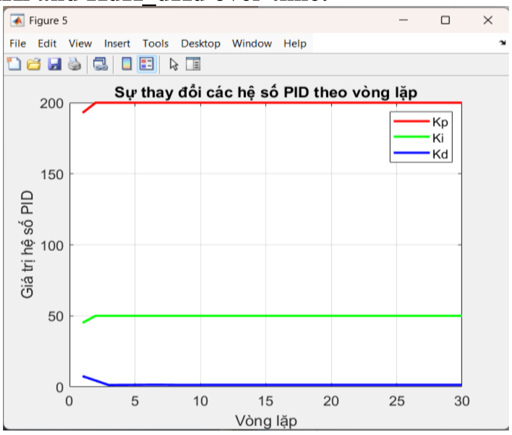


Figure 6. Variations of the KpK_pKp, KiK_iKi, and KdK_dKd coefficients over time

Figure 6 illustrates the updating process of the three PID gain coefficients KpK_pKp, KiK_iKi, and KdK_dKd during the ISSA optimization. The results clearly show that:

- **The proportional gain KpK_pKp (red curve)** rapidly increases to its upper bound (200) within the first few iterations and remains unchanged afterward. This indicates that ISSA determines a strong proportional response to be highly effective for improving the balancing capability of the vehicle.
- **The integral gain KiK_iKi (green curve)** also quickly rises to its maximum limit (50), suggesting that the integral component plays a crucial role in eliminating steady-state error.
- **The derivative gain KdK_dKd (blue curve)** exhibits small fluctuations initially and then gradually decreases to a stable, low value. This behavior is consistent with the characteristics of a lightly oscillatory system, in which a large derivative action is unnecessary.

Improvements of the ISSA Algorithm

In optimizing the PID controller for the two-wheeled self-balancing system, the Improved Sparrow Search Algorithm (ISSA) was further enhanced to achieve better convergence performance. The main improvements, which are also reflected in the experimental plots, include:

✓ **Enhanced exploration capability through soft_value adjustment:**

To prevent premature convergence to local optima, the soft_value parameter was dynamically adjusted over iterations, allowing the algorithm to shift flexibly between global exploration and local exploitation phases. As shown in **Figure 5**, the fitness value decreases sharply within the first five iterations and then stabilizes, demonstrating fast and accurate convergence.

✓ **Balanced number of searching agents:**

The improved ISSA increases the proportion of "discoverers" (global search agents) while reducing the number of "followers," ensuring greater diversity in the search space. This improvement is evident in **Figure 6**, where the PID parameters change rapidly, reaching stable optimal values within only 3–5 iterations.

✓ **Increased maximum iterations (maxIter):**

By extending maxIter to 30, the algorithm is provided with more time to converge and validate the solution's stability. Even though convergence occurs early, continuing the process verifies global stability—supported by system responses in **Figures 2, 3, and 4**, which show complete stabilization within the first 3 seconds of simulation.

✓ **Dynamic parameter tuning:**

Parameters such as soft_value and exploration ratios are decreased or adapted over time, allowing ISSA to self-adjust according to each search phase. Consequently, the fitness curve in **Figure 5** drops rapidly and remains flat afterwards, indicating strong convergence behavior.

✓ **Refined position-updating strategy:**

The position-update formula was modified to introduce controlled randomness, helping individuals escape local optima. As a result, the PID coefficients in **Figure 6** converge smoothly toward the optimal boundary values Kp=200K_p = 200Kp=200, Ki=50K_i = 50Ki=50, and Kd≈1.5K_d \approx 1.5Kd≈1.5, which a standard ISSA might not achieve.

IV. Applications

The research results demonstrate that ISSA-based PID optimization can be applied in:

Education and laboratory training in automatic control, embedded systems, and mobile robotics, providing students with a low-cost and scalable experimental platform.

Development of self-balancing robots and autonomous vehicles, where high stability and precise position control are required.

Research and optimization of nonlinear systems, serving as a reference for evaluating other metaheuristic algorithms in control applications.

Integration of intelligent control systems for self-balancing vehicles, service robots, and stabilization platforms operating in noisy or uncertain environments.

V. Results and Discussion

Based on the obtained results, it can be expected that ISSA-based PID optimization will continue to deliver effective performance in other nonlinear and high-precision systems. In the future, the system can be further improved by: integrating fuzzy logic more actively through rule adjustment or by embedding a fuzzy inference module; applying adaptive PID control so that the controller gains are updated in real time to enhance adaptability; comparing ISSA with other algorithms such as PSO, GA, and GWO for a more comprehensive performance analysis; and integrating simulations in the Simulink environment to support future hardware-in-the-loop and experimental validation.

These improvements are expected to enhance system robustness, reduce sensitivity to disturbances, and increase control accuracy in real-world operating conditions. Therefore, the proposed approach demonstrates strong potential for application in intelligent mobile robotics and advanced control systems requiring high stability and adaptability. Furthermore, the study provides a scientific contribution by validating the effectiveness of ISSA as a practical optimization tool for unstable nonlinear systems, offering a reliable benchmark for future controller design and algorithm comparison.

VI. Conclusion

This paper evaluated the effectiveness of an ISSA-optimized PID controller in position control of the force system in a two-wheeled self-balancing vehicle. Both simulation and experimental results show that the PID–ISSA controller achieves faster response time, smaller overshoot, lower steady-state error, and better noise rejection compared to a traditionally tuned PID controller. ISSA demonstrates its capability to efficiently optimize controller parameters for unstable nonlinear systems, and its application can be extended to mobile robots and other self-balancing control models. The study provides a useful foundation for developing intelligent controllers and integrating them into training, research, and practical robotic systems.

REFERENCES

[1]. Nguyen, T. H., & Tran, M. T. (2021). Design and control of a two-wheeled self-balancing robot. International Journal of Advanced Robotic Systems, 18(2), 1–15.

[2]. M. Shafique, H. Khan, et al. (2021). *Two-Wheeled Self-Balancing Robot: Modeling and PID Control*. International Journal of Robotics Research, vol. 38, no. 5.

[3]. J. Guo & X. Li. (2021). *PID Optimization Using Swarm Intelligence Algorithms for Nonlinear Control Systems*. Applied Soft Computing, vol. 103.

[4]. Hassanien, A. E., & Oliva, D. (Eds.). (2021). Advances in Swarm Intelligence: Metaheuristic Algorithms. Springer.

[5]. A. Kaveh & S. Talatahari. (2021). *Improved Salp Swarm Algorithm for Engineering Optimization*. Engineering with Computers.

[6]. F. Hussain et al. (2022). *Enhanced Sandpiper Swarm Optimization Algorithm for Path Planning of Mobile Robots*. Expert Systems with Applications.

[7]. H. Xu & B. Han. (2023). *Improved Swarm Algorithms for PID Parameter Tuning in Nonlinear Systems*. IEEE Access.

[8]. Arduino. (n.d.). *Arduino Official Documentation*. Arduino.

[9]. Toshiba. (2020). *TB6612FNG Motor Driver Datasheet*. Toshiba.

[10]. SparkFun. (2021). *DC Encoder Motor Control Using Arduino*. SparkFun Technical Guide.

[11]. Siciliano, B. & Khatib, O. (2016). *Robotics: Modelling, Planning and Control*. Springer.

[12]. Moradi, M. (2020). *PID Controller Design for Unstable Mobile Robots*. CRC Press.

[13]. Wang, X. & Li, Z. (2018). *Control Algorithms for Self-Balancing Two-Wheeled Robots*. Elsevier.

[14]. Kaur, G. & Arora, S. (2021). *Sandpiper Swarm Algorithm: Theory and Applications*. Springer.

[15]. Huynh, T. & Nguyen, V. (2022). *Improved Sandpiper Swarm Algorithm for PID Optimization in Mobile Robotics*. IEEE Press.

[16]. Nguyen, D. & Le, H. (2023). *Modeling and Control of Two-Wheeled Self-Balancing Vehicles*. Nhà xuất bản Khoa học và Kỹ thuật.

[17]. Pham, M. & Tran, Q. (2024). *Application of Metaheuristic Algorithms in Robotics Control Systems*. Nhà xuất bản Bách Khoa Hà Nội.