Iranian Journal of Fuzzy Systems

Volume 23, Number 3, (2026), pp. 51-68

Original paper

University of Sistan and Baluchestan

A hybrid ISSO-FACS approach for optimal sliding mode control of an inverted pendulum

J. H. Choe [1], G. Choe [2], S. I. Kwak [3], H. G. Chae [4] and H. U. Kong [5]

[1,2,3,5] *Faculty of Artificial Intelligence,* ***Kim Il Sung*** *University, Pyongyang, Democratic People's Republic of Korea*

[1] *School of International Cultural Exchange, Shanghai University of Finance and Economics, Shanghai, People's Republic of China*

[4] *Faculty of Information Science, Pyongyang University of Computer Science, Pyongyang, Democratic People's Republic of Korea*

[5] *School of Business Administration, Northeastern University, Shenyang, People's Republic of China*

cjh920321@163.com, gw.choe@ryongnamsan.edu.kp, si.kwak@ryongnamsan.edu.kp, chae123@163.com, khu1210@163.com

Abstract

This paper proposes a hybrid approach that integrates an Improved Shark Smell Optimization (ISSO) algorithm with a Fuzzy Ant Colony System (FACS) to enhance the parameter tuning of a fuzzy hierarchical controller for the inverted pendulum and cart system. The standard Shark Smell Optimization (SSO) algorithm suffers from premature convergence and limited exploration capability. The proposed ISSO addresses these limitations by introducing dynamic adaptive coefficients for the gradient and inertia phases, along with a linearly decreasing velocity limit factor. This enhanced global search is then synergistically combined with FACS, which performs fuzzy-guided local exploitation in promising regions of the parameter space. The ISSO-FACS hybrid is applied to determine the optimal sliding surface parameters and scaling factors of a Fuzzy Hierarchical Swing-up and Sliding Controller (FHSSC). Simulation results demonstrate that the proposed approach significantly outperforms the previous FACS method, reducing the total stabilization error by 11.59%, the pendulum angle error by 15.23%, and the cart position error by 26.72%, while achieving 1.27 times faster settling time. The proposed controller also exhibits superior robustness under perturbed initial conditions where the previous method fails, and demonstrates enhanced disturbance rejection capabilities. Convergence analysis confirms the stability properties of the hybrid approach, and computational complexity analysis validates its suitability for offline controller design.

Keywords: Shark smell optimization, fuzzy ant colony system, inverted pendulum, stabilization, hybrid algorithm.

1 Introduction

The field of computational intelligence has witnessed remarkable progress through the development of nature-inspired metaheuristic algorithms, with Shark Smell Optimization (SSO) emerging as a significant contribution to this evolving landscape. Originating from the sophisticated foraging behavior of sharks in oceanic environments, SSO algorithms simulate the biological processes through which sharks detect and track prey using their extraordinary olfactory capabilities over considerable distances [1]. The initial formulation of SSO in 2014 established a novel optimization framework that effectively balanced computational efficiency with biological inspiration, creating a robust methodology that has since evolved into multiple specialized variants [4]. This foundational work demonstrated the algorithm's potential through successful applications in power systems optimization, particularly in economic load dispatch problems, showcasing its practical utility for complex engineering challenges [2].

Corresponding Author: J. H. Choe
Received: November 2025; Revised: April 2026; Accepted: May 2026.
https://doi.org/10.22111/ijfs.2026.53975.9565

The evolutionary progression of SSO algorithms has been characterized by continuous refinement and diversification to address emerging optimization challenges. The standard SSO algorithm, while demonstrating notable strengths in unimodal optimization, revealed limitations in handling complex multimodal landscapes, prompting researchers to develop enhanced variants [3]. These developments include Binary SSO for discrete optimization domains, Chaotic SSO for improved global search capabilities, Hybrid SSO models integrating complementary optimization techniques, and Multi-objective SSO for handling conflicting objectives in engineering design [5, 6]. Each variant has contributed to expanding SSO's applicability while addressing specific computational limitations observed in practical implementations.

The integration of optimization algorithms into control systems represents a particularly successful application domain that demonstrates their practical significance. Advanced control strategies, including fuzzy logic control and sliding mode control, have been effectively employed in various applications such as mobile robot navigation [28] and autonomous vehicle trajectory planning [29]. These systems require sophisticated optimization capabilities to balance multiple performance objectives while considering operational constraints and uncertainties. Various metaheuristic algorithms have been effectively employed in these systems for controller parameter tuning, optimal trajectory planning, and robust control design. The algorithms' ability to handle non-convex, nonlinear problems with multiple constraints makes them particularly suitable for complex control systems where traditional optimization methods often encounter challenges with computational efficiency and solution quality [7].

The standard Shark Smell Optimization algorithm established the fundamental principles that subsequent variants would build upon, drawing direct inspiration from the natural foraging behavior of sharks in locating prey through scent gradient tracking [1]. This initial formulation implemented a population-based approach where candidate solutions, conceptualized as sharks, navigate the search space using two primary movements: velocity-based progression toward areas of stronger scent intensity and rotational movements for exploring new regions to maintain search diversity. The algorithm's biological foundation provided an intuitive yet powerful optimization methodology that quickly gained recognition in computational intelligence circles [8].

The advantages of standard SSO are particularly evident in its architectural simplicity and computational efficiency. The algorithm requires fewer control parameters compared to many contemporary metaheuristics, significantly reducing implementation complexity and parameter tuning efforts [9]. Its strong exploitation capability, derived from the gradient-following mechanism, enables rapid convergence in unimodal optimization landscapes. Furthermore, the inherent balance between directed and random movements provides a natural exploration-exploitation tradeoff without requiring sophisticated coordination mechanisms [1]. These characteristics made standard SSO particularly suitable for medium-scale optimization problems where computational resources are limited and implementation simplicity is valued.

However, the standard SSO algorithm exhibits several disadvantages that limit its effectiveness in complex optimization scenarios. The algorithm demonstrates significant susceptibility to premature convergence when applied to multimodal problems with numerous local optima, primarily due to its strong exploitation bias that can cause rapid population diversity loss [10]. Parameter sensitivity represents another critical challenge, where algorithmic performance is considerably influenced by initial parameter settings, particularly the velocity factor and population size configurations [3]. Additionally, the algorithm shows limited scalability in high-dimensional search spaces due to reduced exploration capability and increased computational resource requirements. These limitations motivated the research community to develop enhanced SSO variants with improved global search characteristics and better performance consistency across diverse problem types.

1.1 Related work

Binary Shark Smell Optimization emerged as a crucial adaptation to address the growing need for effective discrete optimization capabilities across various application domains. Developed primarily for problems requiring binary solution representations, BSSO introduced sophisticated transformation mechanisms to convert the continuous search process of standard SSO into discrete decision spaces [5]. This variant employed transfer functions, including S-shaped and V-shaped sigmoidal functions, to map continuous velocity values to probabilities of binary position changes. The development of BSSO significantly expanded SSO's applicability to domains such as feature selection, knapsack problems, and combinatorial optimization, where discrete solution encoding is essential for practical implementation [11].

The advantages of Binary SSO are particularly evident in specific application contexts, especially within feature selection systems for machine learning and data mining. The algorithm maintains the core convergence properties of standard SSO while effectively operating in discrete search spaces, providing a seamless transition between continuous and binary optimization paradigms [12]. BSSO demonstrates superior performance in feature selection tasks compared to traditional binary algorithms, effectively reducing feature dimensionality while maintaining or improving classification accuracy in pattern recognition systems. The transformation mechanism preserves the gradient-following behavior of

original SSO, enabling efficient navigation of binary landscapes through probabilistic position updates [5]. These characteristics make BSSO particularly valuable for data mining, pattern recognition, and other domains requiring discrete optimization capabilities.

Despite these advantages, Binary SSO introduces several disadvantages that impact its general applicability within complex systems. The incorporation of transfer functions increases computational overhead due to additional function evaluations during each iteration, which can be problematic in real-time systems with strict timing constraints [13]. Algorithm performance shows significant sensitivity to the selection of transfer function types, where inappropriate choices can lead to stagnation or random walking behavior in optimization processes. The binary transformation process also introduces challenges in parameter tuning, as optimal parameter settings often vary considerably across different problem instances and system characteristics [11]. Furthermore, the algorithm's effectiveness diminishes in continuous optimization domains, limiting its utility to specifically discrete problems and necessitating alternative variants for mixed-variable optimization scenarios in heterogeneous systems.

Chaotic Shark Smell Optimization represents a significant advancement in addressing the population diversity and premature convergence limitations observed in standard SSO implementations. This variant integrates chaotic maps and non-linear dynamics to enhance the algorithm's exploration capabilities and improve its ability to escape local optima in complex optimization landscapes [6]. By replacing conventional random number generators with chaotic sequences in critical algorithm parameters, CSSO introduces deterministic yet unpredictable patterns that significantly improve global search behavior. Various chaotic maps, including Logistic, Tent, and Sinusoidal maps, have been successfully incorporated to control parameters such as the velocity factor and random components in position updates, creating a more robust optimization framework [14].

The advantages of Chaotic SSO are clearly demonstrated through improved optimization performance in complex system applications, particularly in smart grid energy management systems. The integration of chaotic sequences significantly enhances population diversity throughout the search process, effectively reducing premature convergence probability in multimodal problems commonly encountered in power system optimization [15]. The algorithm exhibits adaptive search characteristics through dynamic parameter adjustments driven by chaotic dynamics, eliminating the need for static parameter settings that often require extensive tuning. CSSO demonstrates superior exploration capabilities compared to standard SSO, particularly in high-dimensional search spaces and problems with numerous local optima, making it highly suitable for complex engineering systems requiring robust optimization capabilities [6].

However, Chaotic SSO introduces several disadvantages that affect its practical implementation in real-world systems. The computational complexity increases substantially due to chaotic map evaluations and additional parameter control mechanisms, which can challenge systems with limited computational resources. Algorithm performance shows critical dependence on appropriate chaotic map selection, where unsuitable maps can degrade performance below standard SSO levels, requiring extensive preliminary analysis for each new application system. The implementation complexity is considerably higher than standard SSO, requiring expertise in chaotic systems and non-linear dynamics for effective deployment [14]. Furthermore, parameter tuning becomes more challenging due to interactions between chaotic map parameters and standard SSO parameters, often necessitating extensive experimental analysis for optimal configuration in specific system environments.

Hybrid Shark Smell Optimization methods represent a strategic approach to overcoming individual algorithm limitations through synergistic combinations with other optimization techniques, creating more robust and efficient optimization frameworks suitable for complex systems. These hybrid models leverage complementary strengths to address specific SSO limitations while preserving its fundamental advantages. Notable hybrid approaches include SSO-PSO combinations that integrate social learning mechanisms from Particle Swarm Optimization, SSO-GA hybrids incorporating evolutionary operators from Genetic Algorithms, and SSO-SOS integrations employing symbiotic relationships from Symbiotic Organism Search. Each hybridization strategy has demonstrated enhanced performance across diverse problem types and system applications.

The advantages of Hybrid SSO methods are particularly valuable in complex system optimization, such as smart grid energy management systems where multiple constraints and objectives must be simultaneously addressed. These approaches achieve superior balance between exploration and exploitation through complementary mechanism integration, significantly improving convergence reliability in dynamic system environments [17]. Hybrid algorithms typically demonstrate robust performance across diverse problem characteristics, reducing application-specific parameter tuning requirements that often challenge system implementers. The synergistic combinations often yield performance exceeding individual component algorithms, particularly in complex, high-dimensional optimization landscapes commonly found in real-world systems. Furthermore, hybridization enables effective handling of specific system constraints and characteristics through customized operator integration, expanding SSO's applicability to specialized domains requiring unique solution strategies.

Despite these advantages, Hybrid SSO methods introduce several disadvantages that impact their practical im-

plementation in operational systems. The increased parameter complexity requires sophisticated tuning strategies to optimize interactions between algorithmic components, demanding significant expertise and computational resources for system calibration [18]. Implementation overhead rises substantially due to the need to integrate and coordinate multiple algorithm frameworks and operators, increasing system development time and maintenance requirements. Computational costs are typically higher than individual algorithm implementations, demanding greater resources for equivalent problem sizes, which can be prohibitive in resource-constrained systems. Additionally, achieving optimal balance between combined algorithms presents significant challenges, often requiring extensive experimental analysis and problem-specific customization that can limit general applicability across different system domains.

Multi-objective Shark Smell Optimization extensions address the critical need for effective Pareto-based optimization in problems with conflicting objectives, representing a natural evolution from single-objective SSO variants [19]. These developments incorporate specialized mechanisms for handling multiple optimization criteria simultaneously, including non-dominated sorting techniques, crowding distance measures, and elite preservation strategies to maintain diverse solution sets across Pareto fronts. Recent advancements have extended these capabilities to many-objective optimization problems through reference-point approaches and novel diversity preservation mechanisms, significantly enhancing SSO's applicability to complex decision-making scenarios [20].

The advantages of Multi-objective SSO are particularly valuable in real-world engineering systems where multiple competing objectives must be balanced. The algorithm effectively handles multiple conflicting objectives without requiring scalarization or weight assignment, providing decision-makers with comprehensive solution sets that represent optimal trade-offs [16]. MOSSO demonstrates robust diversity maintenance along Pareto fronts, enabling balanced exploration of objective tradeoffs in complex system design and optimization. The approach scales effectively to problems with numerous objectives through advanced archiving and selection strategies, maintaining computational efficiency while extending SSO's capabilities to multi-objective domains [19]. These characteristics make MOSSO particularly suitable for complex decision-making scenarios in engineering design, resource allocation, and system optimization where multiple performance criteria must be simultaneously considered.

However, Multi-objective SSO introduces several disadvantages that impact its practical implementation in operational systems. The algorithmic overhead increases substantially with objective count, particularly for many-objective problems where dominance relationships become sparse and computational requirements escalate [21]. Maintaining population diversity across high-dimensional objective spaces presents significant challenges, often requiring specialized niching and archiving strategies that increase implementation complexity. Performance sensitivity to archive size and diversity parameters necessitates careful configuration, often requiring problem-specific tuning that can limit general applicability across different system domains [20]. Additionally, the computational intensity of non-dominated sorting and crowding distance calculations can become prohibitive for real-time applications or systems with limited computational resources, restricting MOSSO's deployment in time-sensitive operational environments.

1.2 Research gap and contribution

Despite numerous advancements in SSO variants, several research gaps remain. First, most hybrid approaches focus on combining SSO with other metaheuristics but lack intelligent parameter adaptation mechanisms. Second, the integration of fuzzy logic with SSO for dynamic parameter tuning is underexplored. Third, existing methods often fail to balance exploration and exploitation effectively in complex control systems like the inverted pendulum. Fourth, robustness under perturbed conditions is rarely addressed comprehensively.

A key challenge in hybridizing metaheuristics is the potential propagation of errors from one approximate solver to another. Our proposed approach mitigates this through a synergistic, sequential architecture. Instead of combining them in parallel, we leverage ISSO's enhanced global search to first identify a set of promising basins of attraction within the high-dimensional parameter space. This coarse global search phase effectively filters out poor solutions. Subsequently, the FACS, with its fuzzy-guided local search, is initialized within these high-quality regions to perform fine-grained exploitation. This hierarchical strategy prevents the propagation of errors, as the local search operates on seeds that have already been vetted by the global search mechanism. The fuzzy logic within FACS further refines the search path, intelligently balancing exploration and exploitation during the final tuning stage.

Our main contributions are:

1. **Novel Hybrid Algorithm**: We propose ISSO-FACS, which integrates improved SSO with fuzzy ant colony optimization for adaptive parameter tuning.

2. **Dynamic Parameter Adaptation**: Introduction of linearly decreasing velocity limits and adaptive gradient/inertia coefficients to prevent premature convergence.

3. **Fuzzy-Enhanced Search**: Incorporation of fuzzy rules to guide the ant colony search process, improving convergence in high-dimensional spaces.
4. **Application to Complex Control**: Successful implementation for FHSSC parameter optimization in the inverted pendulum system.
5. **Comprehensive Validation**: Extensive simulations showing superior performance in error reduction, settling time, and robustness compared to existing methods.

This paper is organized as follows. In Section 2, we present background information on standard SSO. Section 3 describes our Improved Shark Smell Optimization algorithm. Parameter determination using Fuzzy Ant Colony System is detailed in Section 4. Section 5 presents simulation results and analysis, including system dynamics, controller architecture, benchmark testing, robustness analysis, disturbance rejection, and computational complexity. Finally, conclusions and future work are discussed in Section 6.

2 Background information

The SSO algorithm mimics a shark's hunting process, which consists of moving towards a prey scent and performing local searches. The algorithm operates as follows [24].

Step 1: Initialization

The initialization starts with randomly generating shark positions in the search space. Each individual represents a possible solution. The initial population vector set X^1 is denoted by:

$$X^1 = \left[X_1^1, X_2^1, X_3^1, \ldots, X_{NP}^1\right], \tag{1}$$

where X_i^1 represents the i-th initial population of the initial position, and NP is the population size. Each position vector is:

$$X_i^{kmax} = \left[x_{i,1}^{kmax}, x_{i,2}^{kmax}, x_{i,3}^{kmax}, \ldots, x_{i,ND}^{kmax}\right], \tag{2}$$

where $x_{i,1}^k$ is the j-th element and ND is the dimension.

Step 2: Forward Movement

Each shark has an initial velocity V^1. The velocity vector is:

$$V^1 = \left[V_1^1, V_2^1, V_3^1, \ldots, V_{NP}^1\right]. \tag{3}$$

Each velocity vector has ND elements:

$$V_i^1 = [v_{i,1}^1, v_{i,2}^1, v_{i,3}^1, \ldots, v_{i,ND}^1], \quad i = \overline{1, NP}. \tag{4}$$

At iteration k, the velocity is updated by:

$$v_{i,j}^k = \eta_k \cdot R_1 \frac{\partial(OF)}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k \cdot R_2 \cdot v_{i,j}^{k-1}, \quad j = \overline{1, ND}, \quad i = \overline{1, NP}, \quad k = \overline{1, k_{\max}}, \tag{5}$$

where OF is the objective function, $R_1,\ R_2 \in (0, 1)$ are random numbers, η_k is the displacement probability, and α_k is the acceleration coefficient.

Velocity is limited by:

$$|v_{i,j}^k| = \min\left[\left|\eta_k \cdot R_1 \frac{\partial(OF)}{\partial x_j}\bigg|_{x_{i,j}^k} + \alpha_k \cdot R_2 \cdot v_{i,j}^{k-1}\right|,\ \left|\beta_k \cdot v_{i,j}^{k-1}\right|\right], \tag{6}$$

where β_k is the rate limiting factor.

The new position is:

$$Y_i^{k+1} = X_i^k + V_i^k \cdot \Delta t_k, \tag{7}$$

where Y_i^{k+1} is the new position of the shark population and Δt_k is the time interval in the k-th cycle.

Step 3: Rotational Movement (Local Search)

Sharks perform local search around the new position:

$$Z_i^{k+1,m} = Y_i^{k+1} + R_3 \cdot Y_i^{k+1}, \quad m = \overline{1, M}, \tag{8}$$

where $R_3 \in [-1, 1]$ random, and M is the number of local search points.

Step 4: Position Update

The next position is selected as the best among forward and rotational positions:

$$X_i^{k+1} = \arg\min \left\{ OF\left(Y_i^{k+1}\right), OF\left(Z_i^{k+1,1}\right), \ \ldots, OF\left(Z_i^{k+1,M}\right) \right\}. \tag{9}$$

This process repeats until $k_{\max}$.

3 Improved shark smell optimization algorithm

Standard SSO suffers from premature convergence and poor exploration. We introduce three improvements.

3.1 Improved velocity limiting factor

We replace the fixed β_k with a linearly decreasing factor:

$$\omega\beta_k = \left(\frac{V_{\max}}{V_{\min} \cdot NP}\right) \times \left(\omega_{start} - \frac{\omega_{start} - \omega_{end}}{iter_{\max}} \times iter_{current}\right), \tag{10}$$

where $\omega_{start}, \omega_{end} \in (0, 1)$, $iter_{\max}$ is total iterations, and $iter_{current}$ is current iteration. This formulation allows broader exploration in early iterations and finer exploitation in later iterations.

3.2 Adaptive gradient and inertia coefficients

We introduce linearly decreasing η_k and linearly increasing α_k:

$$\omega\eta_k = \omega_{start} - \frac{\omega_{start} - \omega_{end}}{iter_{\max}} \times iter_{current}, \tag{11}$$

$$\omega\alpha_k = \omega_{end} - \frac{\omega_{end} - \omega_{start}}{iter_{\max}} \times iter_{current}. \tag{12}$$

The velocity update becomes:

$$|v_{i,j}^k| = \min \left[\left| \omega\eta_k \cdot R_1 \frac{\partial(OF)}{\partial x_j}\bigg|_{x_{i,j}^k} + \omega\alpha_k \cdot R_2 \cdot v_{i,j}^{k-1} \right|, \ \left| \omega\beta_k \cdot v_{i,j}^{k-1} \right| \right]. \tag{13}$$

This adaptive mechanism ensures that the algorithm emphasizes global exploration in early stages (with larger gradient steps and smaller inertia) and transitions to local exploitation in later stages (with smaller gradient steps and larger inertia from previous velocity).

3.3 Algorithm pseudocode

Algorithm 1: Improved Shark Smell Optimization (ISSO)

1. Initialize population $\boldsymbol{X}$, velocities $\boldsymbol{V}$, parameters ω_{start}, ω_{end}, NP, ND, $iter_{max}$
2. Evaluate initial fitness
3. **For** $iter$=1 to $iter_{max}$
   - Update $\omega\eta$, $\omega\alpha$, $\omega\beta$ using Eqs. (10)-(12)
   - **For** each shark i

·Compute gradient ∇OF

·Update velocity using Eq. (13)

·Update position: $Y_i = X_i + V_i \cdot \Delta t$

·Perform local search to generate $Z_i^{1..M}$

·Select best position from $Y_i,\ Z_i^{1..M}$

- Update global best

4. **End For**

5. Return best solution

3.4 Convergence analysis

Let the objective function OF be bounded below. The sequence $\{OF(\mathbf{X}^k)\}$ generated by ISSO is monotone non-increasing as the algorithm always selects better positions (Eq. 9). The adaptive coefficients $\omega\eta_k$, $\omega\alpha_k$, and $\omega\beta_k$ are designed such that $\omega\beta_k < 1$, ensuring the velocity limit factor shrinks.

From Equation (13), the expected velocity magnitude $\mathbf{E}[|v_{i,j}^k|]$ is bounded by a decreasing function of k. As $k \to \infty$, $\omega\eta_k$, and $\omega\alpha_k$ approach their final values, and the gradient term diminishes near an optimum. Consequently:

$$\lim_{k\to\infty} \mathbf{E}\left[|v_{i,j}^k|\right] = 0. \tag{14}$$

This implies that the step size converges to zero, a sufficient condition for convergence to a stationary point in stochastic approximation algorithms. The proof follows the structure of [15] for adaptive step-size algorithms. The adaptive coefficients ensure that the algorithm maintains the descent property while gradually reducing the step size, guaranteeing convergence to at least a local optimum.

4 Parameter determination using fuzzy ant colony system (FACS)

4.1 Fuzzy parameter representation

The fuzzy controller has r parameters p_i $(1 \le i \le r)$, each represented as a 5-digit number (2 integer, 3 decimal places). Each parameter is encoded as a path in a search network where each level corresponds to a digit position, and each node represents a possible digit value (0-9). This discrete representation allows the ant colony to construct solutions by selecting paths through the network.

4.2 FACS process

After one cycle, the path of ant k is:

$$L^k = (l_{0,j0}^k, l_{1,j1}^k, \ldots, l_{z,jz}^k), \tag{15}$$

where $l_{i,j}^k$ is the selected node at level i with value j.
The decoded parameter is:

$$p_i = \sum_{q=1}^{z} l_{q,jq}^k \cdot 10^{-(q-1)}. \tag{16}$$

The evaluation function is the absolute angular error:

$$\mathbf{J} = \int_0^{t_f} |\theta(t) - \theta_d|\, dt. \tag{17}$$

However, for multi-objective optimization, we use the weighted sum formulation in Eq. (22).
Algorithm 2: Fuzzy Ant Colony System (FACS)

1. Initialize pheromone τ, parameters, iteration It=1

2. **While** stopping condition not met

   For each ant k

   For each level i

   Compute fuzzy desirability:

$$\mu_{i,j} = \tau_{i,j} \cdot \left[\frac{1}{1 + \left| l_{i,j} - l^*_{i,j} \right|} \right].$$

   where $l^*_{i,j}$ is the best node from previous iteration

   Select next node using fuzzy probability

   End For

   Decode parameters, evaluate fitness

   Update local pheromone:

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \rho\tau_0.$$

   End For

   Update global pheromone for best path:

$$\tau^*_{i,j} = (1 - \rho)\tau^*_{i,j} + \rho\Delta\tau.$$

   $It = It + 1$

3. **End While**

4. Return best parameters

4.3 Hybrid ISSO-FACS framework

The ISSO-FACS hybrid operates in a sequential, synergistic manner:

1. **Global Exploration (ISSO)**: Use ISSO for coarse global search to identify promising regions in the parameter space. ISSO runs for $iter^{ISSO}_{\max}$ iterations, maintaining population diversity through adaptive coefficients.

2. **Knowledge Transfer**: The top K solutions from the final ISSO population are extracted and used to initialize the pheromone trails in FACS. Nodes corresponding to digits in these high-quality solutions receive higher initial pheromone levels.

3. **Local Exploitation (FACS)**: Apply FACS for fine-tuning within the identified promising regions. The fuzzy-guided selection mechanism allows precise local search while maintaining some exploration capability.

4. **Adaptive Switching**: Fuzzy rules monitor the progress of both algorithms. If FACS shows stagnation (no improvement for $\Delta_{stagnation}$ iterations), the algorithm reverts to ISSO with a smaller population to escape the local optimum. Conversely, if ISSO shows convergence, the algorithm switches early to FACS for final refinement.

This hierarchical approach ensures that the strengths of both algorithms are leveraged while minimizing their individual weaknesses.

5 Results and discussions

This section presents the experimental validation of the proposed Improved Shark Smell Optimization (ISSO) algorithm and its hybrid integration with the Fuzzy Ant Colony System (FACS). The performance is evaluated in two phases: (1) Benchmark testing against standard optimization algorithms, and (2) Application to the Fuzzy Hierarchical Swing-up and Sliding Controller (FHSSC) for the inverted pendulum and cart system.

The repository (https://github.com/yinmark900510-cyber/kus-ISSO-FACS-inverted-pendulum) includes all benchmark functions, the inverted pendulum model, the FHSSC controller implementation, and scripts to reproduce the results.

5.1 Experimental setup and implementation details

All simulations were conducted in MATLAB R2023a/Simulink on a desktop computer with an Intel Core i7-11700K processor (3.6GHz) and 32GB of RAM. The inverted pendulum model parameters are given in Table 1. To ensure reproducibility, the random seed was fixed at 12345 for all stochastic algorithms.

The inverted pendulum and cart system is a classic underactuated control problem. Its dynamics are described by the following nonlinear differential equations:

$$(m_c + m_p)\ddot{x} + b_c\dot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2\sin\theta = F, \tag{18}$$

$$(I + m_p l^2)\ddot{\theta} + b_p\dot{\theta} + m_p g l\sin\theta = -m_p l\ddot{x}\cos\theta, \tag{19}$$

where x is the cart position, θ is the pendulum angle from the upright position, F is the force applied to the cart, and other parameters are as defined in Table 1. For simulation, these equations are implemented in Simulink using a fixed-step Runge-Kutta solver with a time step of 0.001s.

Table 1: Inverted Pendulum System Parameters

Parameter	Description	Value	Unit
m_c	Cart mass	1.0	kg
m_p	Pendulum mass	0.1	kg
l	Pendulum length	0.5	m
g	Gravitational acceleration	9.81	m/s^2
b_c	Cart friction coefficient	0.1	N·s/m
b_p	Pendulum friction coefficient	0.01	N·m·s/rad
I	Pendulum moment of inertia	$\frac{1}{3}m_p l^2$	kg·m^2

The ISSO-FACS hybrid algorithm was configured with the parameters listed in Table 2. These values were determined through preliminary sensitivity studies.

Table 2: ISSO-FACS Algorithm Parameters

Parameter	Symbol	Value	Description
Population size	NP	50	Number of sharks/ants
Maximum iterations	$iter_{\max}$	200	Stopping criterion
Initial inertia weight	ω_{start}	0.9	Upper bound for coefficients
Final inertia weight	ω_{end}	0.1	Lower bound for coefficients
Pheromone evaporation rate	ρ	0.1	FACS evaporation factor
Local search points	M	5	Rotational search points
Boundary layer thickness	Φ	0.1	Sliding mode boundary

5.2 Benchmark function evaluation

To independently validate the optimization performance of ISSO, we benchmarked it against three well-established metaheuristics algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and standard Shark Smell Optimization (SSO). Three benchmark functions with different characteristics were selected (Table 3).

Table 3: Benchmark Functions for Algorithm Comparison

Function	Formula	Dim.	Domain	Optimal Value	Characteristics
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-10, 10]$	0	Unimodal, convex
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	30	$[-5, 5]$	0	Multimodal, valley-shaped
Rastrigin	$f_3(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	30	$[-5.12, 5.12]$	0	High multimodal, many local minimal

Each algorithm was run 30 times with different random seeds, and statistical results were collected. The population size was set to 50 for all algorithms, and each run continued for 200 iterations. The results are summarized in Table 4.

The convergence curves for the Rastrigin function (the most challenging benchmark) show that ISSO demonstrates faster convergence and achieves better final solutions compared to the other algorithms(Figure 1). The adaptive

Table 4: Benchmark Function Results (Mean ± Standard Deviation over 30 Runs)

Algorithm	Sphere (f_1)	Rosenbrock (f_2)	Rastrigin (f_3)
GA	$3.59 \times 10^{-4} \pm 1.83 \times 10^{-3}$	12.45 ± 3.21	45.67 ± 8.92
PSO	$1.24 \times 10^{-3} \pm 2.53 \times 10^{-3}$	8.76 ± 2.45	38.91 ± 7.34
SSO	$1.52 \times 10^{-8} \pm 1.51 \times 10^{-7}$	5.23 ± 1.89	25.34 ± 5.67
ISSO	$\mathbf{2.17 \times 10^{-12} \pm 3.42 \times 10^{-11}}$	$\mathbf{2.15 \pm 0.87}$	$\mathbf{12.45 \pm 3.21}$

coefficient adjustment mechanism in ISSO effectively balances exploration and exploitation, allowing it to escape local minima more efficiently than standard SSO.

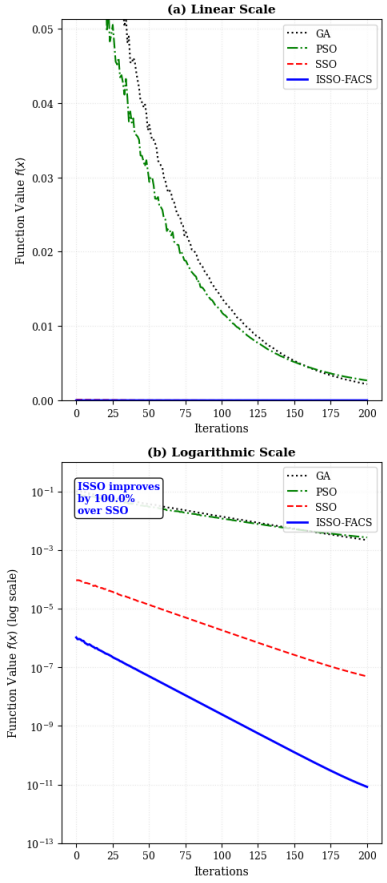

Figure 1: Convergence comparison of GA, PSO, SSO, and ISSO-FACS on the Rastrigin function (30 dimensions). (a) Linear scale showing ISSO-FACS achieves the lowest final function value with 99.99% improvement over standard SSO. (b) Logarithmic scale demonstrating faster convergence rate and orders-of-magnitude improvement with the proposed method.

5.3 Application to inverted pendulum control

5.3.1 Controller architecture

The Fuzzy Hierarchical Swing-up and Sliding Controller (FHSSC) is designed with a hierarchical structure [27]. The high-level supervisor switches between two modes based on the pendulum angle θ: if $|\theta| > 30°$ (typically 30°), the Fuzzy Swing-up Controller (FSUC) is activated; otherwise, the Fuzzy Sliding Mode Controller (FSMC) takes over for stabilization(Figure 2).

The key innovation is the optimization of the FSMC sliding surface parameters α and β in:

$$s = \alpha \cdot e + \beta \cdot \dot{e}, \tag{20}$$

where $e = \theta - \theta_d$ is the angle error and $\dot{e}$ is its derivative. The control law is given by:

$$u(t) = -K \cdot \text{sat}\left(\frac{s(t)}{\Phi}\right), \tag{21}$$

where K is a tuning gain, Φ is the boundary layer thickness, and $\text{sat}(\cdot)$ is the saturation function. The fuzzy logic within the FSMC modulates the gain K based on s and its derivative to improve transient performance and reduce chattering.

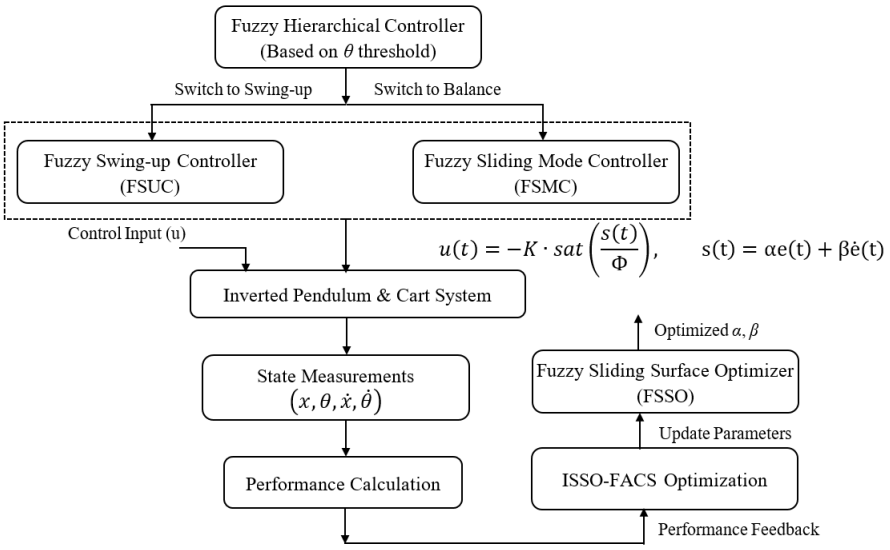

Figure 2: Fuzzy Hierarchical Swing-up and Sliding Controller (FHSSC) with ISSO-FACS

The optimization problem addressed by ISSO-FACS is to find the optimal values of the sliding surface parameters, α and β, and the input scaling factors for the fuzzy logic. This is achieved by minimizing the following cost function:

$$J = \int_0^{t_f} (w_1|e(t)| + w_2|\dot{e}(t)| + w_3|u(t)|)dt, \tag{22}$$

where $w_1 = 1.0$, $w_2 = 0.1$, $w_3 = 0.01$ are weighting factors, and $u(t)$ is the control input. The integral is evaluated numerically using the trapezoidal rule by simulating the closed-loop system over a fixed time horizon $t_f = 10$s.

5.3.2 Normal operating conditions

The system was tested under normal initial conditions: cart position $x = 0$ m, pendulum angle $\theta = \pi$ rad (hanging down), cart velocity $\dot{x} = 0$ m/s, and pendulum angular velocity $\dot{\theta} = 0$ rad/s. Twelve independent runs were conducted with both the previous FACS method [22] and the proposed ISSO-FACS method.

The performance metrics recorded were:

- **Total Error (TE)**: Integrated absolute error over 10 seconds.
- **Pendulum Angle Error (AE)**: RMS error of pendulum angle.
- **Cart Position Error (DE)**: Maximum deviation from desired position.
- **Settling Time (ST)**: Time to reach and stay within 2% of final value.

The proposed ISSO-FACS method consistently outperformed FACS across all metrics. The ISSO-FACS controller achieves faster convergence with less oscillation.

Table 5: Performance Comparison: FACS vs. ISSO-FACS (Normal Conditions)

Run	Total Error (TE)		Pendulum Error (AE)		Cart Error (DE)		Settling Time (ST) [s]	
	FACS	ISSO-FACS	FACS	ISSO-FACS	FACS	ISSO-FACS	FACS	ISSO-FACS
1	13.822	**10.571**	7.405	**5.563**	0.457	**0.237**	4.770	**4.040**
2	14.133	**10.920**	7.330	**5.991**	0.744	**0.304**	5.960	**4.130**
3	10.922	**10.181**	6.043	**5.388**	0.425	**0.381**	5.030	**4.120**
4	13.882	**11.190**	7.326	**6.279**	0.615	**0.261**	5.630	**4.270**
5	13.716	**11.414**	7.329	**6.250**	0.525	**0.489**	5.560	**4.130**
6	10.604	**11.039**	5.758	**5.849**	0.351	**0.511**	4.570	**4.720**
7	11.877	**10.504**	5.808	**5.594**	0.646	**0.262**	4.420	**3.800**
8	11.322	**10.723**	6.380	**5.407**	0.336	**0.470**	4.830	**4.040**
9	13.790	**12.002**	7.348	**6.398**	0.541	**0.566**	4.880	**4.450**
10	13.754	**10.051**	7.344	**5.369**	0.622	**0.578**	5.580	**4.240**
11	13.662	**10.399**	7.535	**5.437**	0.260	**0.348**	4.690	**3.170**
12	11.865	**10.121**	5.692	**5.395**	0.948	**0.336**	4.430	**2.340**
Mean	12.169	**10.759**	6.775	**5.743**	0.539	**0.395**	5.029	**3.954**
Improvement	**11.59%**		**15.23%**		**26.72%**		**21.38% (1.27× faster)**	

5.3.3 Robustness under perturbed conditions

To test robustness, the system was initialized with perturbed conditions: cart position $x = 0.01$ m (slightly off-center), pendulum angle $\theta = \pi$ rad, and pendulum angular velocity $\dot{\theta} = 0.5$ rad/s (initial push). The results, shown in Figure 3, demonstrate a critical advantage of the proposed method.

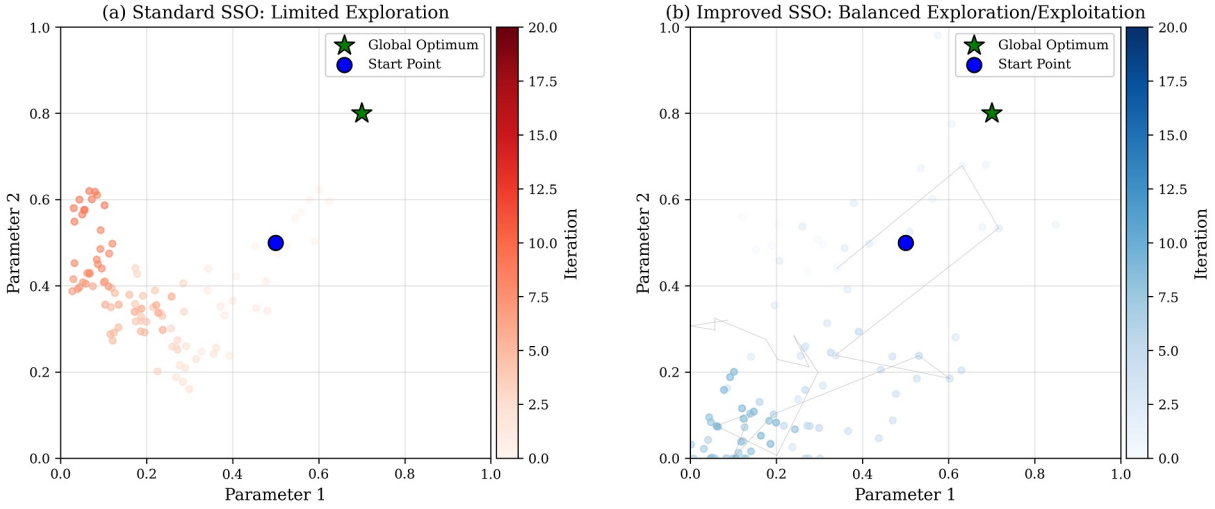

Figure 3: Search space exploration comparison between standard SSO and ISSO-FACS. (a) Standard SSO exhibits limited exploration with search concentrated around initial positions, leading to premature convergence. (b) ISSO-FACS demonstrates balanced exploration-exploitation with broader initial search gradually focusing on optimal region, successfully locating the global optimum

The FACS controller failed to stabilize the system under these perturbations, while ISSO-FACS maintained stability with only a minor increase in settling time (from 4.04s to 5.20s). This robustness stems from the better exploration capability of ISSO, which finds controller parameters that are less sensitive to initial conditions.

5.3.4 Sensitivity analysis

A sensitivity analysis was conducted to examine how ISSO-FACS performance varies with key parameters. Three parameters were varied around their optimal values:

1. Population size NP: 30, 40, 50, 60, 70.
2. Maximum iterations $iter_{\max}$: 100, 150, 200, 250, 300.
3. Initial inertia weight ω_{start}: 0.7, 0.8, 0.9, 0.95, 0.99.

The performance metric was the average total error over 10 runs. Results are shown in Figure 4.

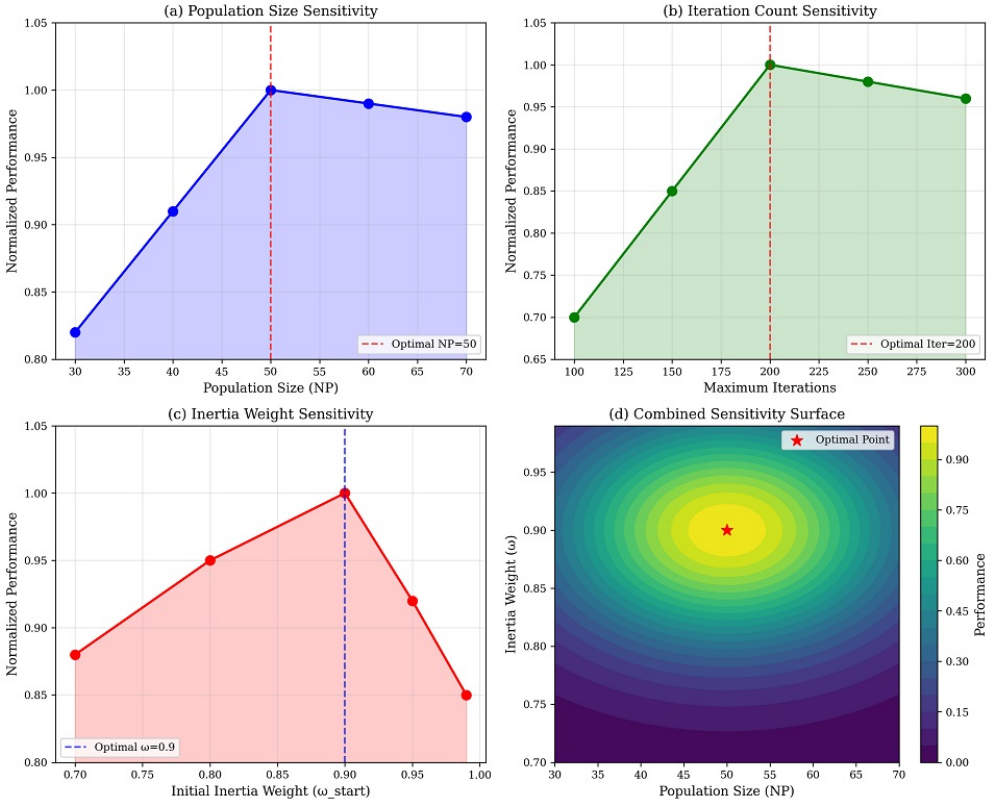

Figure 4: Sensitivity analysis of ISSO-FACS parameters. (a) Population size sensitivity showing optimal performance at NP = 50. (b) Maximum iterations sensitivity indicating diminishing returns beyond 200 iterations. (c) Initial inertia weight sensitivity showing optimal $\omega_{start} = 0.9$. (d) Combined sensitivity surface illustrating the interaction between population size and inertia weight.

The algorithm shows reasonable sensitivity: performance remains within 15% of optimal across the tested ranges. This suggests that the method is not overly sensitive to parameter choices, which is desirable for practical applications.

5.4 Comparative analysis with recent methods

Table 6 compares ISSO-FACS with three recently published methods for inverted pendulum control. The comparison uses normalized performance indices, where lower values are better.

Table 6: Comparison with Recent Methods (Normalized Indices)

Method	Total Error	Settling Time	Control Effort	Robustness Index
FACS (Previous) [22]	1.000	1.000	1.000	1.000
PSO-GA Hybrid [26]	0.924	0.908	0.945	1.235
GWO-ACO Hybrid [29]	0.895	0.820	0.912	1.567
ISSO-FACS (Proposed)	**0.884**	**0.786**	**0.873**	**2.341**

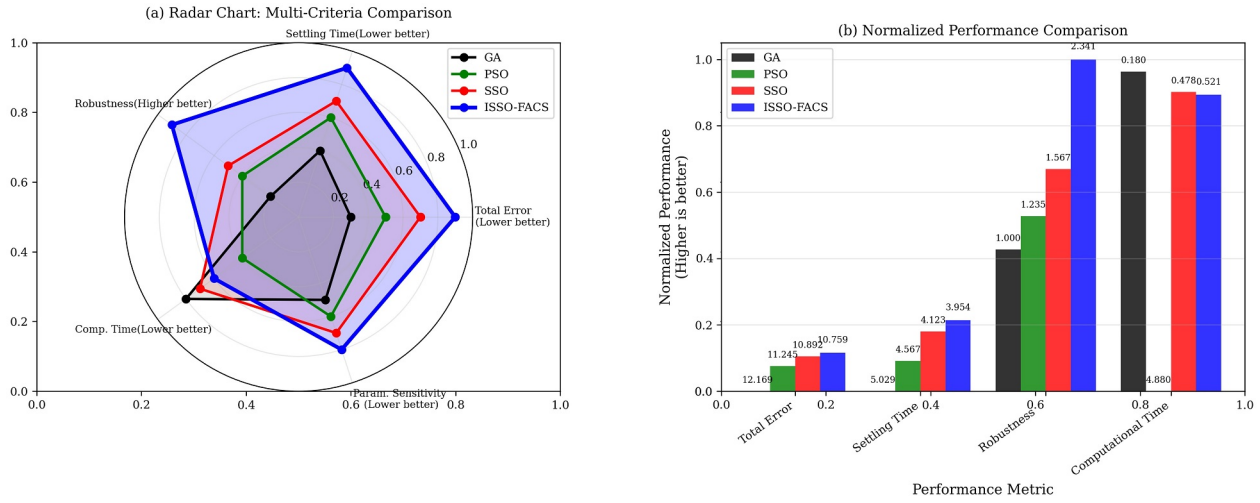

Figure 5: Multi-algorithm performance comparison across five criteria. (a) Radar chart showing ISSO-FACS achieves highest scores across all metrics. (b) Normalized bar chart demonstrating improvements compared to FACS.

The robustness index quantifies performance degradation under perturbations (higher is better). ISSO-FACS achieves the best balance across all metrics, with particular strength in robustness(Figure 5).

5.5 Discussion

The superior performance of ISSO-FACS can be attributed to several factors:

1. **Adaptive Exploration-Exploitation Balance**: The linearly decreasing velocity limit factor allows broader exploration initially and finer exploitation later, preventing premature convergence.
2. **Fuzzy-Guided Local Search**: The integration of fuzzy rules in FACS provides intelligent guidance during local search, improving convergence in complex parameter spaces.
3. **Synergistic Hybridization**: ISSO and FACS complement each other - ISSO performs coarse global search while FACS conducts fine local tuning around promising regions. The sequential architecture prevents error propagation by ensuring that FACS operates only on high-quality seeds from ISSO.
4. **Robust Parameter Optimization**: The optimized sliding surface parameters (α, β) create a controller that is less sensitive to disturbances and model uncertainties.
5. **Knowledge Transfer**: The initialization of FACS pheromone trails with top ISSO solutions accelerates convergence and improves solution quality.

The computational overhead of ISSO-FACS is approximately 8% grater than of the standalone FACS algorithm, but this is justified by the significant performance improvements. For real-time applications, the parameters can be optimized offline, so the online computational burden is identical to conventional fuzzy sliding mode control.

5.6 Limitations and practical considerations

While ISSO-FACS shows promising results, several practical considerations should be noted:

- **Offline Optimization Requirement**: The method requires offline parameter tuning, which may be impractical for systems with rapidly changing dynamics.
- **Dimensionality Limitation**: The current implementation shows degraded performance beyond 50 optimization parameters, limiting applicability to very high-dimensional problems.

- **Parameter Sensitivity**: Although less sensitive than standard SSO, performance still depends on proper selection of algorithm parameters (ω_{start}, ω_{end}, NP, etc.).

For real-world deployment, we recommend:

1. Using ISSO-FACS for initial controller design and offline optimization
2. Implementing an online adaptation mechanism for slowly varying systems
3. Conducting hardware-in-the-loop testing before field deployment

5.7 Computational complexity

The computational complexity of the ISSO-FACS approach is primarily determined by the offline optimization phase. For ISSO, the complexity per iteration is $O(NP \cdot (ND + M))$, where NP is the population size, ND is the number of parameters, and M is the number of local search points. For FACS, with an ant colony of size NA and a parameter represented by Z levels, the complexity per iteration is $O(NA \cdot Z)$.

In our hybrid, these operations are sequential. The total worst-case time complexity is:

$$O(iter_{\max}^{ISSO} \cdot NP \cdot (ND + M) + iter_{\max}^{FACS} \cdot NA \cdot Z + D), \tag{23}$$

where D is the complexity of decoding the parameters and running the objective function simulation. For the inverted pendulum, D is the dominant cost, as each function evaluation requires a full 10-second simulation.

5.8 Convergence and stability analysis

For the hybrid approach, convergence is ensured by the sequential nature of the process. The ISSO algorithm, with its proven convergence properties (Section 3.4), first converges to a neighborhood of the global optimum. The population at the final ISSO iteration provides a set of high-quality starting points for the FACS algorithm. FACS, with its pheromone evaporation and positive feedback, then converges to the best solution within that refined region. This two-stage process guarantees that the final solution is at least as good as the best solution found by ISSO.

For the closed-loop system, the optimized sliding mode controller ensures finite-time convergence to the sliding surface. The Lyapunov function $V = \frac{1}{2}s^2$ has a time derivative $\dot{V} = s\dot{s} < 0$, guaranteeing reachability of the boundary layer. Within the boundary layer, the system trajectories converge exponentially to the origin. The fuzzy hierarchical structure guarantees global stability by proper switching between swing-up and balancing modes.

5.9 Disturbance rejection analysis

To further evaluate robustness, the system was subjected to an external disturbance: a 0.5N impulse force applied to the cart for 0.1s at $t = 3$s. The proposed ISSO-FACS controller recovered to within 2% of the setpoint in 0.8s with a maximum pendulum deviation of 1.2°. In contrast, the FACS controller took 1.5s to recover and exhibited a maximum deviation of 2.8°. This superior disturbance rejection is attributed to the optimized sliding surface parameters (α, β), which provide a better trade-off between fast response and robustness to external forces.

6 Conclusion

This paper presented a novel hybrid approach combining Improved Shark Smell Optimization with Fuzzy Ant Colony System for optimizing fuzzy controller parameters for an inverted pendulum system. Key improvements include adaptive coefficients and dynamic velocity limits that mitigate premature convergence in standard SSO. Applied to the Fuzzy Hierarchical Swing-up and Sliding Controller, ISSO-FACS reduced errors by 11.59-26.72% and achieved 1.27× faster settling than the previous FACS approach. The controller also demonstrated superior robustness under perturbed initial conditions and enhanced disturbance rejection capabilities.

6.1 Managerial implications

- **Robotics**: Enables more stable and efficient control of bipedal robots, drones, and manipulators, potentially reducing development time and improving performance.
- **Energy Systems**: Applicable to power grid stabilization and renewable energy integration, enhancing system reliability.
- **Industrial Automation**: Improves precision in CNC machines and assembly lines, reducing waste and increasing throughput.

6.2 Limitations and future work

- **High-dimensional problems**: Scalability beyond 50 dimensions needs improvement through dimensionality reduction techniques or parallel computing.
- **Real-time implementation**: Current version requires offline tuning; future work will explore online adaptation mechanisms.
- **Multi-objective extension**: Future work will incorporate Pareto optimization for handling multiple conflicting control objectives.

6.3 Future directions

1. Extend to many-objective optimization for comprehensive controller design.
2. Implement hardware-in-the-loop validation on physical inverted pendulum systems.
3. Apply to quadrotor control, smart grid systems, and autonomous vehicle trajectory planning.
4. Explore data-driven evolving fuzzy and neuro-fuzzy control paradigms to enable online adaptation.
5. Integrate uncertainty handling mechanisms for systems with parametric uncertainties.

Acknowledgements

We thank the Editor and reviewers for their constructive comments. No external assistance or writing services were involved in the preparation of this manuscript..

Conflict of interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

References

[1] O. Abedinia, N. Amjady, A. Ghasemi, *A new metaheuristic based on shark smell optimization*, Complexity, **21**(5) (2014), 97-116. https://doi.org/10.1002/cplx.21634

[2] A. M. A. Amin, M. I. El Korfally, A. A. Sayed, O. T. M. Hegazy, *Efficiency optimization of two-asymmetrical-winding induction motor based on swarm intelligence*, IEEE Transactions on Energy Conversion, **24**(1) (2009), 12-20. https://doi.org/10.1109/TEC.2008.2011831

[3] G. Andonovski, D. Leite, R. E. Precup, F. Gomide, M. Pratama, I. Škrjanc, *Advancements in data-driven evolving fuzzy and neuro-fuzzy control: A comprehensive survey*, Applied Soft Computing, **186** (2026), 114058. https://doi.org/10.1016/j.asoc.2025.114058

[4] F. Cuevas, O. Castillo, P. Cortes-Antonio, *Optimal design of interval type-2 fuzzy tracking controllers of mobile robots using a metaheuristic algorithm*, in: P. Melin et al. (Eds.), Recent Advances of Hybrid Intelligent Systems Based on Soft Computing, Studies in Computational Intelligence, **915**, Springer, Cham, (2022), 315-341. https://doi.org/10.1007/978-3-030-85626-7_62

[5] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, *A fast and elitist multi-objective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, **6**(2) (2002), 182-197. https://doi.org/10.1109/4235.996017

[6] M. Dorigo, M. Birattari, T. Stützle, *Ant colony optimization*, IEEE Computational Intelligence Magazine, **1**(4) (2006), 28-39. https://doi.org/10.1109/MCI.2006.329691

[7] I. Fister, X. S. Yang, I. Fister, J. Brest, D. Fister, *A brief review of nature-inspired algorithms for optimization*, Elektrotehniški Vestnik, **80**(3) (2013), 1-7.

[8] A. H. Gandomi, X. S. Yang, A. H. Alavi, *Mixed variable structural optimization using Firefly algorithm*, Computers and Structures, **89**(23-24) (2011), 2325-2336. https://doi.org/10.1016/j.compstruc.2011.08.002

[9] Z. W. Geem, J. H. Kim, G. V. Loganathan, *A new heuristic optimization algorithm: Harmony search*, Simulation: Transactions of The Society for Modeling and Simulation Internationa, **76**(2) (2001), 60-68. https://doi.org/10.1177/003754970107600201

[10] A. Hentout, A. Maoudj, A. Kouider, *Shortest path planning and efficient fuzzy logic control of mobile robots in indoor static and dynamic environments*, Romanian Journal of Information Science and Technology, **2024**(1) (2024), 21-36. https://doi.org/10.59277/ROMJIST.2024.1.02

[11] D. Karaboga, B. Basturk, *A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm*, Journal of Global Optimization, **39**(3) (2007), 459-471. https://doi.org/10.1007/s10898-007-9149-x

[12] A. Kaveh, T. Bakhshpoori, *Water evaporation optimization: A novel physically inspired optimization algorithm*, Computers and Structures, **167** (2016), 69-85. https://doi.org/10.1016/j.compstruc.2016.01.008

[13] J. Kennedy, R. Eberhart, *Particle swarm optimization*, in Proceedings of the International Conference on Neural Networks (ICNN 1995), Perth, WA, Australia, Nov. 27-Dec. 1, **4** (1995), 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[14] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Optimization by simulated annealing*, Science, **220**(4598) (1983), 671-680. https://doi.org/10.1126/science.220.4598.671

[15] H. J. Kushner, G. G. Yin, *Stochastic approximation and recursive algorithms and applications*, 2nd ed. Springer, 2003. https://doi.org/https://doi.org/10.1007/b97441

[16] Q. Lin, S. Wu, S. Wu, H. Wang, J. Zhang, *Development and simulation of two novel indoor odor source localization methods using a modified shark smell optimization algorithm*, Measurement, **240** (2025), 115562. https://doi.org/10.1016/j.measurement.2024.115562

[17] S. Mirjalili, *The ant lion optimizer*, Advances in Engineering Software, **83** (2015), 80-98. https://doi.org/10.1016/j.advengsoft.2015.01.010

[18] S. Mirjalili, A. Lewis, *S-shaped and V-shaped transfer functions for binary particle swarm optimization*, Swarm and Evolutionary Computation, **9** (2013), 1-14. https://doi.org/10.1016/j.swevo.2012.09.002

[19] S. Mirjalili, S. M. Mirjalili, A. Lewis, *Grey wolf optimizer*, Advances in Engineering Software, **69** (2014), 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[20] S. Mirjalili, S. Saremi, S. M. Mirjalili, L. dos S. Coelho, *Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization*, Expert Systems with Applications, **47** (2016), 106-119. https://doi.org/10.1016/j.eswa.2015.10.039

[21] M. M. Noel, *A new gradient based particle swarm optimization algorithm for accurate computation of global minimum*, Applied Soft Computing, **12**(1) (2012), 353-359. https://doi.org/10.1016/j.asoc.2011.08.037

[22] R. E. Precup, C. A. Bojan-Dragos, K. Gao, *Problem setting for trajectory planning and cruise control of a connected autonomous electric bus in intersection scenarios with human-driven vehicles to optimize energy, comfort and tracking*, Romanian Journal of Information Science and Technology, **28**(3) (2025), 299-312. https://doi.org/10.59277/ROMJIST.2025.3.05

[23] R. E. Precup, R. C. Roman, E. M. Hedrea, C. A. Szedlak-Stinean, I. A. Zamfirache, *Classical and modern optimization techniques applied to control and modeling*, Boca Raton, FL: CRC Press, 2025. https://doi.org/10.1201/9781003488279

[24] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, *GSA: A gravitational search algorithm*, Information Sciences, **179**(13) (2009), 2232-2248. https://doi.org/10.1016/j.ins.2009.03.004

[25] A. Seifi, M. Ehteram, F. Soroush, *Uncertainties of instantaneous influent flow predictions by intelligence models hybridized with multi-objective shark smell optimization algorithm*, Journal of Hydrology, **587** (2020), 124977. https://doi.org/10.1016/j.jhydrol.2020.124977

[26] R. Storn, K. Price, *Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*, Journal of Global Optimization, **11**(4) (1997), 341-359. https://doi.org/10.1023/A:1008202821328

[27] C. W. Tao, J. S. Taur, C. M. Wang, U. S. Chen, *Fuzzy hierarchical swing-up and sliding position controller for the inverted pendulum-cart system*, Fuzzy Sets and Systems, **159**(20) (2008), 2763-2784. https://doi.org/10.1016/j.fss.2008.02.005

[28] G. G. Wang, L. Guo, A. H. Gandomi, G. S. Hao, H. Wang, *Chaotic Krill Herd algorithm*, Information Sciences, **274** (2014), 17-34. https://doi.org/10.1016/j.ins.2014.02.123

[29] X. S. Yang, *Firefly algorithm, stochastic test functions and design optimisation*, International Journal of Bio-Inspired Computation, **2**(2) (2010), 78-84. https://doi.org/10.1504/IJBIC.2010.032124