IJRCS

International Journal of Robotics and Control Systems
Vol. 6, No. 1, 2026, pp. 173-189
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

A Memory-Based Simultaneous Perturbation Stochastic Approximation Method for the Optimal PID Controller Tuning in Flexible Joint Manipulators

Nik Mohd Zaitul Akmal Mustapha [a,1], Mohd Ashraf Ahmad [a,2,*], Mohd Riduwan Ghazali [a,3], Mohd Helmi Suid [a,4]

[a] Faculty of Electrical and Electronic Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia
[1] nikmdzaitul@umpsa.edu.my; [2] mashraf@umpsa.edu.my; [3] riduwan@umpsa.edu.my; [4] mhelmi@umpsa.edu.my
[*] Corresponding Author

ARTICLE INFO

Article History
Received August 16, 2025
Revised November 03, 2025
Accepted January 03, 2026

Keywords
Memory-Based Simultaneous Perturbation Stochastic Approximation;
Data-Driven Control;
PID Controller;
Flexible Joint Manipulator;
Optimal Control

ABSTRACT

This paper presents a memory-based Simultaneous Perturbation Stochastic Approximation (M-SPSA) method for the optimal Proportional–Integral–Derivative (PID) controller tuning in flexible joint manipulators. The study addresses convergence instability in standard SPSA variants by introducing a memory-retention mechanism that preserves the best-performing design variable. The research contribution is the formulation of a memory-driven SPSA framework that improves convergence stability and robustness while maintaining computational efficiency. The proposed method is validated through simulation using a nonlinear flexible joint manipulator model. Its performance is compared with two established variants, namely Norm-Limited SPSA (NL-SPSA) and Normalized SPSA (N-SPSA), to evaluate convergence behavior, energy efficiency, and tracking accuracy. Results show that the proposed M-SPSA achieves consistent and stable convergence across all simulation trials. The algorithm effectively minimizes the objective-function value and reduces control energy while maintaining accurate trajectory tracking. Compared with NL-SPSA and N-SPSA, the proposed method exhibits improved convergence reliability and smoother control input behavior. These outcomes indicate that incorporating memory retention enhances stability and resilience in stochastic optimization applied to flexible robotic systems. Future work will focus on real-time validation, and extension of the approach to multi-input multi-output and adaptive control systems to strengthen its practical applicability in advanced robotics.

1. Introduction

The use of optimization algorithms in data-driven controller tuning has proven essential for accurately estimating control parameters, which often outperform classical tuning methods [1], [2]. These algorithms have been widely applied in motor drives [3], robotics [4], process control [5], aerospace systems [6], and thermal power plants [7]. They play a crucial role in enabling adaptive and robust control, particularly when mathematical models are difficult to obtain or when real-time reconfiguration is required in dynamic environments [8], [9].

Flexible Joint Manipulators (FJMs) represent a particularly significant and complex domain for data-driven control methodologies. These systems, including lightweight robotic arms and space-based manipulators, exhibit highly nonlinear dynamics, underactuated structures, and strong coupling between rigid and flexible components [10], [11]. These characteristics make FJMs a suitable and challenging benchmark for evaluating data-driven optimization algorithms because they require precise coordination between flexibility and control accuracy. Ensuring both compliance and precision in FJMs is crucial in high-stakes applications such as medical robotics and aerospace operations. Conventional model-based control strategies often require precisely identified mathematical models and are highly sensitive to parameter uncertainties, limiting their practicality in dynamic or unpredictable environments. As a result, data-driven optimization techniques have gained prominence as robust, model-independent, and adaptive alternatives for controller tuning in Flexible Joint Manipulator (FJM) systems [12], [13]. Nevertheless, these methods often demand substantial computational resources, especially in high-dimensional or real-time scenarios. This limitation highlights the need for lightweight, efficient algorithms that maintain control accuracy while minimizing computational overhead [14], [15].

Generally, optimization strategies can be categorized into population-based and trajectory-based methods. Population-based metaheuristics such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Archimedes Optimization Algorithm (AOA), and Grey Wolf Optimizer (GWO) have been extensively applied in controller tuning across various domains. PSO and its enhanced variants have achieved faster response times, reduced overshoot, and improved robustness in DC motor and robotic manipulator control [16]–[21]. Likewise, ACO, AOA, and GWO have demonstrated superior performance in electric-vehicle control, voltage regulation, and automatic voltage regulation systems [22]–[24]. GA-based PID tuning has also been used to enhance stability and minimize steady-state deviation in DC-motor and power-system applications [25], [26]. Comprehensive surveys [27]–[29] confirm that hybrid and adaptive versions of these metaheuristics deliver strong global performance across many control problems.

Despite their success, population-based methods remain computationally intensive because each iteration requires numerous function evaluations across large populations. This scalability issue limits their suitability for real-time and embedded controller tuning, where computation time and sample efficiency are critical [30], [31]. Consequently, trajectory-based optimization approaches, particularly SPSA, have gained attention as lightweight, data-efficient alternatives capable of achieving comparable performance with drastically fewer evaluations. This distinction forms the rationale for focusing on SPSA and its variants in this study.

In contrast, trajectory-based optimization methods recently employed in data-driven control primarily focus on tuning PID [32], [33], fuzzy logic [34], and neural network [35], [36] controllers. These techniques operate by refining a single candidate solution through stochastic perturbations of its design parameters, thereby leading to significantly reduced computational effort compared to population-based algorithms. Among these, the Simultaneous Perturbation Stochastic Approximation (SPSA) [37] algorithm has emerged as a particularly attractive candidate due to its ability to approximate gradients with only two objective function measurements. As a result, SPSA and its variants have found practical applications in diverse areas such as level control system [38]–[40], well placement [41], [42], system identification [43], training neural network [44], Dynamic Traffic Assignment (DTA) [45], optimal control problem [46], distributed sensor networks [47], Unmanned Aerial Vehicle (UAV) [48], and quality control for medium voltage insulator [49]. Since its inception, SPSA has evolved into multiple advanced formulations aimed at addressing specific convergence, efficiency, and robust challenges. Early developments include a one-measurement SPSA to enhance efficiency in non-stationary systems by requiring only a single objective evaluation per iteration. This was followed by second-order SPSA and adaptive variants, which leverage concurrent Hessian estimation to exploit local curvature, thereby accelerating convergence, particularly in ill-conditioned optimization

landscapes [50]. Generalized SPSA (G-SPSA) has been proposed to broaden the standard algorithm by introducing more flexible perturbation and gradient estimation schemes, thereby improving convergence across a range of problem structures [51]. Meanwhile, one-sided SPSA techniques approximate gradients by contrasting perturbed and unperturbed measurements, thereby reducing the number of required evaluations while maintaining convergence behavior [52]. For high-dimensional parameter tuning, multi-resolution SPSA (MR-SPSA) decomposes the optimization process into sequential phases with varying design vector dimensions, leading to more rapid convergence through progressive refinement [53], [54]. More recently, weighted SPSA (W-SPSA) approaches integrate stochastic gradient perturbations with consensus dynamics to enable distributed cooperative target tracking, enhancing convergence rates and track accuracy in multi-agent systems under uncertainty [55]. Collectively, these advancements underscore the versatility of SPSA and its adaptations in tackling diverse optimization problems across engineering domains.

In data-driven control frameworks, SPSA serves as a powerful tool for optimizing controller parameters. However, directly applying the standard SPSA algorithm to such tuning tasks can be challenging. Due to its stochastic update mechanism, parameter adjustments may become excessively large, potentially driving the system toward instability. Furthermore, since the performance objective function is often not defined in closed form, it becomes difficult to ensure that the resulting tuned parameters will consistently yield a stable closed-loop response under all operating conditions. To mitigate these issues, several SPSA variants have been proposed. The Norm-Limited SPSA (NL-SPSA) [53], [56] introduces a saturation function to restrict the magnitude of parameter updates, thereby enhancing convergence stability. Nevertheless, this improvement comes at the cost of additional parameters and a narrower search space, which may constrain exploration. To address this, the Normalized SPSA (N-SPSA) [57] normalizes perturbation-based measurements relative to their peak values at each iteration, improving convergence uniformity without introducing extra tuning parameters. While both approaches enhance robustness, they remain reactive in nature and operate solely on instantaneous feedback, without leveraging historical information from previous iterations.

Although these improvements are notable, both NL-SPSA and N-SPSA remain memoryless optimizers. Each iteration updates parameters based only on immediate perturbation feedback, without considering the historical quality of prior solutions. As a result, in noisy or time-varying environments, these algorithms may exhibit erratic convergence behavior or discard previously optimal parameter sets. This limitation reveals a critical gap: the absence of a stabilizing mechanism capable of retaining useful historical information while maintaining SPSA's lightweight computational structure. The proposed memory-retention mechanism directly addresses this shortcoming by introducing a persistent feedback path that reuses the best-performing design variable as a reference for subsequent updates. Conceptually, this transforms the SPSA framework from a purely reactive process into a history-aware stochastic approximation model, marking a new paradigm that embeds memory within single-trajectory optimization rather than merely adjusting algorithmic parameters. In contrast to saturation-based or normalization-based variants, the memory mechanism operates as a stabilizing anchor that continuously references the best historical state, thereby preventing stochastic divergence and premature convergence. It also enhances exploration stability in multimodal landscapes without introducing additional hyperparameters, ensuring that the search trajectory remains both adaptive and computationally lightweight.

Building upon this identified gap, our earlier work introduced the memory-based Simultaneous Perturbation Stochastic Approximation (M-SPSA) [58], a variant that enhances tuning stability by retaining the best-performing design variable associated with the minimum objective function value throughout the optimization process, thereby preventing deterioration due to stochastic noise. When previously validated on a liquid-slosh suppression system, M-SPSA achieved lower objective function values, reduced variance, and faster convergence than the NL-SPSA while maintaining comparable computational cost. Extending this algorithm to FJMs introduces additional nonlinear coupling,

underactuated dynamics, and oscillatory flexible-link behavior, creating a more demanding optimization landscape that further validates the robustness and adaptability of the memory mechanism under complex and non-stationary conditions.

This paper investigates the application of the proposed M-SPSA for optimal PID tuning of FJMs. The study begins by illustrating the convergence instability commonly encountered with the standard SPSA algorithm, followed by the presentation of the proposed M-SPSA incorporating a memory-retention mechanism that preserves the best-performing parameter set throughout the optimization process. This strategy is then integrated into a PID tuning framework specifically designed for flexible manipulators to evaluate its practical effectiveness. The algorithm's performance is assessed through simulation, demonstrating notable improvements in convergence reliability, tracking accuracy, and control efficiency. The research contributions of this study are as follows:

i. Formulation of the M-SPSA algorithm that enhances convergence stability through adaptive memory-based parameter updating.
ii. Integration and validation of M-SPSA for data-driven PID tuning in FJMs, achieving improved tracking accuracy, energy efficiency, and convergence reliability compared to previous SPSA variants.

2. Methodology

This section begins by introducing the standard SPSA algorithm and highlighting its typical convergence challenges. To maintain methodological consistency and ensure fair benchmarking, the illustrative example from [59] is adopted for evaluating the proposed approach.

2.1. Standard SPSA

Fig. 1 illustrates the general framework of data-driven optimization tools, where algorithms such as SPSA and its variants adaptively tune controller parameters in real time. The controller generates the control input based on the reference signal, while the plant output is compared with the reference to compute the error. This error, evaluated through a performance index, guides the optimization tool in updating the controller parameters and minimizing the objective function. This process enables adaptive, data-driven tuning without the need for an explicit plant model.

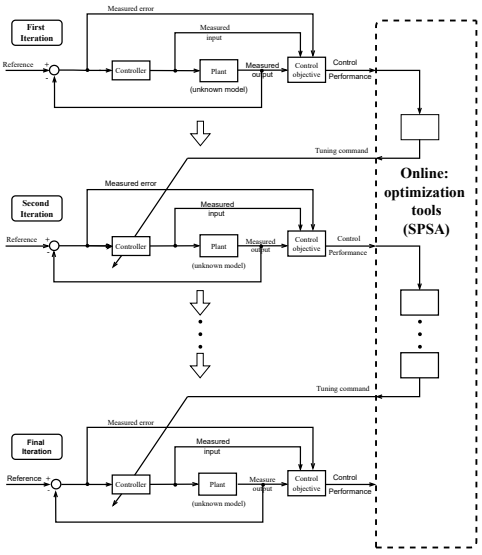

Fig. 1. Research flow of data-driven optimization tools framework [60]

Building upon this framework, the standard SPSA algorithm serves as the baseline optimization approach for this study. Originally introduced by Spall [37], SPSA is a stochastic optimization method that estimates the gradient of an objective function through simultaneous perturbations of all parameters. The algorithm requires only two function evaluations per iteration, regardless of the problem dimension, making it computationally efficient for noisy or high-dimensional optimization tasks.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be the objective function, and $z \in \mathbb{R}^n$ represent the design variable to be optimized. The optimization goal can be expressed as:

$$\min_{z \in \mathbb{R}^n} f(z) \tag{1}$$

Within the SPSA framework, the design variable z is iteratively updated as

$$z_{k+1} = z_k - a_k g(z_k) \tag{2}$$

Here, $z_k \in \mathbb{R}^n$ is the design variable at iteration k, $a_k \in \mathbb{R}_+$ is a positive scalar gain, and $g(z_k) \in \mathbb{R}^n$ represents the gradient approximation obtained via stochastic perturbation.

$$g(z_k) = \begin{bmatrix} \dfrac{f(z_k + c_k\Delta_k) - f(z_k - c_k\Delta_k)}{2c_k\Delta_{1k}} \\ \dfrac{f(z_k + c_k\Delta_k) - f(z_k - c_k\Delta_k)}{2c_k\Delta_{2k}} \\ \vdots \\ \dfrac{f(z_k + c_k\Delta_k) - f(z_k - c_k\Delta_k)}{2c_k\Delta_{nk}} \end{bmatrix} \tag{3}$$

The stochastic gradient is estimated as (3), where the scalar $c_k \in \mathbb{R}_+$ defines a secondary gain parameter, $\Delta_k \in \mathbb{R}^n$ is a vector of randomly generated perturbations, and $\Delta_{ik} \in \mathbb{R}$ denotes the i-th component of Δ_k. The expected value of $g(z_k)$ closely approximates the true gradient of $f(z_k)$, allowing SPSA to operate as a randomized form of the classical steepest-descent method. Table 1 presents a stepwise summary of the standard SPSA procedure, which forms the basis for subsequent variants, including N-SPSA [50], NL-SPSA [59], and the proposed memory-based SPSA (M-SPSA) developed in this study.

2.2. Unstable Convergence of Standard SPSA

To highlight the limitations of the standard SPSA, a benchmark scenario consistent with [59] and [50] is adopted. The nonlinear objective function is defined as:

$$f(z) = \left((z-1)^{\mathrm{T}}(z-1)\right)^3 \tag{4}$$

And possesses a global minimum at $z = \mathbf{1}$ for dimension $n = 10$. Following [50], the gain sequences are defined as $a_k = 0.05/(k+200)^{0.602}$ and $c_k = 0.01/(k+1)^{0.101}$, while the perturbation vector Δ_k is drawn from a symmetric Bernoulli distribution and the initial design variable is initialized as $z_0 = \mathbf{0}$.

Table 1. Standard SPSA algorithm [37]

Step	Description
S1	Initialize the gain sequence parameters a, A, α, c, and γ for $a(k)$ and $c(k)$. Assign an initial guess for the design variable $z(0)$, define the total number of iterations $k_{\max}$, and set the iteration index $k = 0$.
S2	Construct a random perturbation vector $\Delta(k)$, with each component typically sampled from a symmetric Bernoulli distribution.
S3	Perform two evaluations of the objective function at the perturbed design variable: $f(z(k) + c(k)\Delta(k))$ and $f(z(k) - c(k)\Delta(k))$.
S4	Estimate the gradient $g(z_k)$ using: $$g(z_k) = \frac{f(z(k) + c(k)\Delta(k)) - f(z(k) - c(k)\Delta(k))}{2c(k)} \cdot \Delta(k)^{-1}$$
S5	Update the current design variable as follows: $$z(k+1) = z(k) - a(k) \cdot g(z_k)$$
S6	If $k = k_{\max}$, return the final solution $z^* = z(k_{\max})$; otherwise, increment $k = k + 1$ and return to Step S2.

To demonstrate the convergence characteristics of the standard SPSA, the algorithm is evaluated on this benchmark configuration. Fig. 2 illustrates the divergence of the objective function value over successive iterations, highlighting the algorithm's inability to approach the global minimum under these gain settings.

This result shows that divergence can occur even when theoretical convergence conditions are satisfied. The instability arises from unbounded update norms inherent in the stochastic approximation process, rather than parameter mis-tuning or implementation error.

Consequently, this example motivates stabilization strategies such as the NL-SPSA [59], the N-SPSA [50], and ultimately the M-SPSA proposed in this study.

Building upon these prior stabilization strategies, the following section introduces a memory-based formulation that further mitigates stochastic divergence while preserving the computational efficiency of the SPSA framework.

2.3. Memory-based SPSA (M-SPSA)

This part of the study focuses on addressing convergence instability that commonly arises with the standard SPSA algorithm. To overcome this, the M-SPSA method integrates a memory-based strategy that continually stores the best design variable encountered during the optimization.

Let $z(k) \in \mathbb{R}^n$ represent the design variable at iteration k, and let $f(z)$ be the objective function to minimize. At each iteration, a perturbation vector $\Delta(k) \in \mathbb{R}^n$ is generated, with each element $\Delta_i(k)$ drawn from a symmetric Bernoulli distribution. To facilitate a more concise mathematical representation, the perturbed parameters are defined as $z^+ = z_k + c_k\Delta_k$ and $z^- = z_k - c_k\Delta_k$. The associated objective function evaluations are denoted by $f(z^+) = f(z_k + c_k\Delta_k)$ and $f(z^-) = f(z_k - c_k\Delta_k)$, respectively. Using these evaluations, the gradient is approximated as:

$$\hat{g}_i(k) = \frac{f(z^+(k)) - f(z^-(k))}{2c(k)\Delta_i(k)}, \quad i = 1, \ldots, n \tag{5}$$

In the following, we present an improved update formulation for the design variable, modifying (2) as:

$$z_{k+1} = z_k - a_k\hat{g}(z_k) \tag{6}$$

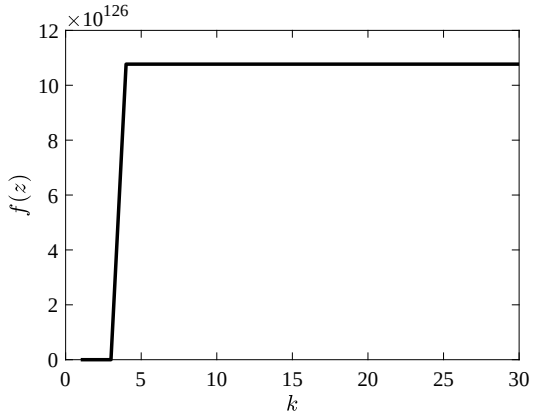

Fig. 2. Illustration of unstable convergence of $f(z)$ under standard SPSA

To enhance stability, the algorithm retains the best-performing solution across iterations using the following rule:

$$z_{\text{best}}(k+1) = \begin{cases} z(k+1), & \text{if } f(z(k+1)) < f(z_{\text{best}}(k)) \\ z_{\text{best}}(k), & \text{otherwise} \end{cases} \tag{7}$$

The algorithm proceeds iteratively until the specified maximum number of iterations, $k_{\max}$, has been completed. At this point, the optimal solution is defined as $z^* = z_{\text{best}}(k_{\max})$. Table 2 summarizes the complete procedural steps of the proposed M-SPSA algorithm, where the memory mechanism is explicitly incorporated following the gradient-evaluation and update stages (corresponding to Steps S3 and S5 in Table 1).

The performance of the M-SPSA is assessed by re-evaluating the objective function defined in (4). To ensure a fair comparison, both algorithms are limited to the same Number of Function Evaluations (NFE) rather than the number of iterations. While the standard SPSA performs two objective function evaluations per iteration, the M-SPSA requires three evaluations due to the additional memory update. Therefore, the total number of iterations for each algorithm is proportionally adjusted to maintain equivalent computational effort. For example, the standard SPSA with $k_{\max} = 30$ ($2\times30 = 60$ evaluations) is compared against the M-SPSA with $k_{\max} = 20$ ($3\times20 = 60$ evaluations), ensuring both algorithms consume the same number of objective function evaluations.

This adjustment provides an unbiased and computationally fair basis for performance comparison. Under these identical conditions, the proposed approach formulated in (5) is tested using the same experimental setup and gain sequences, except for the modified gain factor defined as $a_k = 0.004/(k+200)^{0.5}$. Fig. 3 clearly demonstrates that the M-SPSA significantly enhances stability and achieves faster and more reliable minimization of the objective function, thereby addressing the convergence limitations observed in the standard SPSA.

The inclusion of the memory mechanism in M-SPSA is theoretically motivated by the elitist-retention principle widely used in population-based optimization algorithms such as PSO and GA. Following the elitist principle that preserves the global–best solution to guide swarm behavior, the M–SPSA algorithm retains the best historical design variable, θ_{best}, and only updates it when a new candidate solution yields a lower objective value.

This adaptive retention rule constrains the search trajectory within a stable region, preventing stochastic divergence while maintaining exploration capability. By incorporating this bounded feedback mechanism, M-SPSA achieves improved convergence reliability without compromising global search efficiency. When both algorithms are compared under an equivalent number of function eval-

uations, as clarified in this section, the proposed method demonstrates enhanced stability and faster convergence trends.

Table 2. Memory-based SPSA (M-SPSA) algorithm [58]

Step	Description
S1	Set the parameters a, A, α, c, and γ for the gain sequences $a(k)$ and $c(k)$. Define the initial design variable $z(0)$, maximum iteration count $k_{\max}$, and initialize $k = 0$. Also set $f_{\text{best}} = f(z(0))$ and $z_{\text{best}} = z(0)$.
S2	Generate a random perturbation vector $\Delta(k)$.
S3	Evaluate the objective function at two perturbed points: $f^+ = f(z(k) + c(k)\Delta(k))$ and $f^- = f(z(k) - c(k)\Delta(k))$.
S4	Update the best solution: If $f^+ < f_{\text{best}}$, then set $f_{\text{best}} = f^+$, $z_{\text{best}} = z(k) + c(k)\Delta(k)$; else if $f^- < f_{\text{best}}$, set $f_{\text{best}} = f^-$, $z_{\text{best}} = z(k) - c(k)\Delta(k)$.
S5	Approximate the gradient $\hat{g}(z_k)$ using $$\hat{g}(z_k) = \frac{f^+ - f^-}{2c(k)} \cdot \Delta(k)^{-1}.$$
S6	Update the design variable $$z(k+1) = z(k) - a(k) \cdot \hat{g}(z_k).$$
S7	If $f(z(k+1)) < f_{\text{best}}$, then set $f_{\text{best}} = f(z(k+1))$, $z_{\text{best}} = z(k+1)$; else, assign $z(k+1) = z_{\text{best}}$.
S8	If $k = k_{\max}$, return $z^* = z_{\text{best}}$; otherwise, increment $k = k + 1$ and repeat from Step S2.

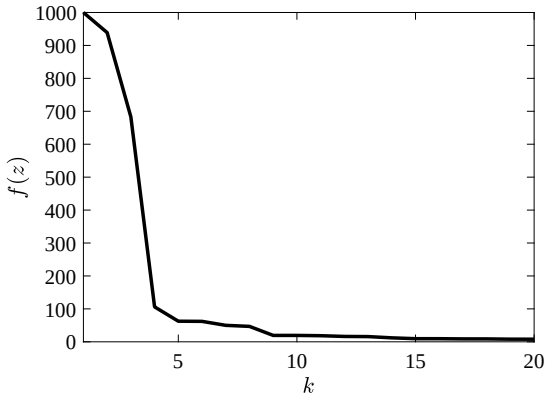

Fig. 3. Illustration of stable convergence of $f(z)$ under M-SPSA

2.4. Computational Complexity and Comparative Analysis with Elitist PSO

To provide a comprehensive perspective on the computational implications of the proposed approach, the complexity of M-SPSA is examined and compared with that of the elitist PSO algorithm. The computational complexity of the proposed M-SPSA is mainly governed by the parameter update and objective-evaluation steps described in Table 2.

Updating the design variable $z(k)$ requires $O(p)$ arithmetic operations per iteration, where p denotes the dimensionality of the problem, while the evaluation of the objective function at three perturbed points, f^+, f^-, and $f(z_{k+1})$, adds a constant number of function evaluations independent of p. Hence, for a maximum iteration count $k_{\max}$, the overall computational complexity of M-SPSA can be expressed as $O(k_{\max} \times p)$. This linear relationship with respect to both iteration count and

problem size ensures scalability in high-dimensional control tuning applications. In contrast, the elitist PSO algorithm, as detailed in the pseudocode of [61], maintains a population of N_p particles, each requiring independent objective evaluations and updates of local $(pbest_i, fbest_i)$ and global $(gbest, f_{gbest})$ best positions.

Accordingly, its arithmetic and memory costs scale as $O(k_{\max} \times N_p \times p)$, reflecting a substantially higher computational load due to its population-based structure. While PSO employs an elitism mechanism that preserves both global and local best solutions across all particles, the proposed M-SPSA retains only a single best pair $(z_{\text{best}}, f_{\text{best}})$, introducing merely $O(p)$ additional memory overhead. This design allows M-SPSA to maintain the trajectory-based efficiency of standard SPSA while integrating an elitist-inspired retention mechanism that improves convergence stability and robustness.

3. Results and Discussion

This section presents the performance analysis of the M-SPSA algorithm applied to the data-driven PID tuning of FJMs. The main objective is to minimize oscillatory motion typically observed in lightweight mechanical structures, such as flexible joint systems [62].

As shown in Fig. 4, the control framework consists of a reference signal $r(t)$ that represents the desired tip angle, the actual output angle $\theta(t)$, the deflection angle $\alpha(t)$ of the flexible link, and the control input $u(t)$. The system dynamics are represented by plant G, adopted from [62], and serve as the foundation for controller design and evaluation.

$$G := \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{8}$$

The matrices A, B, C, and D are formulated as per [62], enabling a full state-space representation of the flexible joint plant. This model is used to configure the PID controller structure for each control loop as:

$$K_i(s) = P_i\left(1 + \frac{1}{I_i s} + \frac{D_i s}{1 + \frac{D_i}{N_i} s}\right) \tag{9}$$

Here, for $i = 1, 2$, the parameters $P_i, I_i, D_i, N_i \in \mathbb{R}$ represent the proportional gain, integral time constant, derivative time constant, and derivative filter coefficient, respectively. In the context of flexible joint manipulation, the primary objective is to ensure that the output angle $\theta(t)$ accurately tracks the reference signal $r(t)$. Concurrently, the control mechanism should suppress vibration at the flexible link, reflected in the deflection angle $\alpha(t)$, while minimizing the magnitude of the control input $u(t)$. To evaluate controller performance, the following objective function is formulated to represent tracking accuracy, vibration suppression, and control-effort efficiency:

$$J(\psi) = 400 \int_0^4 |r(t) - \theta(t)|^2 \, dt + 400 \int_0^4 |\alpha(t)|^2 \, dt + \int_0^4 |u(t)|^2 \, dt \tag{10}$$

Where $\psi = [P_1 \; I_1 \; D_1 \; N_1 \; P_2 \; I_2 \; D_2 \; N_2]^{\mathrm{T}} \in \mathbb{R}^8$ denotes the vector of PID tuning parameters. The reference signal $r(t)$ applied during simulation is defined as:

$$r(t) = \begin{cases} 50t, & 0 \le t \le 1, \\ 50, & 1 < t \le 4. \end{cases} \tag{11}$$

Remark: The reference signal $r(t)$, as defined in (11), follows a ramp profile from 0 to 1 second and then holds steady at 50°.

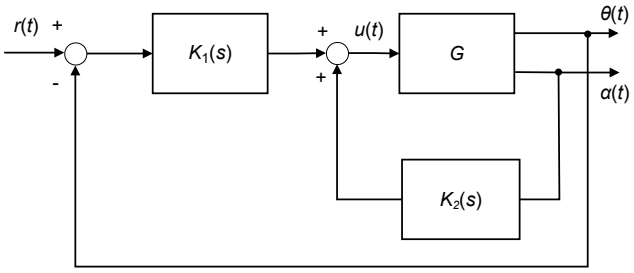

Fig. 4. Block diagram of the PID-based control architecture for the flexible joint system [50]

This excitation strategy is intentionally adopted to evaluate the performance of the data-driven PID controller tuned via M-SPSA, particularly during the ramp interval when tracking precision is most critical. Such input profiles are frequently utilized in control applications to prevent excessive vibrations that can result from abrupt step inputs.

The M-SPSA algorithm, as formulated in (5), is implemented by setting the objective function f to the cost function J, and applying a logarithmic transformation to the decision variables: $z_i = \log \psi_i$ for $i = 1, 2, \ldots, 8$, where z_i denotes the i-th element of vector z. This transformation improves numerical stability and enables more effective exploration of the parameter space. The gain sequences are defined as $a_k = \frac{0.206}{(k+22)^{0.716}}$ and $c_k = \frac{0.09}{(k+1)^{0.0892}}$, with the initial vector set as $z_0 = [0.5 \quad 1.0 \quad 0.0 \quad 1.0 \quad 0.0 \quad 1.0 \quad 0.0 \quad 2.0]$. To assess the robustness of M-SPSA, 30 independent simulations are conducted. Performance metrics include the average, best, worst, and standard deviation (Std.) of the cost function, in addition to the error norm and control input norm.

As illustrated in Fig. 5, the proposed M-SPSA algorithm demonstrates stable convergence within 133 iterations for data-driven PID tuning of the flexible joint system. This is notably faster than the 200 iterations typically required for NL-SPSA and N-SPSA [50], confirming the improved convergence efficiency achieved through the memory-retention mechanism that preserves the best-performing design variable during optimization.

The final design variable at iteration $k = 133$ is obtained as $\mathbf{Z}_{133} = [1.3544 \quad 0.8827 \quad -0.8279 \quad 2.3370 \quad -0.4811 \quad 0.7868 \quad -1.2220 \quad 2.0064]$, which corresponds to the optimized PID parameters $\boldsymbol{\psi}_{133} = [22.62 \quad 2.96 \quad 3.367 \quad 217.27 \quad 0.33 \quad 0.05 \quad 0.02 \quad 101.48]$. These results validate that M-SPSA achieves faster and smoother convergence while maintaining robustness and parameter stability under simulation conditions.

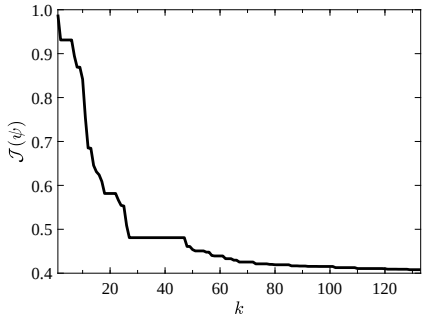

Fig. 5. Convergence of the objective function $J(\psi)$ using M-SPSA

Fig. 6, Fig. 7 and Fig. 8 depict the system responses for $\theta(t)$, $\alpha(t)$, and $u(t)$, respectively. In each figure, the red-dotted line denotes the system output under the initial PID parameters ($k = 0$),

while the solid black curve shows the optimized performance achieved at $k = 133$.

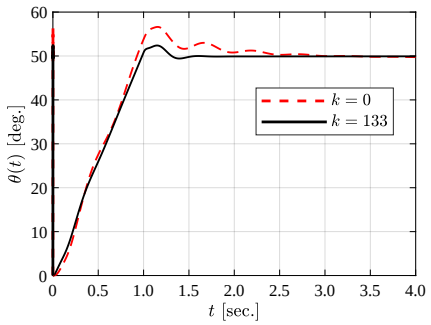

Fig. 6. Evolution of $\theta(t)$ response with M-SPSA iterations

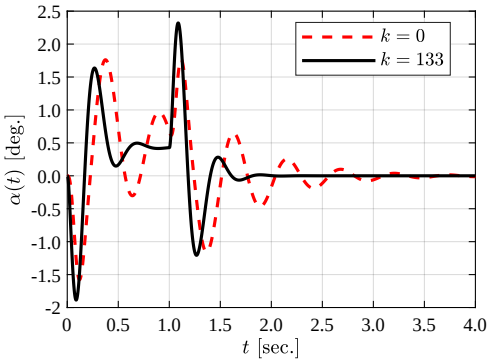

Fig. 7. Evolution of $\alpha(t)$ response with M-SPSA iterations

As observed in Fig. 6, M-SPSA improves angular tip position tracking, achieving negligible steady-state error and minimal overshoot. This outcome aligns with earlier findings on N-SPSA [50], which also demonstrated enhanced tracking under optimized tuning. In this case, M-SPSA attains comparable accuracy with fewer iterations, underscoring its convergence efficiency. These results highlight the algorithm's stability and precise control response.

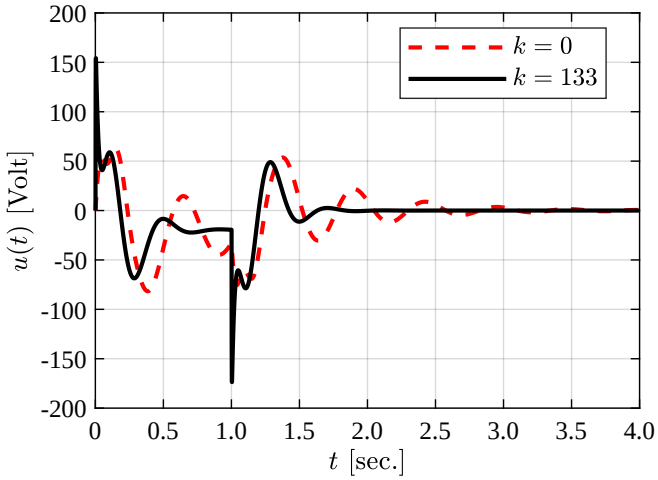

Fig. 8. Evolution of $u(t)$ response with M-SPSA iterations

Furthermore, as shown in Fig. 7, M-SPSA improves the damping of oscillations in the link deflection response, achieving stabilization in approximately 2 seconds. Although the tuned controller produces a slightly larger peak deflection (ranging from $-1.9°$ to $2.4°$), the suppression of oscillatory motion is considerably faster than with the baseline parameters. This outcome is consistent with earlier findings using N-SPSA [50], which also reported enhanced damping through optimized tuning.

In the present study, M-SPSA attains similar suppression performance with fewer iterations, demonstrating improved convergence efficiency. These results highlight the algorithm's capability to enhance transient stability while maintaining computational efficiency.

As shown in Fig. 8, M-SPSA produces a quicker settling response in the control signal. Although the input stabilizes more rapidly than in the baseline case, it exhibits a spike of about $-180\,\mathrm{V}$, which increases the risk of actuator saturation in practical applications. This emphasizes the importance of incorporating saturation-aware control strategies in real implementations.

Consistent with earlier studies using N-SPSA [50], which also reported faster response under optimized tuning, the present results show that M-SPSA achieves comparable performance while improving convergence efficiency. The findings highlight a clear trade-off: improved tracking accuracy and faster settling are accompanied by a transient peak.

Table 3 presents a comprehensive statistical comparison of three SPSA-based data-driven PID control strategies: NL-SPSA and N-SPSA, both adopted from [50], and the proposed M-SPSA. The comparison is based on 30 independent simulation runs and evaluates performance using three quantitative metrics: the overall cost function $J(\psi)$, the combined squared tracking error and control effort, and the total control-input energy.

Note that the bold values in Table 3 indicate the best performance for each metric. These statistical outcomes provide a quantitative benchmark for assessing the convergence efficiency, stability, and energy-optimization capability of the proposed algorithm relative to existing SPSA variants.

Table 3. Statistical performance comparison among NL-SPSA, N-SPSA, and M-SPSA

Algorithm		NL-SPSA [50]	N-SPSA [50]	M-SPSA
$J(\psi)$ $(\times 10^3)$	Average	4.1203	4.1027	**4.1017**
	Best	4.1013	4.0717	**4.0655**
	Worst	**4.1493**	4.1486	4.1544
	Std.	**0.0126**	0.0191	0.0249
$\int_0^4 \lvert r(t)-\theta(t)\rvert^2 dt + \int_0^4 \lvert\alpha(t)\rvert^2 dt$	Average	3.3717	**3.3521**	3.4776
	Best	3.3039	3.2457	**3.0997**
	Worst	**3.4108**	3.4306	3.6442
	Std.	**0.0257**	0.0303	0.1285
$\int_0^4 \lvert u(t)\rvert^2\, dt$ $(\times 10^3)$	Average	2.7716	2.7618	**2.7106**
	Best	2.7486	2.7363	**2.6330**
	Worst	**2.7987**	2.8180	2.8748
	Std.	**0.0126**	0.0187	0.0587

In terms of the cost function $J(\psi)$, M-SPSA achieves the lowest average and best-case performance, corresponding to improvements of 0.45% and 0.87% over NL-SPSA, and 0.02% and 0.15% over N-SPSA, respectively. These results emphasize that M-SPSA attains comparable optimization performance to N-SPSA while offering more efficient convergence relative to NL-SPSA.

Regarding the combined metric of squared tracking error and control effort, N-SPSA achieves the lowest average value, indicating consistent performance across trials. However, M-SPSA provides the best-case outcome, with improvements of 6.18% compared to NL-SPSA and 4.50% compared to N-SPSA. This demonstrates that M-SPSA can reduce transient deviations and sustain accurate trajectory tracking under optimal tuning conditions, while delivering comparable average performance to N-

SPSA.

With respect to control input energy, M-SPSA achieves the lowest average and best-case performance, showing improvements of 2.20% and 4.21% over NL-SPSA, while outperforming N-SPSA by 1.86% and 3.78%. Although the latter improvement is relatively small, it indicates that M-SPSA maintains energy efficiency comparable to the best existing variants, which is particularly important in applications where actuator saturation and energy consumption are critical.

To strengthen the reliability of these results, a Wilcoxon Signed-Rank test was conducted to assess statistical significance among the three methods. The analysis yielded a p-value of 0.0005 at the 5% level when comparing M-SPSA with NL-SPSA, confirming the statistical superiority of M-SPSA. In contrast, the comparison between M-SPSA and N-SPSA indicated no statistically significant difference ($p = 0.992$), suggesting that both algorithms perform at a comparable level. These results indicate that while M-SPSA addresses the instability and variability observed in NL-SPSA, it sustains performance levels consistent with N-SPSA, thereby validating its robustness and reliability for nonlinear system identification of flexible manipulators.

3.1. Practical Implications and Closing Remarks

Collectively, the results confirm that M-SPSA delivers stable control performance within a deterministic simulation setup. The observed improvements translate into practical benefits for FJMs, particularly in applications that require precise motion control, reduced actuator effort, and reliable tracking performance.

From a practical perspective, improved convergence efficiency can shorten controller tuning time, while smoother and more energy-efficient actuation may contribute to extending actuator lifespan and improving operational dependability. These characteristics suggest that M-SPSA could serve as a viable approach for adaptive manipulators, cooperative robotic systems, and precision robotic joints, where robustness and consistency are crucial.

4. Conclusion

This paper presents an enhanced Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm that integrates a memory-retention mechanism for the optimal tuning of PID controllers in Flexible Joint Manipulators (FJMs). By integrating a memory mechanism that preserves the best-performing design variable throughout the optimization process, the proposed M-SPSA enhances convergence stability and reliability compared to other SPSA variants such as NL-SPSA and N-SPSA.

Simulation results confirm that M-SPSA achieves stable and efficient convergence with measurable reductions in the objective function and control energy while maintaining acceptable tracking accuracy and actuator smoothness. Beyond numerical improvements, the key contribution lies in demonstrating that memory retention introduces stability and robustness benefits in nonlinear, underactuated systems where stochastic divergence is common.

The algorithm offers a lightweight and scalable solution for reliable PID tuning in dynamic environments, with applicability across robotics domains such as adaptive manipulators, unmanned aerial vehicles (UAVs), and other flexible systems. Although the improvements are modest in magnitude, they provide meaningful practical gains, including smoother actuation and improved long-term reliability of control components. Future work will focus on real-time validation using an experimental hardware setup and on extending the approach to multi-input multi-output (MIMO) and adaptive control systems to reinforce its relevance for practical robotic applications.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research was funded by International Islamic University Malaysia and Universiti Malaysia Pahang Al-Sultan Abdullah through the IIUM-UMP Sustainable Research Collaboration Grant 2022, grant

number RDU223216.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] N. H. Sahrir and M. A. M. Basri, "Intelligent pid controller based on neural network for ai-driven control quadcopter uav," *International Journal of Robotics and Control Systems*, vol. 4, no. 2, pp. 691–708, 2024, https://doi.org/10.31763/ijrcs.v4i2.1374.

[2] A. Baharuddin and M. A. M. Basri, "Self-tuning pid controller for quadcopter using fuzzy logic," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 728–748, 2023, https://doi.org/10.31763/ijrcs.v3i4.1127.

[3] G. Li, R. Li, H. Hou, G. Zhang, and Z. Li, "A data-driven motor optimization method based on support vector regression—multi-objective, multivariate, and with a limited sample size," *Electronics*, vol. 13, no. 12, p. 2231, 2024, https://doi.org/10.3390/electronics13122231.

[4] M. Shi, Y. Cheng, B. Rong, W. Zhao, Z. Yao, and C. Yu, "Research on vibration suppression and trajectory tracking control strategy of a flexible link manipulator," *Applied Mathematical Modelling*, vol. 110, pp. 78–98, 2022, https://doi.org/10.1016/j.apm.2022.05.030.

[5] G. Mattera, A. Caggiano, and L. Nele, "Optimal data-driven control of manufacturing processes using reinforcement learning: an application to wire arc additive manufacturing," *Journal of Intelligent Manufacturing*, vol. 36, pp. 1291–1310, 2025, https://doi.org/10.1007/s10845-023-02307-w.

[6] S. L. Brunton *et al.*, "Data-driven aerospace engineering: Reframing the industry with machine learning," *AIAA Journal*, vol. 59, no. 8, pp. 2820–2847, 2021, https://doi.org/10.2514/1.J060131.

[7] G. Shi, M. Ma, D. Li, Y. Ding, and K. Y. Lee, "A process-model-free method for model predictive control via a reference model-based proportional-integral-derivative controller with application to a thermal power plant," *Frontiers in Control Engineering*, vol. 4, 2023, https://doi.org/10.3389/fcteg.2023.1185502.

[8] S. Du, Z. Huang, L. Jin, and X. Wan, "Recent progress in data-driven intelligent modeling and optimization algorithms for industrial processes," *Algorithms*, vol. 17, no. 12, p. 569, 2024, https://doi.org/10.3390/a17120569.

[9] H. Zhou, "Design model-free adaptive pid controller based on lazy learning algorithm," *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20220279, 2023, https://doi.org/10.1515/jisys-2022-0279.

[10] Y. Wang, M. Leibold, J. Lee, W. Ye, J. Xie and M. Buss, "Incremental Model Predictive Control Exploiting Time-Delay Estimation for a Robot Manipulator," in *IEEE Transactions on Control Systems Technology*, vol. 30, no. 6, pp. 2285-2300, 2022, https://doi.org/10.1109/TCST.2022.3142629.

[11] C. A. Fernández, "Control of flexible manipulator robots based on dynamic confined space of velocities: Dynamic programming approach," *Journal of Robotics and Control (JRC)*, vol. 3, no. 6, pp. 743–753, 2022, https://doi.org/10.18196/jrc.v3i6.16454.

[12] D. Pavlichenko and S. Behnke, "Real-Robot Deep Reinforcement Learning: Improving Trajectory Tracking of Flexible-Joint Manipulator with Reference Correction," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2671-2677, 2022, https://doi.org/10.1109/ICRA46639.2022.9812023.

[13] S. Xu, Z. Wu, and T. Shen, "High-precision control of industrial robot manipulator based on extended flexible joint model," *Actuators*, vol. 12, no. 9, p. 357, 2023, https://doi.org/10.3390/act12090357.

[14] H. Hizarci and S. İkizoğlu, "Position Control of Flexible Manipulator using PSO-tuned PID Controller," *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1-5, 2019, https://doi.org/10.1109/ASYU48272.2019.8946440.

[15] M. I. F. M. Hanif, M. A. Ahmad, and J. J. Jui, "PID tuning method using chaotic safe experimentation dynamics algorithm for elastic joint manipulator," *Journal Europeen des Systemes Automatises*, vol. 54, no. 5, pp. 693–698, 2021, https://doi.org/10.18280/jesa.540504.

[16] U. Kruthika and S. Paneerselvam, "Improved adaptive pso-based gain tuning for pid controllers in utility boilers," in *Procedia Computer Science*, vol. 230, pp. 183–192, 2023, https://doi.org/10.1016/j.procs.2023.12.073.

[17] E. S. Rahayu, A. Ma'arif, and A. Cakan, "Particle Swarm Optimization (PSO) tuning of PID control on DC motor," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, pp. 435–447, 2022, https://doi.org/10.31763/ijrcs.v2i2.476.

[18] D. K. Somwanshi and P. Sonwane, "PSO-tuned PID controller optimization for DC motor speed control," *China Petroleum Processing and Petrochemical Technology*, vol. 23, pp. 657–665, 2023, https://doi.org/10.1007/978-981-16-7664-2_7.

[19] J. E. Oche, H. A. Bashir, and T. J. Shima, "PSO-optimized model reference adaptive PID controller for precise DC motor speed control," *Nigerian Journal of Technological Development*, vol. 21, no. 4, pp. 135–144, 2024, https://doi.org/10.4314/njtd.v21i4.2473.

[20] G. Ahmed, A. Eltayeb, N. M. Alyazidi, I. H. Imran, T. Sheltami, and S. El-Ferik, "Improved particle swarm optimization for fractional order pid control design in robotic manipulator system: A performance analysis," *Results in Engineering*, vol. 24, p. 103089, 2024, https://doi.org/10.1016/j.rineng.2024.103089.

[21] J. Xu *et al*., "Design of Engine Cooling System Using Improved Particle Swarm Optimization Algorithm," in *IEEE Sensors Journal*, vol. 23, no. 17, pp. 19060-19072, 2023, https://doi.org/10.1109/JSEN.2023.3294961.

[22] A. S. A. Nair, "Ant Colony Optimization- Tuned Fractional Order PID Controller for Speed Regulation In Hybrid Electric Vehicle," *2024 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, pp. 1-5, 2024, https://doi.org/10.1109/RAEEUCCI61380.2024.10547797.

[23] L. K. Fong, M. S. Islam, and M. A. Ahmad, "Optimized pid controller of dc-dc buck converter based on archimedes optimization algorithm," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 658–672, 2023, https://doi.org/10.31763/ijrcs.v3i4.1113.

[24] M. A. Şen and M. Kalyoncu, "Optimal tuning of pid controller using grey wolf optimizer algorithm for quadruped robot," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 1, pp. 29–35, 2018, https://doi.org/10.17694/bajece.401992.

[25] E. W. Suseno and A. Ma'Arif, "Tuning of pid controller parameters with genetic algorithm method on dc motor," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 41–53, 2021, https://doi.org/10.31763/ijrcs.v1i1.249.

[26] Z. Qu, W. Younis, X. Liu, A. Khalique Junejo, S. Z. Almutairi and P. Wang, "Optimized PID Controller for Load Frequency Control in Multi-Source and Dual-Area Power Systems Using PSO and GA Algorithms," in *IEEE Access*, vol. 12, pp. 186658-186678, 2024, https://doi.org/10.1109/ACCESS.2024.3445165.

[27] K. Kanwar, J. Vajpai, and S. K. Meena, "Design of pso tuned pid controller for different types of plants," *International Journal of Information Technology*, vol. 14, pp. 2877–2884, 2022, https://doi.org/10.1007/s41870-022-01051-3.

[28] R. S. Patil, S. P. Jadhav, and M. D. Patil, "Review of intelligent and nature-inspired algorithms-based methods for tuning pid controllers in industrial applications," *Journal of Robotics and Control (JRC)*, vol. 5, no. 2, pp. 336–358, 2024, https://doi.org/10.18196/jrc.v5i2.20850.

[29] S. Yamparala and G. Rao, "Optimal design of fopid controller for dfig based wind energy conversion system using grey-wolf optimization algorithm," *International Journal of Renewable Energy Research (IJRER)*, vol. 12, no. 4, 2022, https://doi.org/10.20508/ijrer.v12i4.13446.g8594.

[30] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola, and B. M. Khammas, "Metaheuristic algorithms for pid controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, p. e09399, 2022, https://doi.org/10.1016/j.heliyon.2022.e09399.

[31] M. A. Ahmad, H. Ishak, A. N. K. Nasir, and N. A. Ghani, "Data-based pid control of flexible joint robot using adaptive safe experimentation dynamics algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 79–85, 2021, https://doi.org/10.11591/eei.v10i1.2472.

[32] A. K. Ali, "An optimal design for an automatic voltage regulation system using a multivariable pid controller based on hybrid simulated annealing – white shark optimization," *Scientific Reports*, vol. 14, no. 30218, 2024, URLhttps://doi.org/10.1038/s41598-024-79300-7.

[33] T. Roux-Oliveira, L. R. Costa, A. V. Pino, and P. Paz, "Extremum seeking-based adaptive pid control applied to neuromuscular electrical stimulation," *Anais da Academia Brasileira de Ciencias*, vol. 91, no. 1, 2019, https://doi.org/10.1590/0001-3765201820180544.

[34] X. Zhang, T. Wang, C. Cheng, and S. Wang, "Three-dimensional fuzzy modeling for nonlinear distributed parameter systems using simultaneous perturbation stochastic approximation," *Applied Sciences (Switzerland)*, vol. 14, no. 17, p. 7860, 2024, https://doi.org/10.3390/app14177860.

[35] D. M. Le, O. S. Patil, C. F. Nino and W. E. Dixon, "Accelerated Gradient Approach For Deep Neural Network-Based Adaptive Control of Unknown Nonlinear Systems," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 4, pp. 6299-6313, 2025, https://doi.org/10.1109/TNNLS.2024.3395064.

[36] Y. Angeles-García, H. Calvo, H. Sossa, and Álvaro Anzueto-Ríos, "Dynamic balance of a bipedal robot using neural network training with simulated annealing," *Frontiers in Neurorobotics*, vol. 16, 2022, https://doi.org/10.3389/fnbot.2022.934109.

[37] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," in *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332-341, 1992, https://doi.org/10.1109/9.119632.

[38] X. Li, Z. Yang, Y. Yang, X. Kong, C. Shi, and J. Shi, "Gk-spsa-based model-free method for performance optimization of steam generator level control systems," *Energies*, vol. 16, no. 24, p. 8050, 2023, https://doi.org/10.3390/en16248050.

[39] Z. Yang, X. Kong, P. Geng, X. Li, and C. Shi, "Hk-spsa based performance optimization method for steam generator liquid level control," *Annals of Nuclear Energy*, vol. 198, p. 110326, 2024, https://doi.org/10.1016/j.anucene.2023.110326.

[40] P. Geng, X. Kong, C. Shi, H. Liu, and J. Liu, "Ik-spsa-based performance optimization strategy for steam generator level control system of nuclear power plant," *Energies*, vol. 15, no. 19, p. 7387, 2022, https://doi.org/10.3390/en15197387.

[41] B. Pouladi, A. Karkevandi-Talkhooncheh, M. Sharifi, S. Gerami, A. Nourmohammad, and A. Vahidi, "Enhancement of SPSA algorithm performance using reservoir quality maps: Application to coupled well placement and control optimization problems," *Journal of Petroleum Science and Engineering*, vol. 189, p. 106984, 2020, https://doi.org/10.1016/j.petrol.2020.106984.

[42] C. Li, C. Fang, Y. Huang, H. Zuo, Z. Zhang, and S. Wang, "Infill well placement optimization for secondary development of waterflooding oilfields with spsa algorithm," *Frontiers in Energy Research*, vol. 10, 2022, https://doi.org/10.3389/fenrg.2022.1005749.

[43] N. M. Z. A. Mustapha and M. A. Ahmad, "Normalized spsa for hammerstein model identification of twin rotor and electro-mechanical positioning systems," *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 552–568, 2025, https://doi.org/10.1016/j.ijcce.2025.04.004.

[44] A. T. Abdulsadda and K. Iqbal, "An improved spsa algorithm for system identification using fuzzy rules for training neural networks," *International Journal of Automation and Computing*, vol. 8, pp. 333–339, 2011, https://doi.org/10.1007/s11633-011-0589-x.

[45] M. Qurashi, T. Ma, E. Chaniotakis and C. Antoniou, "PC–SPSA: Employing Dimensionality Reduction to Limit SPSA Search Noise in DTA Model Calibration," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1635-1645, 2020, https://doi.org/10.1109/TITS.2019.2915273.

[46] L. Wang and J. C. Spall, "Improved SPSA Using Complex Variables with Applications in Optimal Control Problems," *2021 American Control Conference (ACC)*, pp. 3519-3524, 2021, https://doi.org/10.23919/ACC50511.2021.9482740.

[47] A. Sergeenko, O. Granichin and A. V. Proskurnikov, "Advanced SPSA-based Algorithm for Multi-Target Tracking in Distributed Sensor Networks," *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2424-2429, 2020, https://doi.org.10.1109/CDC42340.2020.9303942.

[48] S. Çoban, "Optimization of pid controllers for uavs using spsa algorithm for enhanced flight performance," *Journal of Aviation*, vol. 9, pp. 28–33, 2025, https://doi.org/10.30518/jav.1603792.

[49] X. Kong, J. Guo, D. Zheng, J. Zhang, and W. Fu, "Quality control for medium voltage insulator via a knowledge-informed spsa based on historical gradient approximations," *Processes*, vol. 8, no. 2, p. 146, 2020, https://doi.org/10.3390/pr8020146.

[50] N. Mustapha, M. Suid, R. R. Ismail, A. Nasir, M. Ahmad, M. M. Tumari, and J. Jui, "A data-driven pid controller for flexible joint manipulator using normalized simultaneous perturbation stochastic approximation," *Journal of Advanced Manufacturing Technology (JAMT)*, vol. 13, no. 2, pp. 143–154, 2019, https://jamt.utem.edu.my/jamt/article/download/5514/3789.

[51] S. Pachal, S. Bhatnagar and P. L. A., "Generalized Simultaneous Perturbation-Based Gradient Search With Reduced Estimator Bias," in *IEEE Transactions on Automatic Control*, vol. 70, no. 7, pp. 4687-4702, 2025, https://doi.org/10.1109/TAC.2025.3532160.

[52] S. Li, Y. Xia, and Z. Xu, "Simultaneous perturbation stochastic approximation: towards one-measurement per iteration," *Numerical Algorithms*, vol. 94, pp. 1085–1101, 2023, https://doi.org/10.1007/s11075-023-01528-7.

[53] M. A. Ahmad, S. I. Azuma, and T. Sugie, "A model-free approach for maximizing power production of wind farm using multi-resolution simultaneous perturbation stochastic approximation," *Energies*, vol. 7, no. 9, pp. 5624–5646, 2014, https://doi.org/10.3390/en7095624.

[54] R. H. Mok and M. A. Ahmad, "Model-free wind farm power production optimization using multi-resolution optimized relative step size random search," *IETE Journal of Research*, vol. 68, no. 5, pp. 3158–3172, 2022, https://doi.org/10.1080/03772063.2020.1754933.

[55] V. Erofeeva, O. Granichin, O. Granichina, A. Proskurnikov and A. Sergeenko, "Weighted SPSA-based Consensus Algorithm for Distributed Cooperative Target Tracking," *2021 European Control Conference (ECC)*, pp. 1074-1079, 2021, https://doi.org/10.23919/ECC54610.2021.9655195.

[56] M. A. Ahmad, S. I. Azuma, and T. Sugie, "Identification of continuous-time hammerstein systems by simultaneous perturbation stochastic approximation," *Expert Systems with Applications*, vol. 43, pp. 51–58, 2016, https://doi.org/10.1016/j.eswa.2015.08.041.

[57] M. A. Ahmad, N. M. Z. A. Mustapha, A. N. K. Nasir, M. Z. M. Tumari, R. M. T. R. Ismail and Z. Ibrahim, "Using normalized simultaneous perturbation stochastic approximation for stable convergence in model-free control scheme," *2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 935-938, 2018, https://doi.org/10.1109/ICASI.2018.8394422.

[58] N. M. Z. A. Mustapha, M. Z. M. Tumari, M. H. Suid, R. M. T. R. Ismail, and M. A. Ahmad, "Data-Driven PID Tuning for Liquid Slosh-Free Motion Using Memory-Based SPSA Algorithm," *Proceedings of the 10th National Technical Seminar on Underwater System Technology 2018*, vol. 538, pp. 197–210, 2018, https://doi.org/10.1007/978-981-13-3708-6_17.

[59] Y. Tanaka, S. Azuma, and T. Sugie, "Simultaneous perturbation stochastic approximation with norm-limited update vector," *Asian Journal of Control*, vol. 17, no. 6, pp. 2083–2090, 2015, https://doi.org/10.1002/asjc.1153.

[60] G. M. Qian, M. R. B. Ghazali, and M. A. B. Ahmad, "Data-driven variable tracking differentiation sigmoid proportional-integral-derivative controller for nonlinear multiple input multiple output system with white noise," *Journal of Robotics and Control (JRC)*, vol. 6, no. 5, pp. 2399–2412, 2025, https://doi.org/10.18196/jrc.v6i5.27584.

[61] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of pso algorithm," *Applied Sciences*, vol. 12, no. 17, p. 8392, 2022, https://doi.org/10.3390/app12178392.

[62] M. A. Ahmad, M. Z. M. Tumari, and A. N. K. Nasir, "Composite fuzzy logic control approach to a flexible joint manipulator," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, 2013, https://doi.org/10.5772/52562.