A Differential Particle Scheme and its Application to PID Parameter Tuning of an Inverted Pendulum

Victor Parque
Department of Modern Mechanical Engineering
Waseda University
Tokyo, Japan
parque@aoni.waseda.jp

ABSTRACT

Gradient-free stochastic optimization algorithms are well-known for finding suitable parameter configurations over independent runs ubiquitously. Attaining low variability of convergence performance through independent runs is crucial to allow further generalization over distinct problem domains. This paper investigates the performance of a differential particle system in stabilizing a nonlinear inverted pendulum under diverse and challenging initial conditions. Compared to the relevant algorithms in the literature, our experiments show the feasibility of achieving lower convergence variability to stabilize a nonlinear pendulum over independent runs and initial conditions within a reasonable computational load.

CCS CONCEPTS

• **Computing methodologies → Continuous space search**; **Continuous models**; • **Applied computing → Computer-aided design**.

KEYWORDS

Differential Evolution, Particle Swarm, Optimization, PID Tunning, Inverted Pendulum, Nonlinear Control

ACM Reference Format:

Victor Parque. 2021. A Differential Particle Scheme and its Application to PID Parameter Tuning of an Inverted Pendulum. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3449726.3463225

1 INTRODUCTION

Proportional-Integral-Derivative (PID) is a widely known feedback loop mechanism to control industrial systems. With origins in the nineteenth century, tuning the gain parameters of a PID system is known to be challenging due to nonlinearities, which have a considerable effect on control performance adaptation to new conditions[1, 2, 4].

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463225

The most commonly used methods are the Ziegler-Nichols (Z-N) [37], with its origins in the 40's, and the Cohen-Coon[11] algorithms. However, as a matter of practice, the obtained parameters by these methods usually need further re-tuning to be used in industrial practice. For this reason, approaches based on optimization have emerged in the literature.

Gradient-free stochastic optimization schemes are robust and perform well in parameter tuning across independent runs, being Genetic Algorithms (GA) the most popular among evolutionary computing methods. One of the seminal studies is the work by Wang and Kwok[33, 34], showing its potential to deal with nonlinearities and appeal to industrial practice. Particle Swarm Optimization (PSO)[13] is another well-known algorithm that is inspired by swarm behavior. Recently, Chen et. al.[5] and Belkadi et. al.[3] studied the performance of PSO variants in tuning PID systems' parameters and recommended its use to achieve global optima and faster convergence. However, one of the critical challenges in PSO is how to avoid the low variability of convergence performance and high computational load over independent runs, which is crucial to allow the efficient generalization over different problem domains.

Generally speaking, high variability of convergence usually occurs when solutions *stagnate* in local optima, which occurs when generated solutions are not comparatively better than current satisfactory solutions. This problem occurs due to the inherent nature of PSO to sample the search space without considering the source of promising solutions and the inability to evaluate how successful sampled solutions are. Although the stagnation problem has called the attention of the mathematical benchmarking community under well-established synthetic functions[6, 8, 10, 12, 15–17, 29, 35, 36], rendering both generalization and adaptive parameter setups[9, 19, 30, 32](see the references therein), the study of stagnation in swarm-based systems for tuning of PID control tailored to inverted pendulum has received little attention. In this paper, inspired by stagnation-free schemes in swarm systems[10], we propose a mechanism combining the selection pressure and the archive of alternative solutions in a differential particle system[18], and evaluate its effectiveness in stabilizing a nonlinear inverted pendulum under diverse initial conditions. We study the effect of population size and weights on local and global interpolation vectors and compare its performance with relevant algorithms in the literature. Our computational experiments show that the feasibility to compute lower variability of convergence over independent runs within a reasonable computational load.

In the following sections, we describe our proposed approach, experiments and discuss our results.

2 PROPOSED APPROACH

In this section, we describe the central tenets of our approach.

2.1 Differential Particle Scheme (DPS)

Basically, we focus in tackling the well-known gradient-free optimization problem.

$$\underset{x}{\textbf{Minimize}}\ F(x) \text{ subject to } x \in \mathbb{X}, \tag{1}$$

in F denotes the cost function, $\mathbb{X}$ denotes the search space, and x denotes the search variable. We propose tackling (1) by iterative sampling, as follows

$$x_{t+1} = \begin{cases} u_t, \text{ if } F(u_t) \le F(x_t) \\ x_t, \text{ otherwise} \end{cases} \tag{2}$$

$$u_t = x_t^r + v_{t+1} \tag{3}$$

$$v_{t+1} = \omega v_t + c_1 r_1 (pbest_t - x_t^r) + c_2 r_2 (gbest_t - x_t^r) \tag{4}$$

$$x_t^r \in \begin{cases} \mathcal{P}_t, \text{if } q_t \le Q \\ \mathcal{A}_t, \text{otherwise} \end{cases} \tag{5}$$

$$\mathcal{A}_{t+1} = \begin{cases} \mathcal{P}_t, \text{if } t = 0 \\ \mathcal{A}_t \cup \{u_t\}, \text{if } t > 0 \text{ and } F(u_t) \le F(x_t) \end{cases} \tag{6}$$

$$q_{t+1} = \begin{cases} 0 \text{ if } F(u_t) \le F(x_t) \\ q_t + 1, \text{otherwise} \end{cases} \tag{7}$$

where

- t is the subscript considering iteration count;
- x_t is the solution at iteration t;
- v_t is the velocity vector at iteration t;
- c_1, c_2 are *acceleration* constants;
- r_1, r_2 are random numbers uniformly distributed at $U(0, 1)$;
- ω is the *inertia* weight;
- *pbest* is best solution up to iteration t;
- *gbest* is the best solution overall all the population;
- $\mathcal{P}$ is the population set ($x_t \in \mathcal{P}$),
- F is the cost function to be minimized.

Basically, the above is based population-based stochastic sampling of solutions to achieve global optima over the search space. The subscrips in (2) and (5) are inspired by the selection mechanisms of differential evolution[10, 28], whereas (4) is inspired by the behavior of particle swarms[13]. Also, in the above, the gradient of the cost function is unused, and solutions are updated by interpolating local vectors *pbest* and global vectors *gbest*. In terms of particle swarm, local vectors point towards the personal best, while global vectors point towards the global best in the population.

Furthermore,
q_t represents the accumulated count of unsuccessful solutions at iteration $t \in [T]$, u_t is the trial solution, Q is a user-defined stagnation threshold metric, and $\mathcal{A}$ is an archive (set) of recently updated solutions. Here, the initial configuration of the set $\mathcal{A}$ is a copy of the population for $t = 0$, and successful solutions replace oldest solutions, thus, $|\mathcal{A}| = |\mathcal{P}|$. A trial solution u_t is *successful* whenever $f(u_t) \le f(x_t)$ holds, thus the stagnation count q_t is set at zero whenever the successful trial u_t is *successful*.

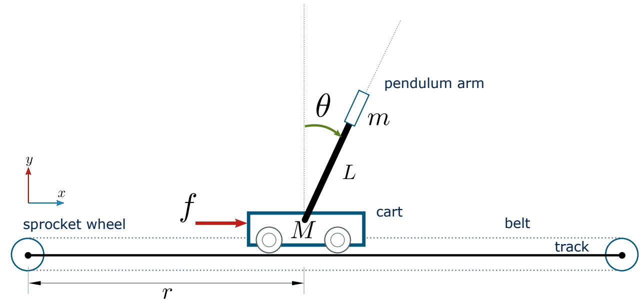


Figure 1: Basic concept of an inverted pendulum.

Figure 2: Inverted pendulum.

Generally speaking, the basic concept of our proposal includes integrating (1) the differential evolution *selection mechanism* in particle swarms, and (2) the *archive of successful vector referents*. The first concept considers the generation of trial vectors u_t and updating the position of sampled vectors according to how the current solutions are improved. The second concept considers the use of different sources to generate new trial vectors: in one hand, the population, and in other hand, an archive of successful trial vectors. By using the above described concepts, it becomes possible to sample new solutions from either in the population or from the archive of *successful solution trials*, in which the archive is updated whenever the *solution trials* lead to improved positions.

2.2 Inverted Pendulum

We aim to tackle the control of a nonlinear inverted pendulum as shown by Fig. 1 to reach stabilization at $\theta = 0$ given a starting position θ_o. The governing equations of the nonlinear pendulum model can be obtained from the Newton's second law:

$$(m + M)\ddot{r} + b\dot{r} + mL\ddot{\theta}cos\theta - mL\dot{\theta}^2 sin\theta = f, \tag{8}$$

$$(I + mL^2)\ddot{\theta} - mgLsin\theta + mL\ddot{r}cos\theta + d\dot{\theta} = 0, \tag{9}$$

where

- m is the pole mass,
- M is the cart mass,
- L is the pole length,
- g is the gravity constant,

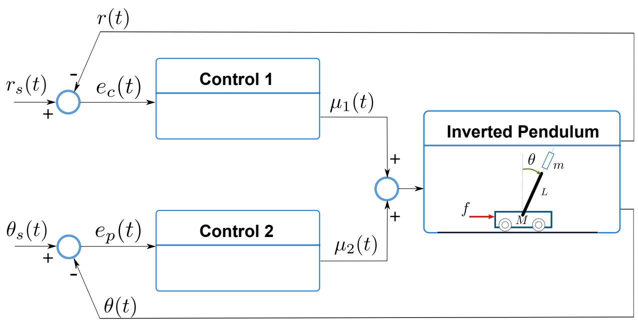


Figure 3: Basic block diagram of the control scheme.

- I is the mass moment of inertia of the pendulum,
- b is the cart friction coefficient,
- d is the pendulum damping coefficient,
- f is the dragging force,
- r is the cart position, and
- θ is the pendulum angle.

The above-described denote the essential elements of a track-enabled inverted pendulum system as Fig. 2 shows. Our objective is to stabilize the pendulum at $\theta_s = 0$ and cart position $r_s = 0$ given a starting position θ_o. For simplicity and without loss of generality, we design two independent PID controllers and sum their control outputs to render the control signal to the inverted pendulum as Fig. 3 shows. Thus, the continuous form of the PID controllers is given by the following:

$$\mu(t) = \mu_1(t) + \mu_2(t) \tag{10}$$

Thus the control signals are computed from the following

$$\mu_1(t) = k_p^c e_c(t) + k_i^c \int_0^t e_c(t)dt + k_d^c \frac{de_c(t)}{dt} \tag{11}$$

$$e_c(t) = r_s(t) - r(t) \tag{12}$$

, where $\mu_1(t)$ is the control signal derived from the error at the cart position, $e^c(t)$ represents the cart tracking error, k_p^c is the proportional gain, k_i^c is the integral gain, k_d^c is the derivative gain, $r_s(t)$ is the desired cart position, and $r(t)$ is the actual measurement.

$$\mu_2(t) = k_p^p e_p(t) + k_i^p \int_0^t e_p(t)dt + k_d^p \frac{de_p(t)}{dt} \tag{13}$$

$$e_p(t) = \theta_s(t) - \theta(t) \tag{14}$$

, where $\mu_2(t)$ is the control signal derived from the error in the pole position, $e^p(t)$ represents the pole tracking error, k_p^p is the proportional gain, k_i^p is the integral gain, k_d^p is the derivative gain, $\theta_s(t)$ is the desired pole position, and $\theta(t)$ is the actual measurement.

2.3 Encoding and Cost Function

The six gains related to the PID control gains in the cart and the pole become the search variables, and the cost function is defined by the sum of squared errors. Thus, the six-tuple consists of the following:

$$x = (k_p^c, k_i^c, k_d^c, k_p^p, k_i^p, k_d^p), \tag{15}$$

and the cost function is defined by the following:

$$F(x) = \int_0^T e_c^2(\tau)d\tau + \int_0^T e_p^2(\tau)d\tau \tag{16}$$

3 COMPUTATIONAL EXPERIMENTS

This section describes our computational experiments and obtained results in evaluating the effectiveness of our proposed approach.

3.1 Pendulum Settings

We implemented our algorithms using Matlab. We considered the following parameters

- mass of the inverted pendulum m= 0.23 kg
- cart mass M= 2.4 kg,
- L= 0.4 m,
- g= 9.81 m/s^2,
- cart friction coefficient b= 0.05 Ns/m,
- pendulum damping coefficient d= 0.005 Nms/rad,
- constant I= 0.099 kg m^2,

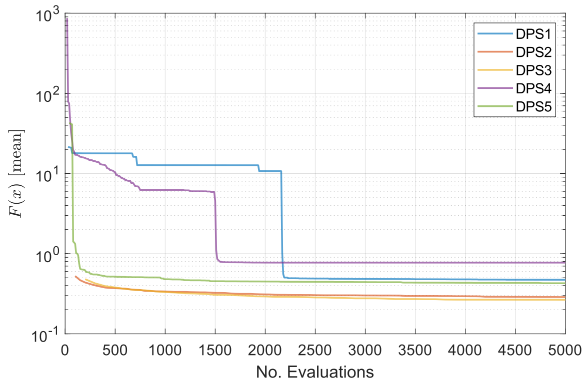
the above configuration is aligned to a real-world pendulum system as shown by Fig. 2. The desired cart position is set at r_o= 0 m. Differential equations of the inverted pendulum are solved numerically by using Runge-Kutta with simulation time T = 100 and step time at 0.01.

3.2 Initial Conditions

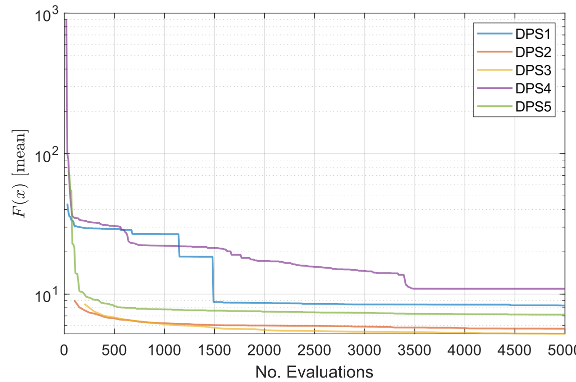
In a set of experiments, we first studied the effect of a relevant set of parameters on convergence behaviour over 30 independent runs. As such, we used $\mathcal{Q}$ = 200 and 5000 function evaluations. Here, the initial angle of the pendulum was set to $\theta_o = \{0.05, 0.2, 0.35, 0.5, 0.65\}$ rad, and evaluated the following variants:

- DPS1. Population size $|\mathcal{P}| = 10$, $\omega = 0.7$, $c_1 = 0.5$, $c_2 = 1$.
- DPS2. Population size $|\mathcal{P}| = 50$, $\omega = 0.7$, $c_1 = 0.5$, $c_2 = 1$.
- DPS3. Population size $|\mathcal{P}| = 100$, $\omega = 0.7$, $c_1 = 0.5$, $c_2 = 1$.
- DPS4. Population size $|\mathcal{P}| = 10$, $\omega = 0.2$, $c_1 = 0.5$, $c_2 = 1$.
- DPS5. Population size $|\mathcal{P}| = 10$, $\omega = 0.7$, $c_1 = 1.5$, $c_2 = 1$.

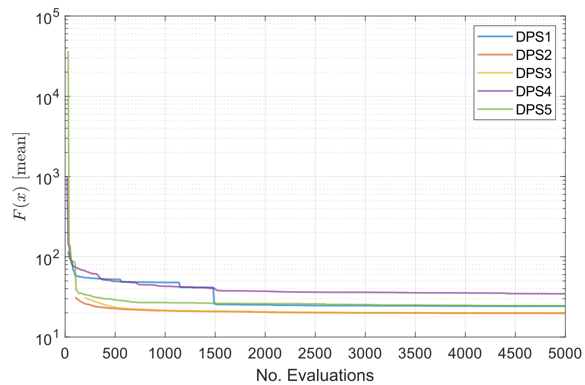
The above set aims to combine distinct population size and distinct weight on local and global interpolations. Fig. 4 shows the convergence behavior and the standard deviation of the convergence of the cost function over 30 independent runs. By observing Fig. 4, we can note that across different initial conditions of the pendulum, relatively high population sizes $|\mathcal{P}| = 50, 100$ are preferable for faster convergence. This observation is due to (4) using the interaction of the population to sample new solutions; thus, a higher number of solutions facilitates providing promising solutions in the search space. It is also possible to observe sudden decreases in the fitness function, in particular in Fig. 4-(a)-(c). We believe this phenomenon occurs due to the extended archive (6) of potential solutions, which enables providing a source of successful and promising solutions. Investigating a different set of population sampling schemes and archive formation is likely to improve the performance of our approach.



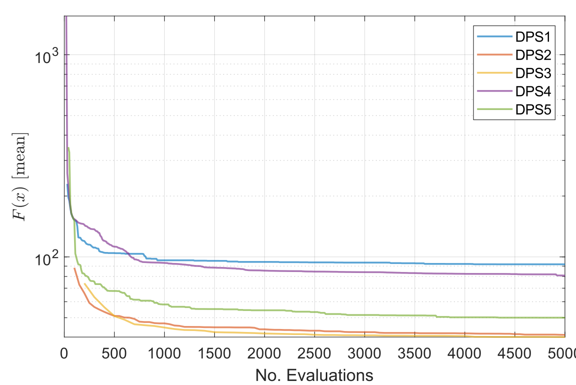
(a) $\theta_o = 0.05$ rad.



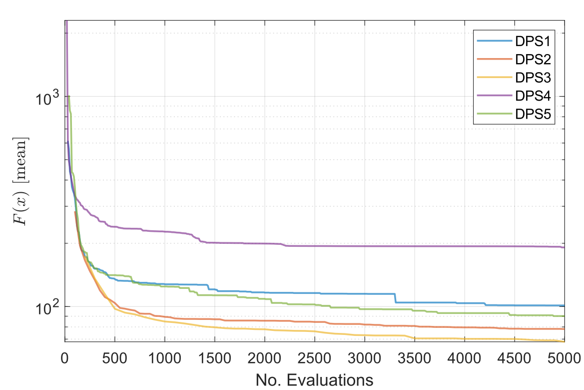
(b) $\theta_o = 0.2$ rad.



(c) $\theta_o = 0.35$ rad.



(d) $\theta_o = 0.5$ rad.



(e) $\theta_o = 0.65$ rad.

Figure 4: Mean convergence over 30 independent runs of our approach under different inverted pendulum conditions ad parameter schemes.

3.3 Algorithm Benchmarks

Since our proposed scheme extends the sampling schemes from Particle Swarm Optimization (PSO) and Differential Evolution (DE), we compared the performance of our proposed approach against the following related and well-known algorithms:

- Differential Evolution (DE)[7],
- Particle Swarm with Fitness Euclidean Ratio (FER)[14],
- Particle Swarm Optimization (PSO)[13],
- Rank-Based Differential Evolution (RBDE)[31],
- Strategy-Adaptation Differential Evolution (SADE)[27],

Performance validation and benchmarks with the recently published works in mathematical function optimization using the well-established datasets from CEC and GECCO from recent years are out of the scope of this paper. For experiments, we set the initial angle of the pendulum to $\theta_o = 0.4$ rad, which is reasonably challenging for any control system.

Fig. 5 shows an example of the converged stabilization behaviour, here variables such as cart position r, pendulum state θ and control signals u are plotted as a function of time. In this figure, the inverted pendulum stabilizes at around time step 30. The reader may note the quick transition from the initial conditions to the stabilization criteria at $\theta = 0$ rad.

Due to the stochastic nature of the algorithms mentioned above, we used 30 independent runs. Based on our observations in the previous section, parameters and settings in the algorithm were set as follows:

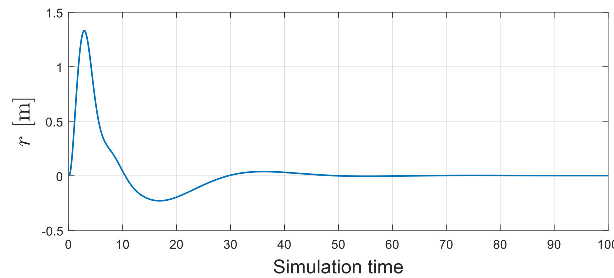
- The proposed algorithm used the following parameters $\omega = 0.7$, $c_1 = 0.5$, $c_2 = 1$, population size $|\mathcal{P}| = 50$, $Q = 200$, and 200 generations.
- Particle Swarm with Fitness Euclidean Ratio (FER)[14] and Particle Swarm Optimization (PSO) used similar parameters and scaling factors.
- Differential Evolution (DE) and Rank-Based Differential Evolution (RBDE)[31] used the DE-current-to-best/2 mutation strategy with scaling factors $F_1 = F_2 = 1.5$ and crossover probability $CR = 0.5$.
- The coefficient β involved in the Whitley distribution scheme in Rank-Based Differential Evolution (RBDE) is set as $\beta = 2$.
- The Strategy-Adaptation Differential Evolution (SADE)[27] used the default 4 mutation strategies: (1) DE/rand/1, (2) DE/rand/2, (3) DE/current-to-rand/2 and (4) DE/rand-to-best/, each of which used the default parameters $\mu_F = \mu_{CR} = 0.5$ and $\sigma = 0.1$ at initialization.

Our crucial motivation behind using the above-described set is to evaluate the convergence under a restricted computational budget, which implies the responsiveness for real-time tasks and the fair comparison with relevant schemes in the literature. Fine-tuning the above-described parameters is out of the scope of this paper and left for future work in our agenda.

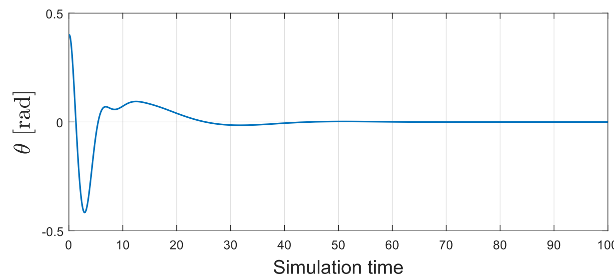
3.4 Comparison Results

Fig. 6 shows the convergence behaviour and standard deviation of the convergence of the cost function over 30 independent runs. Also, Fig. 7 shows the statistical significance test and p-values based on the Wilcoxon rank-sum test at 5% significance level. Cells with '+/=/-' denote instances where an algorithm in the row is significantly better/similar/worse to an algorithm in the column. By observing at the results of Fig. 6 and Fig. 7, we observe the following facts:

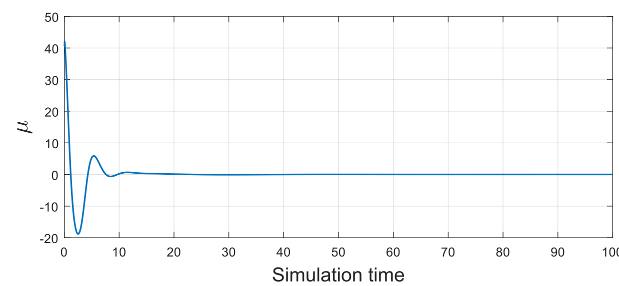
- The y-axis of both Fig. 6 and Fig. 7 are in log-scale, and as such, our proposed approach offers similar performance compared to advanced sampling mechanisms such as the multiple strategies with adaptive parameter tuning of SADE and the implicit neighborhood formation in FERPSO.
- The best performance can be obtained in about 100 generations, which is relatively fast for tasks requiring real-time performance .



(a) Cart Position



(b) Pendulum State

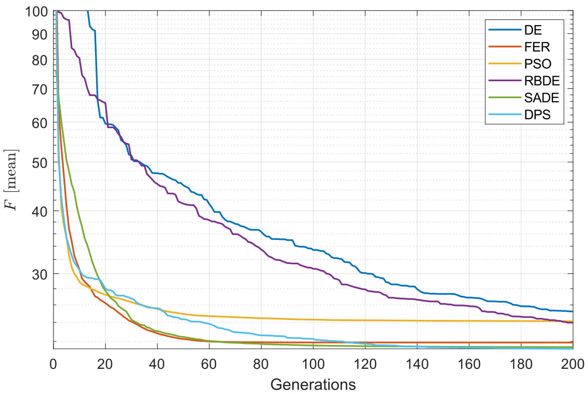


(c) Control

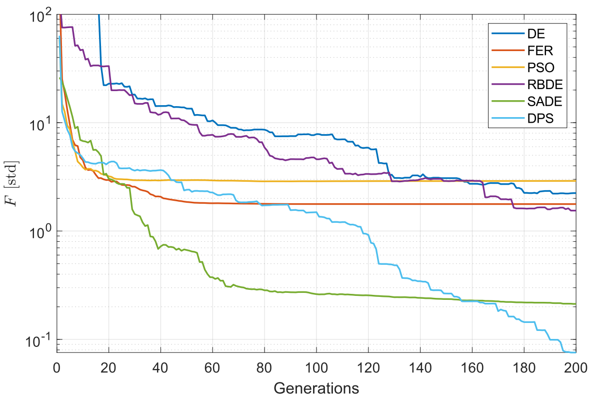
Figure 5: Cart position, pendulum state and control as a function of time.

- Compared to the conventional frameworks, our proposed approach decreased the standard deviation over independent runs consistently. This observation implies reaching the optimal solutions under independent and distinct initialization conditions.
- We can confirm that the proposed approach avoids stagnation at early iterations, whereas the standard PSO portrays higher variability of converged solutions, implying that stagnation occurs despite having independent runs. We believe the reason of avoiding stagnation is due to the ability of our approach to include a selection mechanism as well as the archive of potential referential vectors, which enables to consider successful trial vectors whenever the accumulated stagnation counter achieves higher values.

The above observations confirm the feasibility of using a reasonable computational load to achieve competitive fitness convergence and low variability of convergence over independent runs in challenging stabilization problems of nonlinear inverted pendulum systems. This fact implies the possibility of enhancing the performance by adaptive sampling schemes[19, 20, 23] and knowledge



(a) Mean



(b) Standard Deviation

Figure 6: Mean and standard deviation of convergence behaviour of the evaluated algorithms over 30 independent runs.

	DE	FER	PSO	RBDE	SADE	**DPS**
DE		- 1.20e-08	- 1.03e-02	- 1.44e-02	- 3.02e-11	- 3.02e-11
FER	+ 1.20e-08		+ 7.09e-08	+ 3.08e-08	+ 9.51e-06	1.03e-02
PSO	+ 1.03e-02	- 7.09e-08		= 1.71e-01	- 2.39e-08	- 2.37e-10
RBDE	+ 1.44e-02	- 3.08e-08	= 1.71e-01		- 3.02e-11	- 3.02e-11
SADE	+ 3.02e-11	- 9.51e-06	+ 2.39e-08	+ 3.02e-11		- 1.73e-06
DPS	+ **3.02e-11**	+ **1.03e-02**	+ **2.37e-10**	+ **3.02e-11**	+ **1.73e-06**	

Figure 7: Statistical significance and p-values based on the Wilcoxon rank-sum test at 5% significance level. Cells with '+/=/-' denote instances where an algorithm in the row is significantly better/similar/worse to an algorithm in the column.

acquisition through graphs[22, 25, 26]. We believe our results offer new possibilities to build stagnation-free nature-inspired PID tuning algorithms useful for control and industrial practice. Investigating the performance over nonlinear control problems, such as robot and vehicle control[2, 21, 24], and mathematical function optimization from the relevant literature is our agenda.

4 CONCLUSION

We have proposed a new gradient-free optimization algorithm based on a particle swarm scheme with an archive-based selection mechanism and evaluated its effectiveness in stabilizing a nonlinear inverted pendulum under challenging conditions. Our results show the efficiency of achieving competitive fitness convergence and low variability of performance over independent experiment runs in challenging stabilization conditions of a nonlinear inverted pendulum system. Our future work aims at evaluating adaptive forms of sampling over a large number of nonlinear control problems.

ACKNOWLEDGEMENT

This research was supported by JSPS KAKENHI 20K11998.

REFERENCES

[1] Mohamed Abdelwahab, Victor Parque, A. A. Abouelsoud, and Ahmed M. R. Fath El-Bab. 2021. Navigation of Omni-Directional Mobile Robot in Unstructured Environments using Fuzzy Logic Control. In *IEEE/SICE International Symposium on System Integration, SII 2021, Iwaki, Japan, January 11-14, 2021*. IEEE, 684–689.
[2] Mohamed Abdelwahab, Victor Parque, Ahmed M. R. Fath El-Bab, A. A. Abouelsoud, and Shigeki Sugano. 2020. Trajectory Tracking of Wheeled Mobile Robots Using Z-Number Based Fuzzy Logic. *IEEE Access* 8 (2020), 18426–18441. https://doi.org/10.1109/ACCESS.2020.2968421
[3] A. Belkadi, H. Oulhadj, Y. Touati, Safdar A. Khan, and B. Daachi. 2017. On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. *Applied Soft Computing* 60 (2017), 87 – 100.
[4] Maghade D.K. Sondkar S.Y. et al. Borase, R.P. 2020. A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control* (2020).
[5] J. Chen, M. N. Omidvar, M. Azad, and X. Yao. 2017. Knowledge-based particle swarm optimization for PID controller tuning. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. 1819–1826.
[6] Laizhong Cui, Qiuling Huang, Genghui Li, Shu Yang, Zhong Ming, Zhenkun Wen, Nan Lu, and Jian Lu. 2018. Differential Evolution Algorithm With Tracking Mechanism and Backtracking Mechanism. *IEEE Access* 6 (2018), 44252–44267. https://doi.org/10.1109/ACCESS.2018.2864324
[7] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar. 2009. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation* 13, 3 (June 2009), 526–553.
[8] LiBao Deng, Haili Sun, and Chunlei Li. 2021. JDF-DE: a differential evolution with Jrand number decreasing mechanism and feedback guide technique for global numerical optimization. *Applied Intelligence* 51, 1 (2021), 359–376.
[9] Dan Guo, Xilu Wang, Kailai Gao, Yaochu Jin, Jinliang Ding, and Tianyou Chai. 2021. Evolutionary Optimization of High-Dimensional Multiobjective and Many-Objective Expensive Problems Assisted by a Dropout Neural Network. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021), 1–14. https://doi.org/10.1109/TSMC.2020.3044418
[10] S. Guo, C. Yang, P. Hsu, and J. S. . Tsai. 2015. Improving Differential Evolution With a Successful-Parent-Selecting Framework. *IEEE Transactions on Evolutionary Computation* 19, 5 (2015), 717–730.
[11] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang. 1996. Performance and gain and phase margins of well-known PID tuning formulas. *IEEE Transactions on Control Systems Technology* 4, 4 (1996), 473–477.
[12] Xinfang Ji, Yong Zhang, Dunwei Gong, and Xiaoyan Sun. 2021. Dual-Surrogate Assisted Cooperative Particle Swarm Optimization for Expensive Multimodal Problems. *IEEE Transactions on Evolutionary Computation* (2021), 1–1.
[13] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, Vol. 4. 1942–1948 vol.4.
[14] Xiaodong Li. 2007. A Multimodal Particle Swarm Optimizer Based on Fitness Euclidean-Distance Ratio. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (London, England) *(GECCO '07)*. Association for Computing Machinery, New York, NY, USA, 78–85.

[15] Zhenyu Meng, Yuxin Chen, and Xiaoqing Li. 2020. Enhancing Differential Evolution With Novel Parameter Control. *IEEE Access* 8 (2020), 51145–51167. https://doi.org/10.1109/ACCESS.2020.2979738

[16] Kun Miao and Ziyang Wang. 2019. Neighbor-Induction and Population-Dispersion in Differential Evolution Algorithm. *IEEE Access* 7 (2019), 146358–146378. https://doi.org/10.1109/ACCESS.2019.2945831

[17] Jeng-Shyang Pan, Nengxian Liu, and Shu-Chuan Chu. 2020. A Hybrid Differential Evolution Algorithm and Its Application in Unmanned Combat Aerial Vehicle Path Planning. *IEEE Access* 8 (2020), 17691–17712. https://doi.org/10.1109/ACCESS.2020.2968119

[18] V. Parque. 2021. A Differential Particle Scheme with Successful Parent Selection and its Application to PID Control Tuning. In *2021 IEEE Congress on Evolutionary Computation (CEC)*.

[19] Victor Parque. 2021. On Hybrid Heuristics for Steiner Trees on the Plane with Obstacles. In *Evolutionary Computation in Combinatorial Optimization - 21st European Conference, EvoCOP 2021 (Lecture Notes in Computer Science, Vol. 12692)*. Springer, 120–135.

[20] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. 2013. Reinforced Explorit on Optimizing Vehicle Powertrains. In *Neural Information Processing ICONIP (Lecture Notes in Computer Science, Vol. 8227)*. Springer, 579–586. https://doi.org/10.1007/978-3-642-42042-9_72

[21] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. 2013. Reinforced Explorit on Optimizing Vehicle Powertrains. In *20th International Conference, ICONIP (Lecture Notes in Computer Science, Vol. 8227)*. Springer, 579–586. https://doi.org/10.1007/978-3-642-42042-9_72

[22] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. 2014. Bijections for the numeric representation of labeled graphs. In *2014 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2014, San Diego, CA, USA, October 5-8, 2014*. IEEE, 447–452. https://doi.org/10.1109/SMC.2014.6973948

[23] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. 2014. Neural Computing with Concurrent Synchrony. In *Neural Information Processing - ICONIP (Lecture Notes in Computer Science, Vol. 8834)*. Springer, 304–311.

[24] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. 2014. Neural Computing with Concurrent Synchrony. In *Neural Information Processing ICONIP (Lecture Notes in Computer Science, Vol. 8834)*. Springer, 304–311. https://doi.org/10.1007/978-3-319-12637-1_38

[25] Victor Parque and Tomoyuki Miyashita. 2017. On succinct representation of directed graphs. In *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017, Jeju Island, South Korea, February 13-16, 2017*. IEEE, 199–205. https://doi.org/10.1109/BIGCOMP.2017.7881738

[26] Victor Parque and Tomoyuki Miyashita. 2017. On the Numerical Representation of Labeled Graphs with Self-Loops. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*. IEEE Computer Society, 342–349. https://doi.org/10.1109/ICTAI.2017.00061

[27] A. K. Qin, V. L. Huang, and P. N. Suganthan. 2009. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation* 13, 2 (2009), 398–417.

[28] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* 11, 4 (01 Dec 1997), 341–359.

[29] Gaoji Sun, Chunlei Li, and Libao Deng. 2021. An adaptive regeneration framework based on search space adjustment for differential evolution. *Neural Computing and Applications* (2021).

[30] Jianyong Sun, Xin Liu, Thomas Bäck, and Zongben Xu. 2021. Learning Adaptive Differential Evolution Algorithm from Optimization Experiences by Policy Gradient. *IEEE Transactions on Evolutionary Computation* (2021), 1–1. https://doi.org/10.1109/TEVC.2021.3060811

[31] Andrew M. Sutton, Monte Lunacek, and L. Darrell Whitley. 2007. Differential Evolution and Non-Separability: Using Selective Pressure to Focus Search. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (London, England) *(GECCO '07)*. Association for Computing Machinery, New York, NY, USA, 1428–1435.

[32] Ke Tang, Shengcai Liu, Peng Yang, and Xin Yao. 2021. Few-shots Parallel Algorithm Portfolio Construction via Co-evolution. *IEEE Transactions on Evolutionary Computation* (2021), 1–1. https://doi.org/10.1109/TEVC.2021.3059661

[33] P. Wang and D.P. Kwok. 1994. Optimal design of PID process controllers based on genetic algorithms. *Control Engineering Practice* 2, 4 (1994), 641 – 648.

[34] P. Wang and D. P. Kwok. 1992. Auto-tuning of classical PID controllers using an advanced genetic algorithm. In *Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation*. 1224–1229 vol.3.

[35] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades – Part A. *IEEE Transactions on Evolutionary Computation* (2021), 1–1. https://doi.org/10.1109/TEVC.2021.3060014

[36] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades – Part B. *IEEE Transactions on Evolutionary Computation* (2021), 1–1. https://doi.org/10.1109/TEVC.2021.3060012

[37] J. G. Ziegler and N. B. Nichols. 1993. Optimum Settings for Automatic Controllers. *Journal of Dynamic Systems, Measurement, and Control* 115, 2B (06 1993), 220–222.