Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.11 n.53, p. 186-192. May/June., 2025.
DOI: https://doi.org/10.5935/jetia. v11i53.1672

RESEARCH ARTICLE OPEN ACCESS

OPTIMIZED PID TUNING IN LONGITUDINAL CONTROL OF ELECTRIC AUTONOMOUS VEHICLES: A COMPARATIVE STUDY OF JELLYFISH SEARCH AND GENETIC ALGORITHM

Asmaa Guendouz[1], Mustapha Hatti[2], Abdelhalim Tlemçani[3]

[1,3]Electrical Engineering and Automation Research Laboratory (LREA), University of Medea, Medea, Algeria.
[2]UDES, Unit for the Development of Solar Equipment, BP 386, 42004 Bou Ismail, Tipaza, Algeria.

[1]http://orcid.org/0009-0006-9260-833X, [2]http://orcid.org/0000-0002-8645-207X, [3]http://orcid.org/0000-0002-4385-671X

Email: guendouz.asma@univ-medea.dz, mustapha.hatti@ieee.org, tlemcani.abdelhalim@univ-medea.dz

ARTICLE INFO

Article History
Received: February 28, 2025
Revised: May 20, 2025
Accepted: June 15, 2025
Published: June 30, 2025

Keywords:
Metaheuristic Algorithms,
Electric Autonomous Vehicles,
Jellyfish Search optimizer,
Genetic Algorithms,
PID based Longitudinal Control.

ABSTRACT

Tuning PID controllers to enhance longitudinal control and speed planning of Electric Autonomous Vehicles is a challenge, which can be effectively addressed by evolving metaheuristic algorithms. This paper evaluates the performance of the Jellyfish Search (JS) optimizer and Genetic Algorithms (GA) against conventional PID tuning methods in longitudinal control systems. A modified objective function combining Time-weighted Absolute Error (ITAE) and Integral Square Error (ISE) is proposed based on the weighted sum method, aiming to balance performance metrics and overcome the limitations of conventional objective functions. This function is optimized by both genetic and jellyfish techniques. The simulations are conducted through realistic scenarios in accordance with road safety standards, using a sinusoidal profile as speed reference. The results demonstrate the effectiveness of both GA and JS in outperforming conventional PID, achieving zero overshoot, reduced settling times, and lower steady-state errors. Observably, JS optimizer exhibits a slight advantage over GA in the overall performance, especially fast convergence. These outcomes validate the contribution of the proposed approaches in enhancing the field of autonomous driving.

I. INTRODUCTION

Autonomous driving has been the subject of significant research and development for many years to this day. The control system consists of lateral and longitudinal control. This work presents a contribution in longitudinal control for Electric Autonomous Vehicles (EAVs), specifically addressing optimized techniques and speed profiles to ensure safety, comfort and efficiency. A wide array of control strategies has been employed for longitudinal control, including PID control [1]. However, tuning PID gains is a challenging task, as traditional approaches such as trial-and-error methods often fall short. This matter is addressed via optimization-based tuning techniques such as metaheuristic algorithms [2]. In the literature, the Integral Absolute, Square, Time-weighted Squared, and Time-weighted Absolute Errors (IAE), (ISE), (ITSE), and (ITAE), are generally the most used metrics for the cost function [3],[4], which can give suboptimal performance in certain scenarios.

This paper proposes a combination of these performance metrics to build the cost function based on the weighted sum method [5], where ITAE and ISE are combined to build a robust objective function to guarantee a better tuning and high performance. The weights are selected through observation of the output's behavior and performance, including overshoot, settling time and steady-state error. Moreover, the two optimization techniques used to optimize the selected cost function are Genetic Algorithms (GAs) and the Jellyfish Search (JS) optimizer.

GAs are widely used in several applications, being a class of evolutionary algorithms inspired by the principle of natural selection, their contribution is proven effective, in particular in control systems and PID tuning [6]. JS optimizer is a recently developed nature-inspired optimization algorithm that mimics the motion patterns of jellyfish in the sea. It balances exploration and exploitation to find optimal solutions using passive and active motions. This algorithm demonstrates high performance in certain optimization tasks due to its dynamic adaptation and enhanced global search capability [7].

The configuration of this paper is arranged as follows. The speed control system is presented in section **II**. Section **III** discusses the optimization methods. In section **IV**, simulation results and discussions are given to validate the proposed methods, followed by a conclusion.

II. SYSTEM DESCRIPTION

In this section, the design of the speed control system for an EAV is introduced. The system employs a direct current (DC) motor as the speed actuator and a Proportional-Integral-Derivative (PID) as the main controller.

II.1 DC MOTOR MODELLING

The DC motor is described by the equations given below [8].

DC motor Equations

$$V = R_a \cdot i_a + e + L_a \cdot \frac{di_a}{dt} \tag{1}$$

$$e = k \cdot \Omega \tag{2}$$

$$J \cdot \frac{d\Omega}{dt} = T_{em} - f \cdot \Omega - T_L \tag{3}$$

$$T_{em} = k_{em} \cdot i_a \tag{4}$$

Where, V is the input voltage, e is the back electromotive force (EMF), i_a is the current in the motor winding, T_{em} is the electromagnetic torque, and Ω is the angular velocity.

The parameters of the DC motor are given in Table 1. These latter are selected to approximate the characteristics of a Tesla Model S, with a torque of $T_L = 430Nm$, as specified in the catalog provided in [9].

Table 1: DC Motor Parameters.

Parameter	**value**
Armature resistance R_a	0.193 Ω
Armature inductance L_a	0.00383 H
Back EMF constant k	2.332232 $V.s$
Torque constant k_{em}	2.1717 $N.m/A$
Moment of inertia J	0.6 $kg \cdot m^2$
Coefficient of friction f	2.632177 $N.m.s$

Source: Authors, (2025).

To simulate the motor's performance, it has been modeled in MATLAB Simulink, as shown in Figure 1.

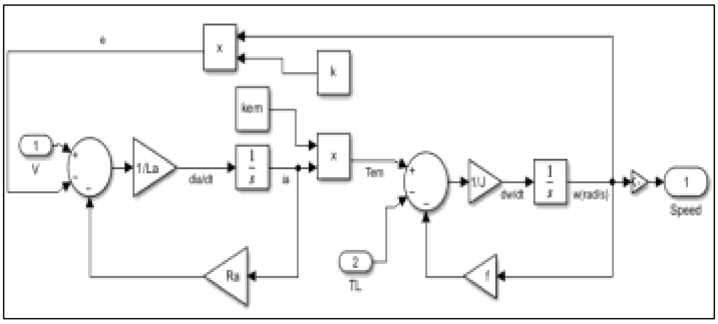

Figure 1: DC motor block diagram.
Source: Authors, (2025).

II.2 PID CONTROLLER

PID controllers are the basis of control systems, they provide a simple yet powerful framework. The transfer function describing a PID controller is given in Equation 5. K_P, K_I, and K_D are the proportional, integral, and derivative gains, respectively [10].

PID transfer function

$$G(s) = K_P + \frac{K_I}{s} + K_D.s \tag{5}$$

The performance of this controller is based on the selection of its gains. However, the outcome of the conventional methods is limited and often demonstrates inadequate disturbance rejection, which highlights the need for more advanced optimization techniques.

III. OPTIMIZATION METHODS

III.1 OBJECTIVE FUNCTION

In order to satisfy the performance criteria, including minimal overshoot, rapid settling time, and low steady-state error. The optimization process focuses on minimizing an objective function (cost function), denoted f, a mathematical expression that quantifies the system's performance for optimization. This latter is formulated based on the performance metrics. IAE, ISE, ITSE, and ITAE [4].

The Integral Absolute Error

$$IAE = \int |e(t)|dt \tag{6}$$

Integral Square Error

$$ISE = \int e^2(t)dt \tag{7}$$

Integral Time-weighted Absolute Error

$$ITAE = \int t|e(t)|dt \tag{8}$$

Integral Time-weighted Squared Error

$$ITSE = \int t\, e^2(t)dt \tag{9}$$

The selection of the cost function in this study is based on the weighted sum method [5], by combining ITAE and ISE. Where $w_1 = 0.2$ and $w_2 = 0.8$. These latter are selected to emphasize steady-state performance, while ensuring reasonable transient performance [11].

Objective function

$$f = w_1 \cdot ITAE + w_2 \cdot ISE \tag{10}$$

III.2 GENETIC ALGORITHM

Genetic Algorithms (GAs) are metaheuristic algorithms inspired by the biological evolution process that employ an intelligent exploitation of a stochastic search. These algorithms evolve a group of potential solutions across multiple generations through selection, recombination, and mutation. [12]. The mechanism's process is illustrated in Figure 2.

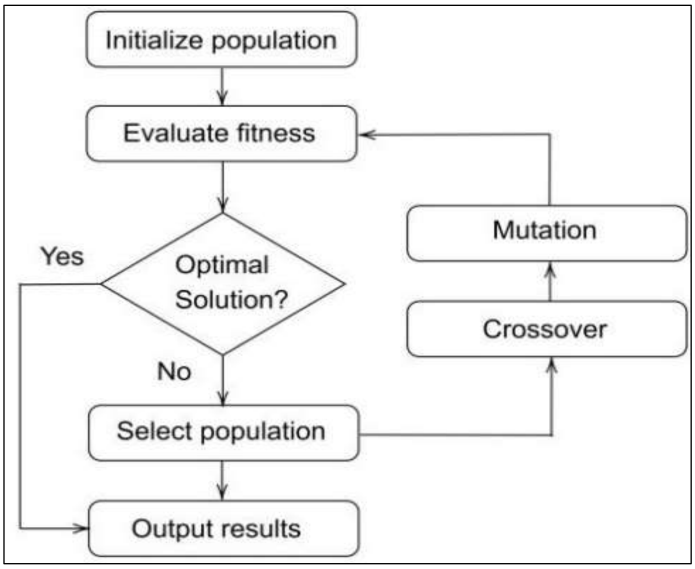

Figure 2: GA process flowchart.
Source:[13].

III.3 JELLYFISH SEARCH OPTIMIZER

The Jellyfish Search (JS) optimizer is a recent metaheuristic optimization algorithm developed by Jui-Sheng Chou et al. [14]. It is built based on an inspiration of the behavior of jellyfish food-seeking process in the ocean. The first step involves following the ocean current to find regions rich in food resources (optimal solutions), this step is called the exploration phase. Over time, a jellyfish swarm is created, and each jellyfish pursues the motion within the swarm to discover a more optimal location, which is translated as active and passive motions, representing the exploitation phase. Meanwhile, time control is introduced to identify the varied motion types and regulate the transition, which makes the optimum phase [15]. The overall process is illustrated in Figure 3.

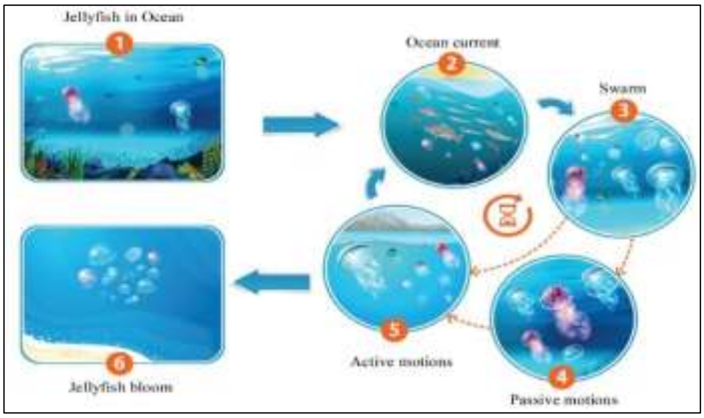

Figure 3: Jellyfish behavior process.
Source: [14].

The mechanism process of the Jellyfish Search (JS) optimizer is governed by a series of mathematical equations given below. These latter are detailed in [14].

Trend Calculation

$$\overrightarrow{tr} = Z^{*} - \beta \times \mathrm{rand}(0,1) \times \mu \qquad (11)$$

Active Motion Update

$$Z_i(t+1) = Z_i(t) + \mathrm{rand}(0,1) \times (Z^{*} - \beta \times \mathrm{rand}(0,1) \times \mu) \qquad (12)$$

Passive Motion Update

$$Z_i(t+1) = Z_i(t) + \gamma \times \mathrm{rand}(0,1) \times (U_b - L_b) \qquad (13)$$

Direction Calculation

$$\overrightarrow{\mathrm{Dr}} = \begin{cases} Z_j(t) - Z_i(t), & \text{if } f(Z_i) \geq f(Z_j) \\ Z_i(t) - Z_j(t), & \text{if } f(Z_i) < f(Z_j) \end{cases} \qquad (14)$$

Position Update

$$Z_i(t+1) = Z_i(t) + \overrightarrow{\mathrm{Step}} \qquad (15)$$

$$\overrightarrow{\mathrm{Step}} = \mathrm{rand}(0,1) \times \overrightarrow{\mathrm{Dr}} \qquad (16)$$

Time Control Function

$$c(t) = \left| \left(1 - \frac{t}{\mathrm{I}_{\max}}\right) \times (2 \times \mathrm{rand}(0,1) - 1) \right| \qquad (17)$$

Where $\overrightarrow{tr}$ is the direction of the ocean current, $Z_i(t)$ is the location of the i^{th} jellyfish at time t, U_b *and* L_b are upper and lower bounds, β is a distribution coefficient, μ is the mean location of all jellyfish, γ is a motion coefficient, and $\mathrm{I}_{\max}$ is the maximum number of iterations.

Algorithm 1 represents the pseudocode of the JS optimization based PID tuning. The corresponding flowchart is illustrated in Figure 4.

```
Algorithm 1: JS PID tuning pseudo-code
Begin
Intialize PID parameters
Objective function f(Z) calculation (Equation 10)
Set: population size $n_{pop}$=100
     maximum iteration $I_{max}$=50
Initialization of the population of jellyfish $Z_i$ (i=1, 2, ..., $n_{pop}$)
through logistic chaotic map
Calculation of the quantity of food at each $Z_i$, f ($Z_i$)
Search of jellyfish at current location with most food (Z*)
Initialize time: t = 1
Repeat
    For i = 1: $n_{pop}$ do
    Calculation of the time control function c(t) (Equation 17)
    If c(t) ≥ 0.5:
        Jellyfish pursuit of the ocean current
        1. Determination of the ocean current (Equation 11)
        2. New location calculation (Equation 12)
    Else:  Jellyfish motion inside a swarm
        If rand (0,1) > (1 - c(t)):  Passive motions
                                     Location update by Equation 13
        Else:  Jellyfish exhibits Active motions
            1. Calculation of the direction of jellyfish (Equation 14)
            2. Location of jellyfish is updated by Equation 15
        End If
    End If
    Boundary check and calculation of the quantity of food at new
    location
    Location update ($Z_i$, Z*)
    End For
    Time Update: t ++
Until stop criterion is met
Output: convergence iteration
        the best results: K_P, K_I, and K_D
        the best optimal value of the objective function
End
```

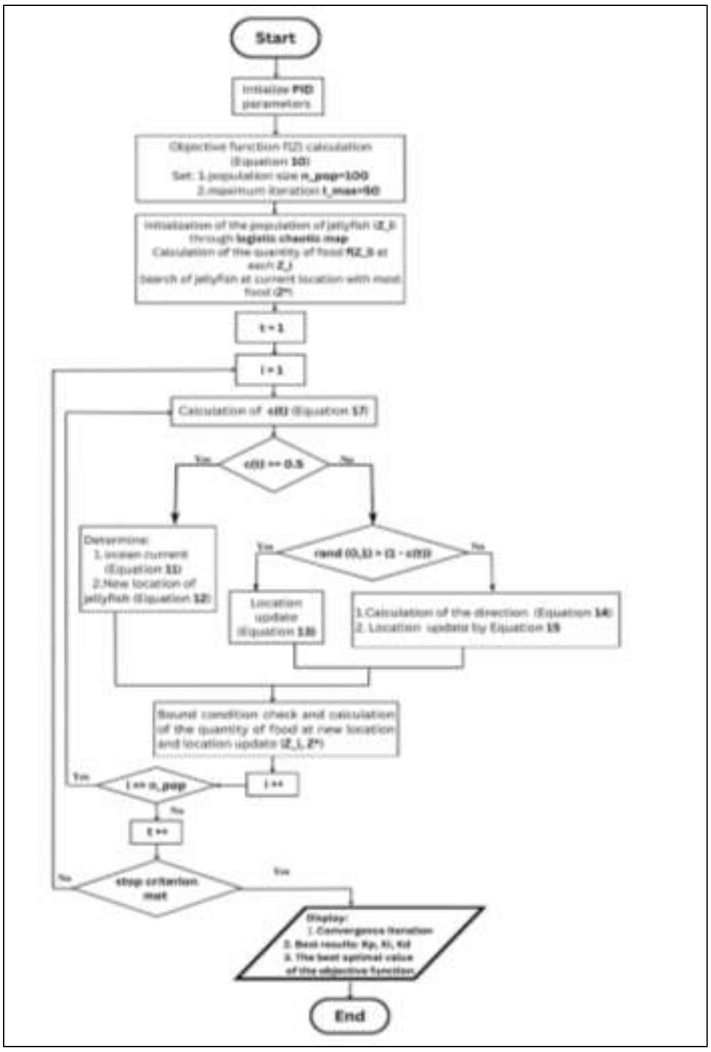

Figure 4: JS Flowchart for PID tuning.
Source: Authors, (2025).

IV. RESULTS AND DISCUSSIONS

The designed PID longitudinal control system is evaluated on a 2 km road segment. Initially, the PID gains are determined using the conventional trial-and-error method. The performance of this approach is then compared to the optimized PID gains via the Jellyfish Search (JS) optimizer and Genetic Algorithm (GA). The overall control system is illustrated in Figure 5.

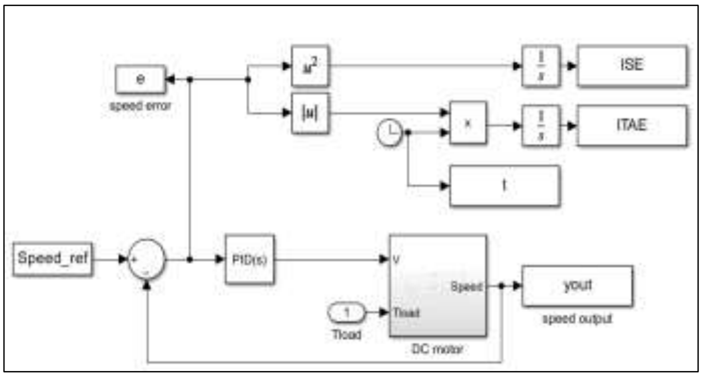

Figure 5: PID longitudinal control system.
Source: Authors, (2025).

The speed input selected for this scenario is the optimized sinusoidal profile (Figure 6), characterized by the following parameters:

A speed limit V_m of $28.8\,km/h\,(8m/s)$, suitable for urban environments as per the ISO 2631 standard**.** An acceleration limit, $A_m = 0.4\,m/s^2$ chosen based on the ISO 22737 standard**.** This speed input considers urban travel whilst maintaining safety, smooth driving, and passenger comfort.

The initialization parameters for both GA and JS optimizer are given in Tables 2 and 3 respectively.

Table 2: GA Parameters.

Parameter	**value**
Number of variables	3
Lower bounds	[0,0,0]
Upper bounds	[20, 500, 10]
Max generation	50
Population size	100
Crossover fraction	0.8
Mutation	Adaptive feasable mutation

Source: Authors, (2025).

Table 3: JS Parameters.

Parameter	**value**
dimensions	3
Lower bounds	[0,0,0]
Upper bounds	[20, 500, 10]
Max Iterations	50
Population size	100
Stopping Condition	Convergence Check
Initialization Method	Logistic Map

Source: Authors, (2025).

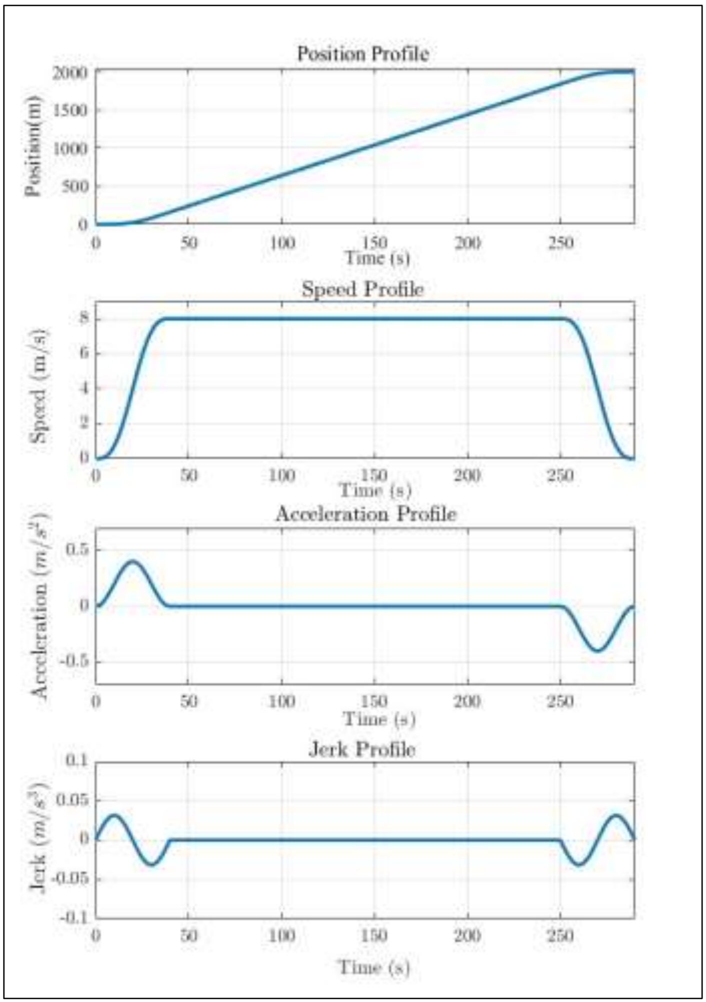

Figure 6: Sinusoidal profiles.
Source: Authors, (2025).

The GA corresponding results are illustrated in Figure 7. The statistical indices of the algorithm's outcome are given in Table 4.

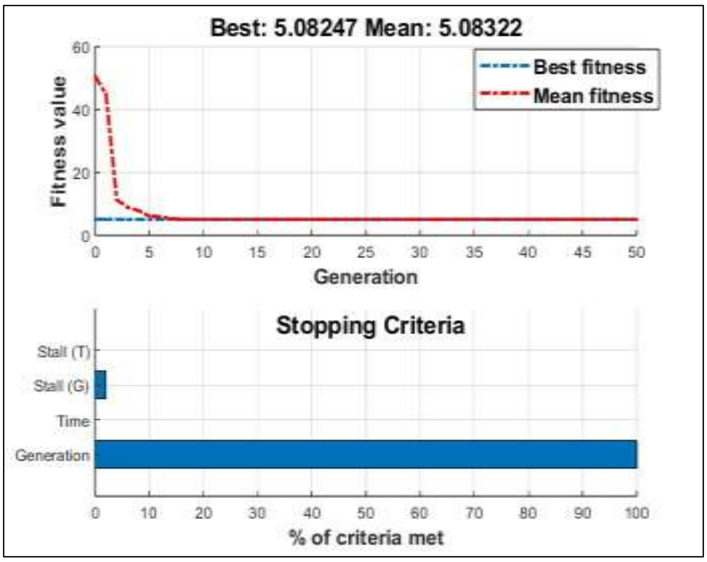

Figure 7: Genetic algorithm results.
Source: Authors, (2025).

Table 4: Progress of the Genetic Algorithm.

Generation	Func-count	Best f(z)	Mean f(z)	Stall Generations
1	200	5.103	44.930	0
2	295	5.094	11.300	0
3	390	5.092	8.905	0
4	485	5.092	7.892	0
5	580	5.092	6.107	1
6	675	5.087	5.937	0
7	770	5.087	5.328	0
8	865	5.087	5.097	0
9	960	5.087	5.095	0
10	1055	5.087	5.094	0
11	1150	5.087	5.093	0
12	1245	5.087	5.093	1
13	1340	5.084	5.092	0
14	1435	5.084	5.091	1
15	1530	5.084	5.087	0
16	1625	5.084	5.087	1
17	1720	5.084	5.086	2
18	1815	5.083	5.086	0
19	1910	5.083	5.086	0
20	2005	5.083	5.086	1
21	2100	5.083	5.085	0
22	2195	5.083	5.084	1
23	2290	5.083	5.084	2
24	2385	5.083	5.083	3
25	2480	5.083	5.083	0
26	2575	5.083	5.083	0
27	2670	5.083	5.083	0
28	2765	5.083	5.084	1
29	2860	5.083	5.083	2
30	2955	5.083	5.083	3
31	3050	5.083	5.083	4
32	3145	5.083	5.083	0
33	3240	5.083	5.083	0
34	3335	5.083	5.083	1
35	3430	5.083	5.083	2
36	3525	5.083	5.083	3
37	3620	5.083	5.083	0
38	3715	5.083	5.083	1
39	3810	**5.082**	5.083	0
40	3905	5.082	5.083	1
41	4000	5.082	5.083	2
42	4095	5.082	5.083	0
43	4190	5.082	5.083	0
44	4285	5.082	5.083	0
45	4380	5.082	5.083	0
46	4475	5.082	5.083	0
47	4570	5.082	5.083	0
48	4665	5.082	5.083	0
49	4760	5.082	5.083	0
50	4855	5.082	5.083	1

Source: Authors, (2025).

The results obtained from the Jellyfish Search (JS) algorithm demonstrate its effectiveness in PID tuning, converging at iteration **7** with optimal parameters. The best cost function value achieved is **5.083**. In Table 5, The performance of the JS optimizer is compared to those of GA and conventional PID.

Table 5: Comparative study of PID gains.

Technique	K_p	K_i	K_d	**Objective Function**
Convetional PID	19	100	0.5	44.7791
JS optimizer	4.3455	499.998	9.7863	5.083
GA	0.00087	499.999	4.2312	5.082

Source: Authors, (2025).

In terms of objective function optimization, metaheuristic algorithms (JS & GA) effectively improve PID tuning compared to conventional methods. These algorithms, produce very similar performance. However, JS algorithm has a slightly more balanced gain distribution. Furthermore, The GA algorithm converged to the best solution around iteration **39** as it's highlighted in Table 4, while JS optimizer converged at the **7th** iteration, reducing time cost while maintaining optimization quality.

The speed outputs and their corresponding Error plot, of the Jellyfish algorithm (JS), Genetic algorithm (GA), and conventional PID (Conv PID) are depicted in Figures 8 and 9 respectively. All methods perform good tracking performance. However, the zoomed-in inset highlights an overshoot exhibited by conventional PID response while, GA and JS optimizers perform smoother responses with less deviation, indicating better control performance.

Figure 9 shows that the Conv PID output falls short in the transitory phase, with larger transient errors, and also struggles with steady-state accuracy. In contrast, GA and JS are proven effective. Table 6 highlights the results discussed earlier, where both JS Optimizer and GA outperform conventional PID tuning, with zero overshoot, faster settling times, and lower steady-state errors. JS Optimizer slightly outperforms GA in rapidity performance.

Table 6: Performance criteria comparison.

Technique	**Overshoot (%)**	**Settling Time (s)**	**Rise time (s)**	**Steady-State Error (m/s)**
Convetional PID	0.52	41.64	34.236	10^{-4}
JS optimizer	0	38.93	34.153	10^{-5}
GA	0	40.41	34.156	10^{-5}

Source: Authors, (2025).

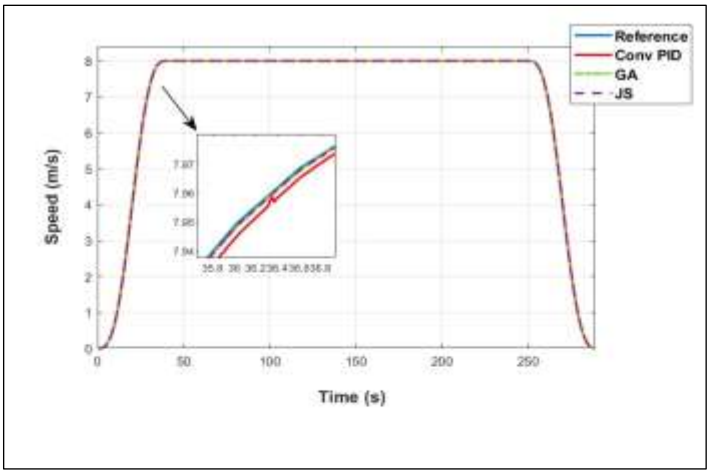

Figure 8: Speed response of the longitudinal PID control system.
Source: Authors, (2025).

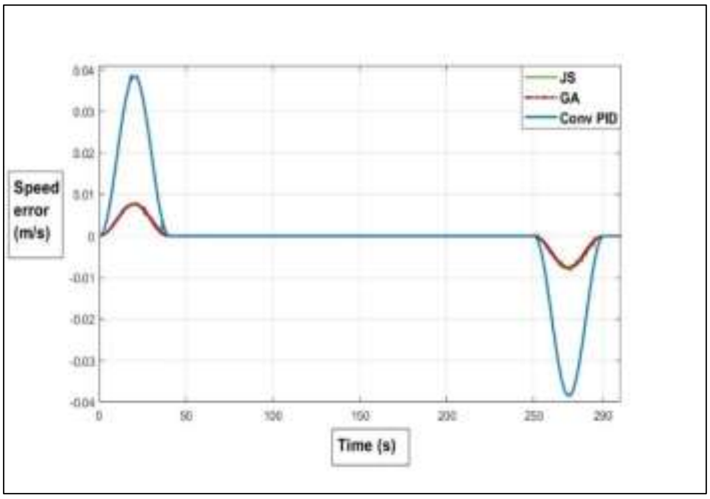

Figure 9: Error plot.
Source: Authors, (2025).

V. CONCLUSIONS

This paper presented a contribution to the implementation of Metaheuristic optimization algorithms in the field of Electric Autonomous Vehicles, by comparing Jellyfish Search (JS) and Genetic Algorithm (GA) to conventional methods in tuning PID controllers for the construction of a robust yet simple longitudinal control system.

Furthermore, simulation results were conducted on a realistic driving scenario using a sinusoidal speed profile, adhering to road standards, leading to smoother, safer, and more efficient driving performance.

The effectiveness of metaheuristic optimization techniques (Jellyfish Search and Genetic Algorithm) has been proven in tuning PID controllers for their significant improvements in key performance metrics such as overshoot, settling time, rise time, and steady-state error.

Both GA and JS eliminated the delay and overshoot exhibited by the conventional PID. Notably, JS provided the fastest convergence and required less computation time due to its ability to balance between exploration and exploitation, making it an excellent metaheuristic algorithm for EAV’s speed control.

However, further work should be considered to explore hybrid optimization techniques and real-time implementation to further validate these findings.

VI. AUTHOR’S CONTRIBUTION

Conceptualization: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Methodology: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Investigation: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Discussion of results: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Writing – Original Draft: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Writing – Review and Editing: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Resources: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Supervision: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.
Approval of the final text: Asmaa Guendouz, Mustapha Hatti, Abdelhalim Tlemçani.

VIII. REFERENCES

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," in *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, March 2016, doi: 10.1109/TIV.2016.2578706.

[2] H. R. Moshtaghi, A. T. Eshlaghy, and M. R. Motadel, “A comprehensive review on meta-heuristic algorithms and their classification with novel approach,” *J. Appl. Res. Ind. Eng.*, vol. 8, no. 2, pp. 151–174, 2021, doi: 10.22105/jarie.2021.238926.1180.

[3] H. S. Elias, “A longitudinal control for an autonomous vehicle using modified particle swarm optimization method,” *Int. J. Eng. Technol.*, vol. 13, no. 1, pp. 123–131, 2024, doi: 10.14419/q6bgfv04.

[4] R. Pazmiño, W. Pavon, M. Armstrong, and S. Simani, “Performance evaluation of fractional proportional–integral–derivative controllers tuned by heuristic algorithms for nonlinear interconnected tanks,” *Algorithms*, vol. 17, no. 7, Art. no. 306, Jul. 2024, doi: 10.3390/a17070306.

[5] R. T. Marler and J. S. Arora, “The weighted sum method for multi-objective optimization: new insights,” *Struct. Multidisc. Optim*, vol. 41, pp. 853–862, 2010, doi: 10.1007/s00158-009-0460-7.

[6] T. Alam, S. Qamar, A. Dixit, and M. Benaida, “Genetic algorithm: Reviews, implementations, and applications,” *Int. J. Eng. Pedagog*., vol. 10, no. 6, pp. 57–66, 2020, doi: 10.3991/ijep.v10i6.14567.

[7] A. Khare, G. M. Kakandikar, and O. K. Kulkarni, “An insight review on jellyfish optimization algorithm and its application in engineering,” *Rev. Comput. Eng. Stud.*, vol. 9, no. 1, pp. 31–40, 2022, doi: 10.18280/rces.090103.

[8] F. Mohamad Farid and P.-W. Hsueh, “Parameters identification of a permanent magnet DC motor: A review,” *Electronics*, vol. 12, no. 12, Art. no. 2559, 2023, doi.3201410.3390/electronics12122559.

[9] LEROY-SOMER, “D.C. motors 2 to 750 kW,” 2020. [Online]. Available: https://www.leroysomer.com/documentation_pdf/3805_en.pdf.

[10] S. Singh, V. Singh, A. Rani and J. Yadav, "Optimization of PID controller based on various tuning methods," *2023 International Conference on Power, Instrumentation, Energy and Control (PIECON)*, Aligarh, India, 2023, pp. 1-6, doi: 10.1109/PIECON56912.2023.10085805.

[11] J. E. Naranjo, F. Serradilla, and F. Nashashibi, “Speed control optimization for autonomous vehicles with metaheuristics,” *Electronics*, vol. 9, no. 4, Art. no. 551, Mar. 2020, doi: 10.3390/electronics9040551.

[12] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm- A Literature Review," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 380-384, doi: 10.1109/COMITCon.2019.8862255.

[13] Y. Kebbati, N. Ait-Oufroukh, V. Vigneron, D. Ichalal and D. Gruyer, "Optimized self-adaptive PID speed control for autonomous vehicles," *2021 26th International Conference on Automation and Computing (ICAC)*, Portsmouth, United Kingdom, 2021, pp. 1-6, doi: 10.23919/ICAC50006.2021.9594131.

[14] J.-S. Chou and D.-N. Truong, “A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean,” *Appl. Math. Comput.*, vol. 389, Art. no. 125535, 2020, doi: 10.1016/j.amc.2020.125535.

[15] A. F. Güven, et al., “Comprehensive optimization of PID controller parameters for DC motor speed management using a modified jellyfish search algorithm,” *Optim. Control Appl. Methods*, 2024, doi: 10.1002/oca.3218.