Enhancing Cruise Performance through PID Controller Tuned with Particle Swarm Optimization Technique

Nema Salem*
Electrical and Computer Engineering Department
Effat University
Jeddah, Saudi Arabia
nsalem@effatuniversity.edu.sa
orcid.org/0000-0002-8440-4111

Rana Hassan
Electrical and Computer Engineering Department
Effat University
Jeddah, Saudi Arabia
ranhassan@effat.edu.sa

Lina Muthanna
Electrical and Computer Engineering Department
Effat University
Jeddah, Saudi Arabia
limuthanna@effat.edu.sa

***Abstract*—The Proportional-Integral-Derivative (PID) controller is a widely used feedback control mechanism in various applications, including automobile cruise control systems. The performance of a PID controller is highly dependent on the values of its tuning parameters, which can be challenging to determine in practice. Particle Swarm Optimization (PSO) has emerged as a powerful optimization algorithm that can tune the PID controller parameters for optimal performance. The PSO is a metaheuristic optimization algorithm inspired by the social behavior of birds and fish. The PSO-PID is a variant of the PID controller that employs PSO to optimize its tuning parameters. PSO-PID offers several advantages over traditional PID tuning methods, including improved accuracy, stability, and robustness. This paper briefly overviews the PSO-PID algorithm and its application to automobile cruise control systems. The paper discusses the key steps involved in PSO-PID tuning, including initialization, evaluation, update, and termination. It provides an example of how PSO-PID can achieve optimal vehicle speed control. The paper highlights the advantages of PSO-PID over traditional PID tuning methods. PSO-PID is a promising technology for cruise control systems and has the potential to significantly improve the safety, comfort, and efficiency of modern automobiles.**

***Index Terms*—PID, PID-PSO, tuning, cruise control, particle swarm optimization, PSO, TIAE cost function**

I. Introduction

Cruise control is a technology that is used in automobiles to maintain a constant vehicle speed without the need for driver intervention. The system is designed to provide a more comfortable and efficient driving experience by automating and controlling the vehicle's speed. Cruise control systems typically utilize a closed-loop feedback control mechanism that compares the vehicle speed to the desired set speed and adjusts the control input accordingly. By maintaining a constant speed, cruise control can help reduce the need for frequent speed adjustments, resulting in a more relaxed and comfortable driving experience for the driver. Additionally, by minimizing unnecessary acceleration and deceleration, cruise control can help to reduce fuel consumption and emissions, resulting in cost savings and environmental benefits [1].

As in [2], the Conventional Cruise Control (CCC) system maintains a constant vehicle speed without requiring the driver to adjust the throttle pedal constantly. It is a relatively simple and widely used system that has been available in vehicles for several decades. Its primary components are a speed sensor, a control module, and an actuator. The speed sensor measures the vehicle's speed and sends this information to the control module. The control module compares the current speed to the setpoint speed, which the driver selects using a button. It adjusts the throttle position accordingly using the actuator. The vehicle speed is maintained at the selected speed until the driver cancels the system or applies the brakes. However, it cannot adapt to changes in traffic conditions and may cause the vehicle to slow down or speed up abruptly. In addition, it can be affected by changes in road gradients and wind resistance, which can cause the vehicle to lose or gain speed.

As in [1], the adaptive cruise control (ACC) is an advanced version of the (CCC). It adjusts the vehicle speed in response to changes in traffic conditions. Its primary components are a radar or camera sensor, a control module, and an actuator. The sensor continuously measures the distance and relative speed of the vehicle ahead and sends this information to the control module. The control module uses this information to adjust the vehicle speed and maintain a safe following distance. The actuator adjusts the throttle position and/or applies the brakes to control the vehicle speed. Depending on the desired distance and driving conditions, it can operate in several modes, such as standard, medium, or long-range. it incorporates additional

features such as collision avoidance and pedestrian detection. It has several advantages over CCC, such as improved safety, increased comfort, and reduced driver fatigue. However, its limitations include reduced effectiveness in adverse weather conditions, limited reliability in detecting certain obstacles, and the potential for false alarms or sudden braking.

A proportional integral derivative (PID) controller is a closed-loop feedback mechanism extensively used in commercial control systems. It calculates an error value as the difference between the measured process variable and the desired set point. The controller tries to minimize errors by using manipulated variables to tune the process. PID can stabilize a system by altering its response to variable factors, allowing it to remain dependable and stable. Those parameters are tuned to ensure that the system will remain stable; it can also be used to control feedback systems since it can be implemented as a PD, PI, or PID; researchers have a wide range of choices when working with this type of controller. There are multiple ways in which a PID can be configured, such as parallel, series PID, and the ideal configuration of PID [3].

The three control elements of a parallel PID controller are combined in parallel to produce a single output signal that adjusts a system [4]. The proportional control element is based on the current error between the system output and the desired setpoint. The integral control element sums up past errors over time to reduce steady-state error, and the derivative control element considers the error rate of change to improve the controller's response time. This controller is expressed in (1).

$$G_{PID} = \frac{C(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d \times s \tag{1}$$

where $c(t)$ is the controller output at time t, $e(t)$ is the error signal at time t, K_p, K_i, and K_d are the proportional, integral, and derivative coefficients, respectively. The parallel PID controller is widely used in industrial applications such as temperature control, speed control, and position control. It is simple to implement and can be tuned easily to achieve optimal performance for a given system.

In contrast to a parallel PID controller, the three elements of the series PID are combined in series. The proportional control element is applied first to the error signal. The output of the proportional control element is then fed into the integral control element, which sums up the past errors over time, and the output of the integral control element is further fed into the derivative control element, which takes into account the rate of change of the error. This controller is expressed in (2). The series PID controller has better disturbance rejection and improved robustness than the parallel type. However, it is generally more difficult to tune than the parallel form.

$$G_{PID} = \frac{C(s)}{E(s)} = \left[K_p\left(1 + \frac{K_i}{s}\right)\right](K_d \times s + 1) \tag{2}$$

An ideal PID controller provides perfect system control with zero steady-state error and a fast response time. In practice, realizing an ideal PID controller is impossible due to various limitations, such as sensor noise, actuator saturation, and modeling errors. This controller is expressed in (3).

$$\begin{aligned} G_{PID} &= \frac{C(s)}{E(s)} = K_p\left[1 + \frac{K_i}{s} + K_d \times s\right] \\ c(t) &= K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(t)\,\mathrm{d}t + T_d\frac{de(t)}{dt}\right] \\ &= K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)\,\mathrm{d}t + K_p T_d\frac{de(t)}{dt} \\ &= K_p e(t) + K_i\int_0^t e(t)\,\mathrm{d}t + K_d\frac{de(t)}{dt} \end{aligned} \tag{3}$$

PID controllers can be used in conjunction with ACC systems to improve the responsiveness and stability of the system. The PID controller is used to adjust the throttle position or apply the brakes to maintain the desired speed set by the ACC system. The basic idea behind using a PID controller with ACC is to use the error between the current vehicle speed and the desired speed as the input to the PID controller. The output of the PID controller is then used to adjust the vehicle speed or braking force to maintain the desired setpoint. The proportional term of the PID controller is used to adjust the throttle position or braking force in proportion to the error between the current and desired speed. The integral term is used to eliminate any steady-state error in the system. The derivative term is used to improve the system's response to changes in the error signal. Adaptive PID controllers can adjust the PID gains based on the system's performance characteristics or operating conditions. These controllers can improve the system's stability and responsiveness in varying driving conditions and help to reduce overshoot and oscillations [1].

PID tuning is the process of adjusting the parameters of a PID controller to achieve a desired control response. The three parameters of a PID controller are used to determine the output of the controller based on the error between the desired setpoint and the measured process variable, as well as the rate of change of the error. PID tuning involves selecting appropriate values for K_p, K_i, and K_d, resulting in stable and responsive system control [5]. This is typically done through a trial-and-error process, where the controller parameters are adjusted, and the system response is observed until the desired performance is achieved. There are several classical methods for PID tuning, including the Ziegler-Nichols method, the Cohen-Coon method, and the Lambda tuning method. Each method has advantages and disadvantages, and the choice of tuning method will depend on the specific application and system requirements [6].

Intelligent PID tuning methods are intelligent techniques such as neural networks [7], fuzzy logic [8], and genetic algorithms (GA) [9], used to automate the tuning process and improve the performance of the controller in complex and nonlinear systems.

This study investigates the effectiveness of using a combination of PID control and Particle Swarm Optimization (PSO)

algorithm for designing a cruise control system. The paper is organized as follows: an overview of classic PID tuning methods is presented in Section II. At the same time, Section III discusses the intelligent PID tuning method: Particle Swarm Optimization, PSO. Section IV shows a selected cruise model. Section V presents the design and analysis of the proposed system, including the design of the PID-PSO cruise control system. The system simulation results are addressed in Section VI. Finally, conclusions are given in Section VII.

II. The Classic PID Tuning Methods

The Ziegler-Nichols (ZN) method is widely used for tuning PID controllers. It involves performing step tests on the system to determine its response characteristics and then using these characteristics to determine appropriate values for the PID controller parameters. The (ZN) procedures are as follows [10].

1) Find the system's step response for $K_d = K_i = 0$ and a small value of K_p.
2) Increase the value of K_p gradually until getting the oscillation response.
3) Record the ultimate gain Ku and the oscillation period Tu.
4) Use the Table I for the calculation of K_p, K_i and K_d.

TABLE I: Ziegler-Nichols tuning method for the classic PID controller.

K_p	T_i	T_d	K_i	K_d
0.6 Ku	$Tu/2$	$Tu/8$	1.2 Ku/Tu	0.075 $Ku \times Tu$

III. The Intelligent PID Tuning Method: Particle Swarm Optimization, PSO

The PSO [11] is a popular metaheuristic optimization algorithm that can be used for PID tuning [12]. It is inspired by the social behavior of birds and fish and involves iteratively adjusting a population of particles to find the optimal solution to a problem. Its mathematical approach involves the following five steps.

1) Define the population of particles, each with a position and a velocity, and randomly initialize their positions and velocities.
2) Evaluate the fitness of each particle based on the objective function.
3) Update the best positions found by each particle and its neighbors based on their fitness.
4) Update the velocity and position of each particle based on its current position, velocity, and the best positions found by itself and its neighbors.
5) Repeat steps (2-4) until the desired level of performance is achieved or a termination criterion is met.

The velocity and position updates are given in (4) and (5).

$$v_i(t+1) = wv_i(t) + c_1 r_1 (pbest_i - x_i(t)) + c_2 r_2 (gbest - x_i(t)) \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

Where $v_i(t)$ is the velocity of particle i at time t, $v_i(t+1)$ is the new velocity computed using the velocity update equation, $x_i(t)$ is its position at time t. w is the inertia weight that represents the impact of the previous velocity on the updated velocity. It is typically set to a value between 0 and 1. c_1 and c_2 are the acceleration coefficients, which control the influence of each particle's personal best position $pbest$ and the global best position $gbest$, respectively. They are typically set to a value between 0 and 2. r_1 and r_2 are random numbers between 0 and 1, introducing stochasticity into the algorithm.

From (4), the new velocity of the particle is a combination of its previous velocity (weighted by the inertia weight), its personal best position (weighted by the acceleration coefficient c_1 and a random term r_1), and the global best position (weighted by the acceleration coefficient c_2 and a random term r_2). The personal best position is the best position the particle has found, while the global best position is the best one found by any particle in the swarm. (5), the position update equation updates the position of the particle based on its new velocity. The velocity and position update equations work together to guide the particles toward the optimal or near-optimal solution to the problem. By adjusting the values of the inertia weight and acceleration coefficients, the PSO algorithm can balance exploration and exploitation and converge to a good solution within a reasonable number of iterations.

IV. Cruise Modeling

Cruise control systems require accurate modeling of the vehicle dynamics and control inputs to achieve stable and effective control [13]. The cruise control system manages the vehicle's speed to respond to the driver's commands and keep the speed at the set level. In an inclined road, two main factors prevent the car from keeping a constant speed: the slope of the road that creates a gravitational downward force at an angle θ, and air resistance that opposes the car's movement. In addition to other parameters such as the vehicle's mass and the road's friction. Figure 1 shows the factors that affect the velocity.

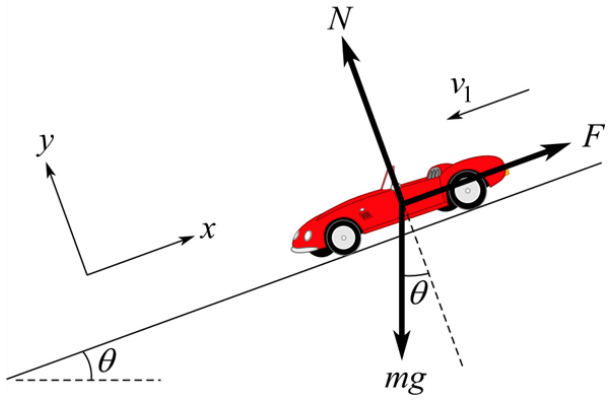

Fig. 1: Factors affecting the cruise's velocity.

According to [14], the transfer function of a vehicle on an inclined road is as in (6), and the final transfer function of the vehicle model is given by (7).

$$G_p(s) = \frac{\triangle V(s)}{\triangle U(s)} = \frac{C_1 e^{-\tau s}/MT}{(s + 2C_a v/M)(s + 1/T)(s + 1/\tau)} \quad (6)$$

$$= \frac{2.4767}{(s + 0.0476)(s + 1)(s + 5)}$$

$$= \frac{2.4767}{s^3 + 6.0476s^2 + 5.2856s + 0.238} \quad (7)$$

C_a is the aerodynamic drag coefficient, M is the vehicle's mass, including passengers, v is the velocity, C_1 is the force saturation constant, and T is the time constant of the first order lag. A time delay is represented by an exponential function approximated to the first-order transfer function.

V. The Proposed Control System

The proposed design of the PID tuning relies on the PSO algorithm. The process focuses on minimizing the error by finding the optimal gain value of each PID parameter. This is achieved by calculating the Time Integral Absolute Error, TIAE, to control the cruise's velocity and decrease the control system's rise time and settling time. The cost function that depends on the TIAE is the difference between the absolute value of the output of the PID-controlled system and the desired output, the reference point, which is the unit step in this case. The block diagram in Fig. 2 represents the flow of the function from the input to the output velocity.

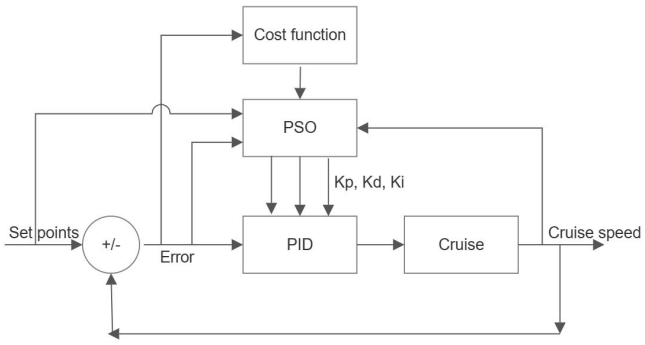

Fig. 2: PID-PSO cruise control system block diagram.

Using MATLAB, the $PSO - PID$ controller is designed through the following steps.

1) Specify the cruise's plant transfer function and find its uncontrolled unity feedback step-response.
2) Let the initial values of the PSO constant parameters: $c_1 = c_2 = 2, w = 0.4$, the number of particles = 30, the number of iterations = 50, and the lower and upper bounds are $a = 0, b = 30$.
3) Start with a zero initial velocity, find the best particle position for each particle and each variable to update the PID parameters K_p, K_i, and K_d. Then, find the step-response of this PSO-controlled system.
4) The Time Integral Absolute Error, TIAE, is the objective function that optimizes the PID parameters. It acts as a counter that finds the absolute value of the PID step response and the desired output for each particle. Track its minimum value at each iteration.
5) According to the minimum and location of TIAE particles, find the three best PID parameters that allow the best performance.
6) Iteration: The PSO updates each particle's velocity and position according to (4) and (5). After the positions have been updated, the particles' positions must be verified within the search space region. A condition statement is passed through that updates the boundaries of the search space. The lower boundary changes if the particle's new position is less than the lower boundary, and similarly, the upper bound is altered. The new position is updated to its value if it exceeds the upper boundary. Then, the model parameters are updated within the new positions of the particles, and again the TIAE is computed accordingly. A comparison is made between the initial and final position values, and the minimum value of ITAE is declared as a new global variable along with its position. The objective function is computed, and the PID parameters are now tuned according to the lowest value of the calculated error. The transfer function of the vehicle is passed through the new controlled system to find its step response and observe the PSO-based PID controller's effect on the system's step response.

VI. Results

Firstly, we utilized MATLAB/Simulink to track the step response of the cruise for the open-loop uncontrolled system. The time performance is shown in Fig. 3 and depicted in the first column of Table II. Secondly, we studied the uncontrolled step response for a negative unity feedback system. The time response and its metrics are shown in Fig. 4, (in blue) and the second column of Table II. Thirdly, for a PID cruise system with the ZN tuning method, we utilized the Simulink shown in Fig. 5. We followed the ZN procedures to find the ultimate gain and ultimate period at which the cruise's step response reaches the oscillation shown in Fig. 6. The simulation measures results gave $Ku = 12.8$ and $Tu = 2.723$. The calculated three PID parameters are $K_p = 7.68, K_d = 2.5248$, and $K_i = 5.8403$. The PID-ZN system's response is shown in Fig. 4, (red curve), and the third column of Table II.

From the obtained results, the open-loop system is unacceptable. The step response of the uncontrolled closed-loop system took much time to reach the maximum speed and also to reach its steady state. This time must be reduced since, in a velocity control system, time plays a vital role as it is the main factor that affects the error signal sent to the car's response system. In addition, the steady state does not comply with the desired output, with a difference of 8.8% of the final output.

We investigated the PSO-PID to decrease the rise and settling time and reach the desired output. The transfer function of the PSO-PID system is given in (8) with optimum PID three parameters: $K_p = 22.7208, K_d = 14.2872$ and $K_i = 16.621$.

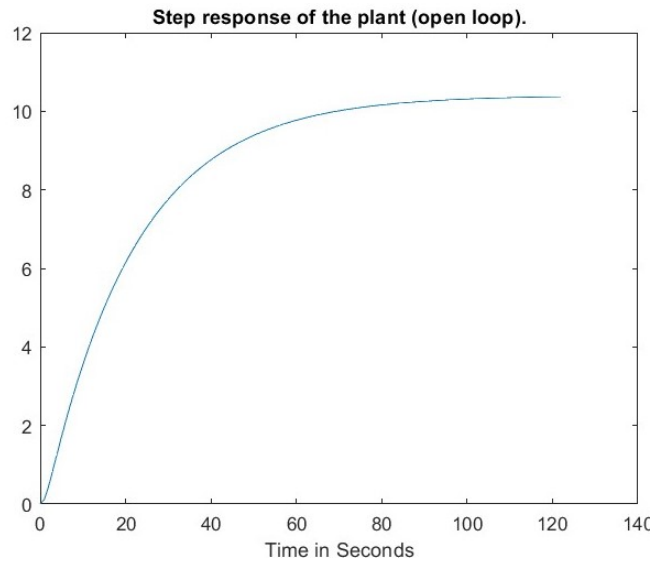

Fig. 3: Step response of the uncontrolled open-loop system.

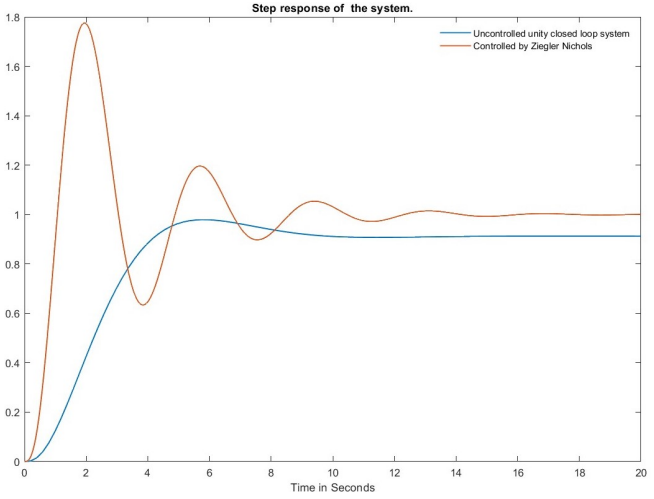

Fig. 4: Step response of the uncontrolled closed-loop and ZN- PID system.

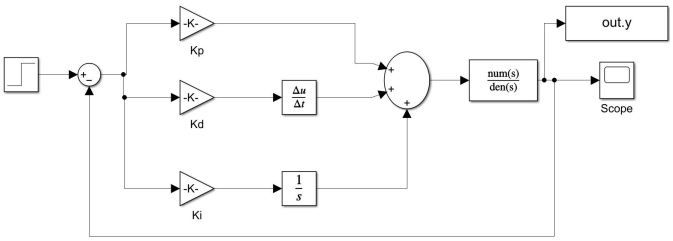

Fig. 5: Simulink for Step response of the PID cruise system tuned by ZN.

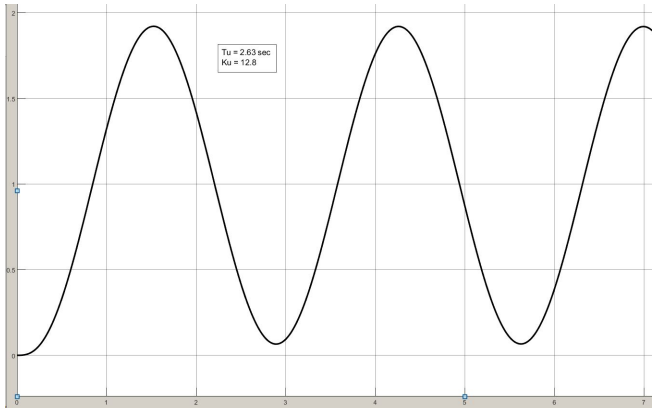

Fig. 6: The sinusoidal ZN cruise system response.

The obtained results are shown in Fig. 7 and the fourth column of Table II.

$$TF = \frac{35.39s^2 + 56.27s + 43.64}{s^4 + 6.048s^3 + 40.67s^2 + 56.51s + 43.64} \quad (8)$$

This system has a rise and settling time of 0.252 and 2.32 seconds, respectively, leading to a faster response than others which is a crucial point in velocity control. In addition, this system has an exceptional improvement in the steady state value reaching precisely the desired output. However, an unwanted overshoot is generated in such a control system as it would result in a sudden increase in the vehicle's velocity that might cause crashes or serious accidents. This issue can be resolved with further studies of the proposed algorithm to cut the overshoot and create a stable, controlled system.

Since the algorithm's main objective is to find the optimal PID parameters to minimize the TIAE of the system response along several iterations, Fig. 8 shows the change in the TIAE with iterations. It shows that the error is fixed at 137 throughout the iterations $(0 - 18)$, then it decreased to 125 and stayed there for the $(18 - 42)$ iterations, and at the end, it rested at 124 for the $(42 - 50)$ iterations.

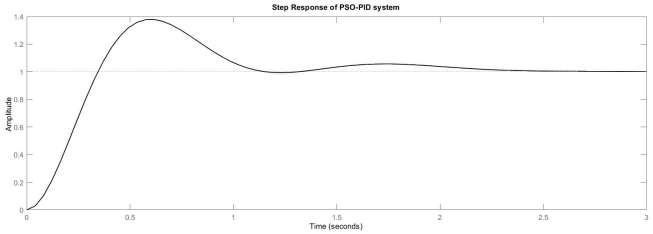

Fig. 7: Step response of the PSO-PID system.

VII. Conclusion

This research proposed the PID system for controlling the speed of a cruise system using the PSO algorithm. The results showed a faster and more accurate system. However, it generates an overshoot that can be eliminated with further enhancements to the proposed algorithm. Compared to an

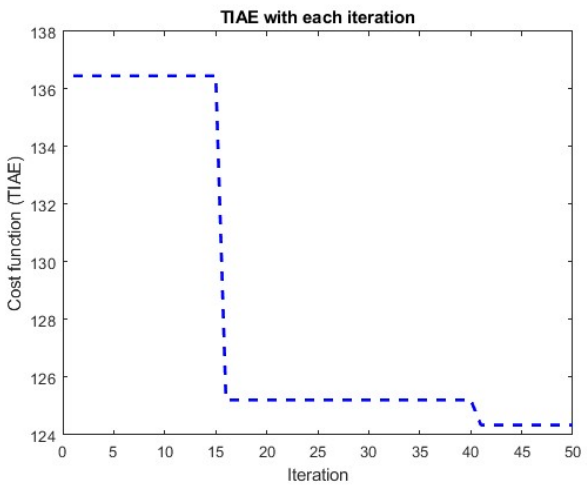

Fig. 8: The cost function: TIAE with each iteration.

TABLE II: Performance specifications for uncontrolled and controlled PSO-PID systems.

	Open loop	Closed loop	ZN-PID	PSO-PID
Rise time $(sec.)$	47	2.71	0.63	0.252
Settling time $(sec.)$	84	8.44	11.738	2.32
% Overshoot	0	7.3	77	38
Peak value	10.37	0.979	1.775	1.38

uncontrolled system, PSO-PID cruise control can maintain a more consistent and accurate vehicle speed, reducing the need for driver intervention and improving driving comfort. Additionally, PSO-PID can reduce fuel consumption by minimizing unnecessary acceleration and deceleration, resulting in significant cost savings over the vehicle's lifetime. Overall, using PSO-PID cruise control can provide a safer, more comfortable, and more efficient driving experience and is, therefore, a valuable technology for modern automobiles. However, it is essential to note that the performance of PSO-PID may be affected by the specific vehicle dynamics and operating conditions and may require periodic tuning and adjustment to maintain optimal performance.

REFERENCES

[1] Yu Zhang et al. "Control design, stability analysis, and traffic flow implications for cooperative adaptive cruise control systems with compensation of communication delay". In: *Transportation Research Record* 2674.8 (2020), pp. 638–652.

[2] Yinglong He et al. "Adaptive cruise control strategies implemented on experimental vehicles: A review". In: *IFAC-PapersOnLine* 52.5 (2019), pp. 21–27.

[3] James Crowe et al. *PID control: new identification and design methods*. Springer, 2005.

[4] Norman S Nise. *Control systems engineering*. John Wiley & Sons, 2020.

[5] Qing-Guo Wang and Zhuo-Yun Nie. "PID control for MIMO processes". In: *PID Control in the Third Millennium: Lessons Learned and New Approaches* (2012), pp. 177–204.

[6] Paraskevas N Paraskevopoulos. *Modern control engineering*. CRC Press, 2017.

[7] Samira Johari, Mahdi Yaghoobi, and Hamid R Kobravi. "Nonlinear model predictive control based on hyper chaotic diagonal recurrent neural network". In: *Journal of Central South University* 29.1 (2022), pp. 197–208.

[8] AH Gomaa Haroun and Li Yin-Ya. "A novel optimized fractional-order hybrid fuzzy intelligent PID controller for interconnected realistic power systems". In: *Transactions of the Institute of Measurement and Control* 41.11 (2019), pp. 3065–3080.

[9] Fulu Cao. "PID controller optimized by genetic algorithm for direct-drive servo system". In: *Neural Computing and Applications* 32.1 (2020), pp. 23–30.

[10] John G Ziegler and Nathaniel B Nichols. "Optimum settings for automatic controllers". In: *Transactions of the American society of mechanical engineers* 64.8 (1942), pp. 759–765.

[11] Ke-Lin Du et al. "Particle swarm optimization". In: *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature* (2016), pp. 153–173.

[12] Zhenglong Xiang et al. "A simple PID-based strategy for particle swarm optimization algorithm". In: *Information Sciences* 502 (2019), pp. 558–574.

[13] Yuncheng Jiang. "Modeling and simulation of adaptive cruise control system". In: *arXiv preprint arXiv:2008.02103* (2020).

[14] Ahmed Abdulnabi. "PID controller design for cruise control system using particle swarm optimization". In: *Iraqi Journal for Computers and Informatics* 43.2 (2017), pp. 30–35.