Modern Applied Science; Vol. 18, No. 1; 2024
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

# Optimized PID Controller for Quarter-Car Active Suspension System Under Various Road Profiles

Mohd Fairus Abdollah[1], Nor Hani Md Desa[1], Hairol Nizam Mohd Shah[1], Mohd Zamzuri Ab Rashid[1], Azhar Ahmad[1] & Mohd Ali Arshad[2]

[1] Center for Robotics and Industrial Automation, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia

[2] Neuro Dynamics Enterprise, Kompleks Niaga Melaka Perdana, Bukit Katil, Melaka, Malaysia

Correspondence: Mohd Fairus Abdollah, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia. E-mail: mfairus@utem.edu.my

Received: August 28, 2024 Accepted: November 15, 2024 Online Published: November 21, 2024

doi:10.5539/mas.v18n1p30 URL: https://doi.org/10.5539/mas.v18n1p30

**Abstract**

The main limitation of passive suspension system lies in their inherent compromise between ride comfort and car handling, resulting from their inability to dynamically adjust to varying road conditions. Efforts to enhance riding comfort often led to trade-offs that may compromise safety, and vice versa. This duality necessitates a more adaptable and flexible solution. Active suspension systems emerge as a transformative methodology, allowing real-time adjustments and dynamic modifications to damping characteristics. This capability effectively separates the compromise between ride enjoyment and safety, enabling an optimal equilibrium by adaptively responding to fluctuations in road conditions. This paper presents a quarter-car active suspension system to improve comfort under various road conditions. A PSO optimized PID controller is implemented to minimize both the sprung mass displacement, and the sprung mass acceleration subjected to single bump and dual bump road profile. The performance of the PSO-based PID controller is illustrated by simulation results in MATLAB, demonstrating significant improvements in body displacement and body acceleration, thereby enhancing the ride comfort by adaptively responding to road conditions in real time.

**Keywords:** active suspension system, particle swarm optimization, proportional-integral-derivative controller, quarter-car, road disturbances

## 1. Introduction

A passive suspension system is a conventional system that generally consists of non-controlled springs and shock absorbing dampers that can store energy via the springs and dissipate it via the dampers. These components can minimize the vertical accelerations transmitted to the passenger and at the same time ensure the tire remains in contact with the road surface. However, these two characteristics conflict with each other where a hard suspension will reduce the passenger comfort, but a soft suspension will reduce the stability of the car during cornering. Thus, a controlled suspension system is necessary where the suspension will react accordingly, depending on the road conditions.

Active suspension systems strongly react towards changes in the road surface due to their ability to provide energy that can be utilized to develop relative motion. Sensors to monitor suspension fluctuations, such as body velocity, suspension displacement, wheel velocity, and wheel or body acceleration, are frequently included in active suspension systems. In the active suspension system as shown in Figure 1, actuators are added to the passive components to provide extra forces dictated by a feedback control law. The controller will calculate either to add or dissipate energy from the system with the help of sensors as the input.

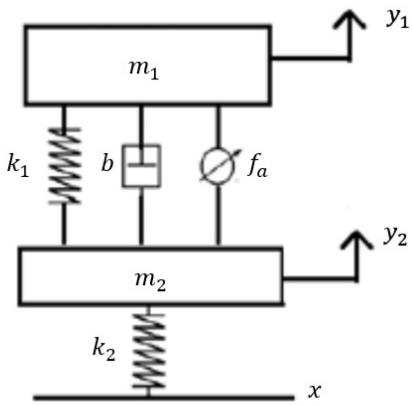


Figure 1. Quarter-car active suspension system

Table 1. Parameters of Quarter-Car System

| Parameters | Symbols | Values |
|---|---|---|
| Sprung mass | $m_1$ | 250 kg |
| Un-sprung mass | $m_2$ | 50 kg |
| Spring constant of suspension system | $k_1$ | 16 000 N/m |
| Spring constant of wheel and tire | $k_2$ | 160 000 N/m |
| Damping constant of suspension system | $b$ | 1500 N.s/m |
| Sprung mass displacement relative to the ground | $y_1$ | - |
| Un-sprung mass displacement relative to the ground | $y_2$ | - |
| Road surface uneven relative to the ground | $x$ | - |

There are different control strategies, such as optimal state feedback by Nekoui and Hadavi (2010), Mohamed, Abidou & Maged (2021), Semion and Presnova (2022) and Bharali and Buragohain (2016) and fuzzy based control by Azizi, Mirzaei, Rafatnia & Mehrabani (2024), Rao and Anusha (2013), Sadeghi, Varzandian & Barzegar (2011), Talib and Darus (2014), Lee, Kang, Pae, Choi & Lim (2022) and Hurel, Amaya, Peralta, Alvarado & Flores (2022) for active suspension system. In this paper, the research focuses on the proportional-integral-derivative (PID) controller as one alternative for designing a reliable active suspension.

PID controller is one method to obtain the feedback of specific systems. The feedback control design that is most often used is the PID controller. The desired closed-loop dynamics are obtained by adjusting the three parameters $K_p$, $K_i$, and $K_d$, often iteratively by "tuning" and without specific knowledge of a plant model. The proportional term is frequently sufficient to provide stability. A step disturbance, which is frequently a startling specification in process control, can be rejected thanks to the integral term. A response's shaping or dampening is accomplished via the derivative term. PID controllers represent the most popular category of control systems.

The contribution of this paper is on the optimization of the PID by using PSO algorithm for active suspension system under the various road conditions. The active suspension system performs better when the Particle Swarm Optimization (PSO) algorithm is used since it makes it easier to alter the PID parameters iteratively. The suspension system is guaranteed to adjust smoothly to changing road conditions and disturbances thanks to this dynamic optimization process. The active suspension system, which demonstrates the adaptability and efficiency of PID controllers in the field of control systems engineering, strikes an ideal balance between stability, disturbance rejection, and response shaping thanks to the cooperative synergy of PID control and the PSO algorithm.

This paper is organized as follows. The system dynamics of a quarter-car active suspension system along with the road profiles are discussed in Section 2. A PSO optimized PID controller is also designed in Section 2. In Section 3, the control results of traditional passive suspension system, Ziegler-Nicols based PID and the PSO optimized PID are compared under various road profiles. Finally, some conclusions are drawn in Section 4.

## 2. Method

*2.1 Quarter-Car Active Suspension System and Road Profiles*

The quarter car model can be said to be the simplest model among all the models. Most quarter car models are analyzed on two degrees of freedom (DOF) which are the vertical displacement of the sprung mass and un-sprung mass. Although it has less variables thus making it less complicated to analyze, it is enough to develop the active suspension based on this model as it will still show the differences between passive and active suspension. The quarter-car model shown in Figure 1 is used to develop the dynamic model of the active suspension system.

2.1.1 Dynamic Model of Quarter-Car Active Suspension System

The following equations are used to generate the quarter car model using Newton's second law.

$$m_1\ddot{y}_1 = -k_1(y_1 - y_2) - b(\dot{y}_1 - \dot{y}_2) \tag{1}$$

$$m_2\ddot{y}_2 = k_1(y_1 - y_2) + b(\dot{y}_1 - \dot{y}_2) + k_2(y_2 - x) \tag{2}$$

By using Laplace transform for (1) and (2) then rearrange to form the following equations

For $Y_1(s)$:

$$m_1\, s^2\, Y_1(s) + k_1Y_1(s) + bsY_1(s) = k_1Y_2(s) + bsY_2(s) \tag{3}$$

$$(m_1s^2 + k_1 + bs)Y_1(s) = (k_1 + bs)Y_2(s) \tag{4}$$

For $Y_2(s)$:

$$m_2s^2Y_2(s) = k_1Y_1(s) + bsY_1(s) - (k_1 + k_2)Y_2(s) - bsY_2(s) + k_2X(s) \tag{5}$$

$$(m_2s^2 + k_1 + k_2 + bs)Y_2(s) = k_1Y_1(s) + bsY_1(s) + k_2X(s) \tag{6}$$

Based on (3)-(6), solving for $Y_1(s)$ and $Y_2(s)$ will yield

$$Y_1(s) = (k_1 + bs)k_2X(s)/[(m_1s^2 + k_1 + k_2 + bs)(m_2s^2 + k_1 + bs) - (k_1 + bs)^2] \tag{7}$$

$$Y_2(s) = k_2X(s)(m_1s^2 + k_1 + bs)/[(m_2s^2 + k_1 + k_2 + bs)(m_1s^2 + k_1 + bs) - (k_1 + bs)^2] \tag{8}$$

Further simplify (7) and (8), the transfer function for body displacement and body acceleration is give in (9) and (10), respectively

$$Y_1(s)/X(s) = (k_1 + bs)k_2/[(m_1s^2 + k_1 + bs)(m_2s^2 + k_1 + k_2 + bs) - (k_1 + bs)^2] \tag{9}$$

$$s^2Y_1(s)/X(s) = s^2(k_1 + bs)k_2)/[(m_1\, s^2 + k_1 + k_2 + bs)(m_2s^2 + k_1 + bs) - (k_1 + bs)^2] \tag{10}$$

Substituting Table 1 into (9) and (10) yield

$$Y_1(s)/X(s) = (2.4 \times 10^8 s + 2.56 \times 10^9)/(1.25 \times 10^4 s^4 + 4.675 \times 10^6 s^2 + 3.75 \times 10^5 s + 2.56 \times 10^8) \tag{11}$$

$$s^2Y_1(s)/X(s) = \frac{s^2(2.4\times10^8 s+2.56\times10^9)}{1.25\times10^4 s^4+4.675\times10^6 s^2+3.75\times10^5 s+2.56\times10^8} \tag{12}$$

2.1.2 Various Road Profiles

Various types of road profiles are engaged in the study of the suspension dynamics in relation to the vehicle's ride comfort. Overall, road disturbances that last for a short time and have a strong consequence have a negative effect on passenger ride comfort. The road disturbances are represented with basic sinusoidal functions for the dynamic response study of the vehicle suspension system on rough roads. In this study, road disturbances with single and dual bumps are applied as the input to the car suspension system for performance examination.

(1) Single Bump

The following equation is used to simulate a single bump road disturbance

$$SB(t) = \begin{cases} \frac{A}{2}\left(1 - \cos\left(2\pi\left(\frac{t}{T_{b1}}\right)\right)\right), & \text{for } T_{b1} \le t \le 2T_{b1} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

In (13), $A$ is the height of the bump and $T_{b1}$ is its duration. The duration of the bump is determined by the ratio of the bump length ($L$) to the vehicle velocity ($V$) and the bump disturbance forms between $T_{b1}$ and $2T_{b1}$.

In this scenario, the height ($A$) 0.05 m and the bump's length ($L$) of 5 m are taken into consideration. It is assumed that the vehicle is moving at a speed of 36 km/h. The vehicle is thought to be crossing the bump between the time interval of 0.5 and 1.0 seconds, as shown by the bump duration $T_{b1}$, which is computed as

$(L/V)$ and equals 0.5 seconds.

(2) Dual Bump

The road disturbance with two bumps of different magnitudes is designed using the following equation.

$$DB(t) = \begin{cases} \frac{A}{2}\left(1-\cos\left(2\pi\left(\frac{t}{T_{b1}}\right)\right)\right), & \text{for } T_{b1} \le t \le 2T_{b1} \\ \frac{B}{2}\left(1-\cos\left(2\pi\left(\frac{t}{T_{b2}}\right)\right)\right), & \text{for } 8T_{b2} \le t \le 9T_{b2} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

The first bump disturbance during the time interval between $T_{b1}$ and $2T_{b1}$ seconds has a height of $A$, and the second bump disturbance of height $B$ occurs during the time interval between $8T_{b2}$ and $9T_{b2}$ seconds.

The two bumps are 0.05 m in height ($A$) and 0.075 m in height ($B$), are taken into consideration. It is assumed that the car is travelling at a magnitude of 36 km/h and that the first and second bumps are 5 m and 6 m long, respectively. For the first bump, the time $T_{b1}$ is equal to the bump length divided by the velocity; in the case of the second bump, this equals 0.75 seconds. The first bump is being crossed by the car between the intervals of 0.5 and 1.0 seconds and 6.0 and 6.75 seconds.

### *2.2 Optimization of PID Controller*

Due to its effectiveness in various applications, PID type controllers remain the industry's most extensively employed control algorithms despite recent advancements in the control system. The availability of well-established rules for tuning the parameters of the controller and their relatively simple structures that can be easily implemented are key reasons why they are desirable in real time applications.

#### 2.2.1 PID Controller

In general, the transfer function of PID controller is given as the following:

$$C_{PID}(s) = K_p + K_i/s + K_d s \tag{15}$$

with $K_p$, $K_i$ and $K_d$ are the proportional gain, integral gain and derivative gain, respectively.

Various method can be implemented to tune these parameters including Ziegler-Nichols method such as described by Africa, Chua & Solis (2023), Meshram and Kanojiya (2012), Aisuwarya and Hidayati (2019) and Mazlan, Thamrin & Razak (2020) and Cohen-Coon method by Basu, Mohanty and Sharma (2016) and Utami, Yuniar, Giyantara & Saputra (2022). However, in this study, a particle swarm optimization algorithm is used to obtain the most optimum values of each of the gains.

#### 2.2.2 Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm who introduced by Kennedy and Eberhart (1995) is like a dance of nature, where each participant, or “particle,” learns from both their own experience and the group’s wisdom. Imagine a flock of birds or a school of fish gliding gracefully, each one adjusting its path by learning from its own best moves and the top performers of the group. This harmony allows them to home in on their target, much like how PSO particles fine-tune their positions to zero in on the ultimate solution. It’s a beautiful blend of individual insight and collective knowledge that guides them through a maze of possibilities to the most efficient answer. PSO’s brilliance shines across various fields, harnessing this shared intelligence to tackle complex challenges with finesse. In the world of Particle Swarm Optimization (PSO), each particle carries with it two essential qualities: velocity and position. Think of velocity as the particle’s inner compass, recalibrated through its personal journey and the shared stories of the swarm. Position, on the other hand, is the particle’s footprint in the vast dance hall of possibilities, shifted by the rhythm of its velocity. Together, these particles waltz through the many dimensions of the search space, united by a common quest to uncover the most exceptional solution—the global best—hidden within the swarm’s collective embrace.

The step-by-step algorithm for PSO based optimization is given below:

(1) First, the randomly generated population initializes the swarm's particles for the PID controller gains, setting their velocities and positions to a constant value.

(2) Set the iteration count to one.

(3) Calculate the fitness function values for each particle.

(4) Determine $P_{best}$ of each particle. If the particle’s fitness value from the current population is higher than

that of the matching previous population update $P_{best}$.

(5) Set the best fitness value from all previous iterations as $G_{best}$.

(6) Calculate the current value of inertial weight with respect to the current iteration value.

(7) Calculate the velocity of each particle for the next iteration. Limit the velocity within the determined boundary, $V_{max}$.

(8) Calculate the position of each particle for the next iteration.

(9) Increase the iteration count by one. If the iteration count is less than the maximum number, simply return to step 3.

The following are the formulas used in the algorithm and Table 2 listed the details of the parameters used in the formulas.

Table 2. Parameters of Particle Swarm Optimization Algorithm

| Parameters | Symbols | Values |
|---|---|---|
| Number of iterations | $iter_{max}$ | 100 |
| Number of particles | - | 30 |
| Number of variables | - | 3 |
| Upper bound of the search space | $x_{max}$ | [3500 5500 25] |
| Lower bound of the search space | $x_{min}$ | [0.1 0.1 0.1] |
| Cognitive component | $c_1$ | 2 |
| Social component | $c_2$ | 2 |
| Maximum value of inertia weight | $w_{max}$ | 0.9 |
| Minimum value of inertia weight | $w_{min}$ | 0.2 |
| Velocity clamping constant | $v_{clamp}$ | 0.2 |

(1) Linearly Decreasing Inertia Weight

In PSO, exploration is required to ensure that the algorithm able to thoroughly search the search space thus avoiding local convergence. Exploration is very important in the early stage of the process while in the latter stage, exploitation is preferable. Thus, a time varying inertia weight is used to ensure that the algorithm will explore the search space during the early iteration and exploitation by the end of the iteration. The following is the mathematical formula:

$$w = w_{max} - iter_{cur} \cdot (w_{max} - w_{min})/iter_{max} \tag{16}$$

where $w_{max}$ and $w_{min}$ are the maximum and minimum value of the inertia weight, respectively, $iter_{cur}$ is the number of current iterations and $iter_{max}$ is the number of iterations of the algorithm will execute.

(2) Velocity and Position of Particles

The velocity and the position of the particle will be updated for each iteration. (17) and (18) represent the formulas to obtain these values

$$v = w \cdot v_{cur} + c_1 r_1 (P_{best,x} - x_{cur}) + c_2 r_2 (G_{best,x} - x_{cur}) \tag{17}$$

$$x = x_{cur} + v \tag{18}$$

where $v$ is the new velocity, $x$ is the new position, $v_{cur}$ is the current velocity, $r_1$ and $r_2$ are random value [0,1], $P_{best,x}$ is the best personal position, $G_{best,x}$ is the global best position, $x_{cur}$ is the current position, $w$ is the inertia weight, $c_1$ is the cognitive component and $c_2$ is the social component.

(3) Velocity Clamping

One of the drawbacks of the standard PSO is the possibility of divergence due to the explosion of the velocity. This can cause the particle to move outside the pre-determined search space. To overcome this problem, the velocity is clamp to a certain value thus limit the particles movement to within the boundary of the search space. However, there is still a possibility of the particles will move outside the search space.

$$v_{max} = v_{clamp}(x_{max} - x_{min}) \tag{19}$$

where $x_{max}$ and $x_{min}$ are the upper and lower bound of the search space, respectively, $v_{max}$ is the maximum value of the velocity of a particle, $v_{clamp}$ is a properly chosen constant value (0,1] such that $v_{max}$ is not too large which causes particle to move in a very random manner thus skip the optimal solution but not too small that the particle would have very narrow search space which cause it to trapped in a local optimum.

(4) Fitness Function

The fitness function is one of the most important aspects of optimization. It quantifies the performance of a design solution as a single figure of merit, indicating how close the solution is to achieving the desired outcome.

In PSO, the particles traverse the search space with the goal of identifying the optimal solution. The incorporation of performance criteria such as the integral time absolute error (ITAE) into the algorithm guides the optimization process towards identifying the optimal set of controller parameters that enhance the control system performance. This enables the fine tuning and optimization of the control systems to achieve specific performance objectives based on the chosen criterion.

$$ITAE = \int_0^\infty t|e(t)|dt \tag{20}$$

where $e(t)$ is the error at time $t$.

**3. Results and Discussion**

Simulation based on the mathematical model for quarter car by using Matlab/Simulink software is performed. The PSO algorithm is implemented based on the parameter given in Table II. The optimized PID gains obtained from the algorithm are $K_p = 3499.8251$, $K_i = 5497.0601$ and $K_d = 2.4959$. The results of the optimized PID (PID-PSO) controller are compared with Ziegler-Nichols tuned PID (PID-ZN) controller and the passive suspension system.

*3.1 Single Bump*

The optimization of the PID controller using the PSO algorithm for a single bump disturbance, reveals substantial improvements in both body displacement and acceleration metrics. From Figure 2, the Active PID-PSO achieves a rise time of 0.3685 seconds, which is faster than the Active PID-ZN which is 0.4575 seconds but slower than the passive system's 0.1192 seconds. However, the Active PID-PSO significantly reduces overshoot to 1.492%, compared to 1.779% for the Active PID-ZN and 5.823% for the passive system. Additionally, the Active PID-PSO decreases the settling time to 5.553 seconds, which is shorter than the Active PID-ZN's 9.466 seconds but longer than the passive system's 4.486 seconds. This indicates that while the PSO-optimized controller is slightly slower to respond initially, it provides a more stable and controlled displacement response with reduced overshoot.

In terms of body acceleration as shown in Figure 3, the PSO-optimized PID controller shows even more pronounced benefits. The rise time for the Active PID-PSO is 0.0576 seconds, faster than both the Active PID-ZN which is 0.1314 seconds and the passive system's 0.0846 seconds. The overshoot for the Active PID-PSO is 23.59%, slightly higher than the Active PID-ZN's 16.24% but still comparable to the passive system's 21.41%. Importantly, the settling time for the Active PID-PSO is 3.709 seconds, which is shorter than both the Active PID-ZN's 5.150 seconds and the passive system's 4.513 seconds. These results highlight that the PSO algorithm effectively fine-tunes the PID controller parameters, resulting in quicker response times and reduced settling times for body acceleration, although with a marginal increase in overshoot.

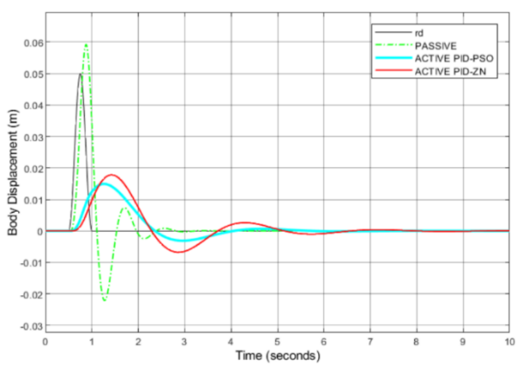


Figure 2. Body displacement response under single bump road profile

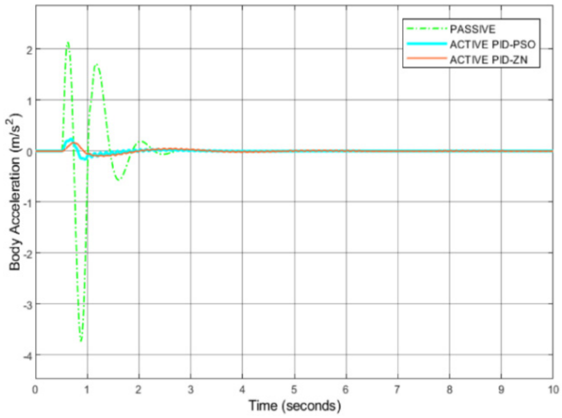


Figure 3. Body acceleration response under single bump road profile

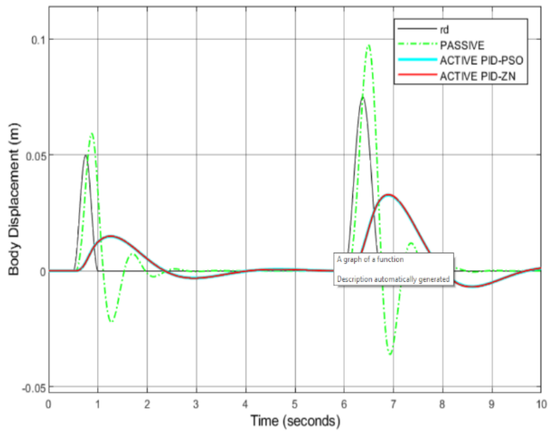


Figure 4. Body displacement response under dual bump road profile

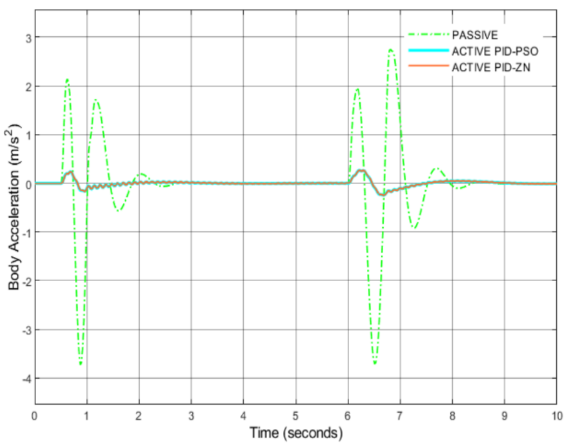


Figure 5. Body acceleration response under dual bump road profile

*3.2 Dual Bump*

The results shown in Figure 4 and 5, illustrate the performance of the optimized PID controller using the PSO algorithm for body displacement and acceleration in dual bump road profile. For body displacement, the PSO-optimized PID controller exhibits a rise time of 0.2914 seconds, a settling time of 5.553 seconds, and an overshoot of 3.257%. These values indicate a well-tuned system with a balance between quick response and stability. Compared to the active PID-ZN controller, which has a rise time of 0.2935 seconds, settling time of 5.277 seconds, and overshoot of 3.943%, the PSO-optimized controller demonstrates slightly faster response and significantly reduced overshoot. The passive system, while showing the quickest rise time of 0.1301 seconds and shortest settling time of 3.141 seconds, suffers from a much higher overshoot of 9.739%, highlighting its instability.

For body acceleration, the PSO-optimized PID controller achieves a rise time of 0.0363 seconds, a settling time of 4.213 seconds, and an overshoot of 26.58%. This indicates a moderate improvement over the active PID-ZN controller, which has a rise time of 0.0229 seconds, settling time of 4.101 seconds, and overshoot of 23.69%. The passive system again shows a faster rise time of 0.0772 seconds but at the cost of a significantly higher overshoot of 27.58% and a settling time of 3.876 seconds. The PSO-optimized controller, therefore, provides a more controlled and stable response, reducing excessive oscillations and improving the overall performance for dual bump road profile. This balanced performance, especially in managing overshoot, underscores the effectiveness of the PSO algorithm in optimizing PID controller parameters for varying road conditions.

## 4. Conclusion

In conclusion, the optimized PID controller using PSO algorithm offers a significant enhancement in car performance over various road profile. By providing quicker and more stable responses in both body displacement and acceleration, the PSO optimization improves ride comfort, making it a superior choice compared to both the passive system and the conventional Active PID-ZN controller. These results clearly show that the Active PID-PSO controller provides a superior driving experience, with faster response times, less overshoot, and quicker stabilization, making it the best choice for enhancing car stability and passenger comfort under various road disturbances.

To enhance the performance of the PID controller for the active suspension system, the focus will be on experimental validation by implementing the optimized controller on a physical vehicle or a high-fidelity simulator. This real-world testing will allow fine-tuning and adjustments based on practical considerations. Additionally, unforeseen challenges or limitations of the optimized controller will be identified through this validation process.

## Acknowledgments

The authors are grateful for the support by Center for Robotics and Industrial Automation, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM) in conducting this research and Ministry of Higher Education.

## References

Africa, A. D. M., Chua, J. O. Q., & Solis, J. L. H. (2023). PID tuning of speed controller using Ziegler-Nichols and manual method dc motor. *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)* (pp. 1-6). https://doi.org/10.1109/HNICEM60674.2023.10589041

Aisuwarya, R., & Hidayati, Y. (2019). Implementation of ziegler-nichols PID tuning method on stabilizing temperature of hot-water dispenser. *2019 16th International Conference on Quality in Research (QIR): International Symposium on Electrical and Computer Engineering* (pp. 1-5). https://doi.org/10.1109/QIR.2019.8898259

Azizi, M. J., Mirzaei, M., Rafatnia, S., & Mehrabani, E. B. (2024). Fuzzy-scheduled constrained control of active vehicle suspension system. *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)* (pp. 1-6). https://doi.org/10.1109/AISP61396.2024.10475260

Basu, A., Mohanty, S., & Sharma, R. (2016). Designing of the PID and FOPID controllers using conventional tuning techniques. *2016 International Conference on Inventive Computation Technologies (ICICT)* (pp. 1-6). https://doi.org/10.1109/INVENTIVE.2016.7824789

Bharali, J., & Buragohain, M. (2016). A comparative analysis of PID, LQR and fuzzy logic controller for active suspension system using 3 degree of freedom quarter car model. *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)* (pp. 1-5). https://doi.org/10.1109/ICPEICES.2016.7853152

Chen, X., Huang, Y., Na, J., Gao, G., & Zhao, J. (2021). Adaptive optimal control of nonlinear active suspension systems with completely unknown dynamics. *2021 33rd Chinese Control and Decision Conference (CCDC)* (pp. 3524-3529). https://doi.org/10.1109/CCDC52312.2021.9601936

Hurel, J., Amaya, J., Peralta, J., Alvarado, D., & Flores, F. (2022). Particle swarm optimization applied on fuzzy control: Comparative analysis for a quarter-car active suspension model. *2022 IEEE International Conference on Industrial Technology (ICIT)* (pp.1-8). https://doi.org/10.1109/ICIT48603.2022.10002809

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks* (pp. 1942-1948). https://doi.org/10.1109/ICNN.1995.488968

Lee, Y. J., Kang, S. W. Pae, Choi, H. D., & Lim M. T. (2022). Robust sampled-data output feedback T-S fuzzy control for vehicle active suspension systems via input delay approach. *2022 22nd International Conference on Control, Automation and Systems (ICCAS)* (pp. 716-719). https://doi.org/10.23919/ICCAS55662.2022.10003854

Mazlan, N. N. B. M., Thamrin, N. M., & Razak, N. A. (2020). Comparison between Ziegler-Nichols and AMIGO tuning techniques in automated steering control system for autonomous vehicle. *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)* (pp. 7-12). https://doi.org/10.1109/I2CACIS49202.2020.9140089

Meshram, P. M., & Kanojiya, R. G. (2012). Tuning of PID controller using Ziegler-Nichols method for speed control of dc motor. *IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM -2012)* (pp. 117-122).

Mohamed, A. H., Abidou, D., & Maged, S. A. (2021). LQR and PID controllers performance on a half car active suspension system. *International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)* (pp. 48-53). IEEE. https://doi.org/10.1109/MIUCC52538.2021.9447609

Nekoui, M. A., & Hadavi, P. (2010). Optimal control of an active suspension system. *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010* (pp. T5-60-T5-63). https://doi.org/10.1109/EPEPEMC.2010.5606776

Rao, T. R., & Anusha, P. (2013). Active suspension system of a 3 DOF quarter car using fuzzy logic control for ride comfort. *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)* (pp. 1-6). IEEE. https://doi.org/10.1109/CARE.2013.6733771

Sadeghi, M. S., Varzandian, S., & Barzegar, A. (2011). Optimization of classical PID and fuzzy PID controllers of a nonlinear quarter car suspension system using PSO algorithm. *2011 1st International eConference on Computer and Knowledge Engineering (ICCKE)* (pp. 172-176). https://doi.org/10.1109/ICCKE.2011.6413346

Semion, A., & Presnova, A. (2022). Optimal control of car active suspension control under delays. *2022 16th International Conference on Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference)* (pp. 1-4). https://doi.org/10.1109/STAB54858.2022.9807521

Talib, M. H. A., & Darus, I. Z. M. (2014). Development of fuzzy logic controller by particle swarm optimization algorithm for semi-active suspension system using magneto-rheological damper. *WSEAS Transactions on Systems and Control, 9*(1), 77-85.

Utami, A. R., Yuniar, R. J., Giyantara, A., & Saputra, A. D. (2022). Cohen-Coon PID tuning method for self-balancing robot. *2022 International Symposium on Electronics and Smart Devices (ISESD)* (pp. 1-5). https://doi.org/10.1109/ISESD56103.2022.9980830