Research Article GU J Sci 34(2): 517-527 (2021) DOI: 10.35378/gujs.762103

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

Particle Swarm Optimization Method Based Controller Tuning for Adaptive Cruise Control Application

Erhan OZKAYA[1], Hikmet ARSLAN[2], Osman Taha SEN[2,*]

[1]AVL Turkey Research and Engineering, 34885, Istanbul, Turkey
[2]Istanbul Technical University, Department of Mechanical Engineering, 34437, Istanbul, Turkey

Highlights

- PID parameter tuning with PSO technique.
- ACC system model embedded with PSO.
- Superiority of PSO method based PID parameter tuning process.

Article Info

Received: 01 July 2020
Accepted: 03 Nov 2020

Keywords

ADAS
ACC
PSO

Abstract

Major developments in relevant technology make the advanced driver assistance systems and autonomous driving functions more attainable. Thus, conventional practices being applied in vehicle production evolves towards highly automated, safer, and more comfortable vehicles. Although advanced driver assistance systems and autonomous driving functions have many advantages, such as enhanced driver convenience, increased comfort, and fuel economy; concerns related to safety still exist. For instance, failures related to sensors or algorithms being used can lead to critical problems. Therefore, controller algorithms should be more robust and well-optimized in order to eliminate these safety issues. This requires the implementation of automated optimization algorithms for the controller parameter tuning process. The main objective of this study is to optimize the designed controller for an adaptive cruise control system by using the particle swarm optimization method, which is a swarm intelligence optimization technique. Based on the results, it is observed that the use of automated optimization techniques for adaptive cruise control systems leads to better accuracy and safety for driving control. Furthermore, the time consumed for the controller parameter tuning process is also decreased significantly. In conclusion, the adaptive cruise control system requirements can be easily and accurately ensured by the use of particle swarm optimization algorithm.

1. INTRODUCTION

According to the data provided by Turkish National Police Academy, the number of accidents that involves road vehicles is about 10 million in Turkey during the last decade [1]. Furthermore, driver fault appears as the major cause of these accidents as evident from Figure 1, where approximately 90% of accidents is caused due to driver faults.

In order to minimize the effect of driver fault, enhance driving comfort and increase fuel efficiency; advanced driver assistance system (ADAS) features are being developed and integrated to modern vehicles in an increasing trend [2]. For example, Gürbüz and Buyruk [3] propose a new model that is used to calculate the safe stopping distance of a vehicle by considering factors due to driver, vehicle and environment. Authors suggest that this information can be displayed on a screen to driver as an assistant for enhancing safety [3]. Nevertheless, safety related concerns for ADAS features still persist. To avoid unexpected behaviors and eliminate these concerns for autonomous features, ADAS algorithms must be generated with robust logics in order to overcome all kinds of traffic scenarios in real life. Hence, optimization of algorithms becomes an important task which must be carefully undertaken.

*Corresponding author, e-mail:senos@itu.edu.tr

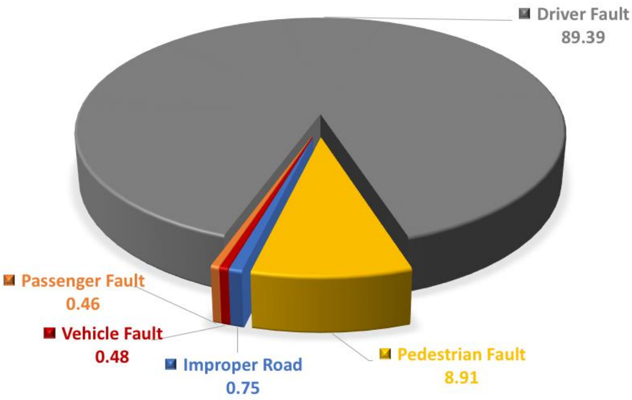


Figure 1. *Distribution of causes for vehicle accidents in Turkey from 2009 to 2018*

As an important ADAS feature, cruise control (CC) aims to keep the vehicle speed at the set value by controlling the engine throttle and brake actuators. Thus, the system usually requires an accurate mathematical model of the vehicle. Though, this could be cumbersome in some cases due to many existing uncertainties. Vedam et al. [4] develops a model free technique, where a complete information of about the vehicle is not required. Their approach determines the integral and derivative coefficients, while keeping the proportional gain intact. Furthermore, the authors suggest that the PID controller parameters can be altered in real time by coupling this approach with a proper algorithm. As an extension to CC, Adaptive Cruise Control (ACC) aims to regulate the speed of a vehicle with respect to a lead vehicle. The system utilizes a range sensor to detect the preceding vehicle and determines the speed of the lead vehicle and relative distance between two vehicles in order to set the vehicle speed [2]. In one particular paper by Jiang et al., authors develop a stochastic optimal control algorithm that can diversify the controller behavior of the ACC system based on driver characteristics [5]. Future trends go towards highly automated and cooperative systems with no driver intervention.

Controller parameter optimization problem for ACC algorithms have been extensively studied by many researchers. In one study, Rout et al. utilize genetic algorithm method for the optimization of PID controller parameters used in an intelligent speed assist system [6]. Authors adopt the transfer function approach in order to model the dynamics of the vehicle on which the CC algorithm is embedded [6]. Results of two systems are compared in terms of maximum overshoot, peak time, rise time, settling time and steady state error and it is observed that the model with controller whose parameters are tuned with genetic algorithm provides superior results over the conventional PID controller. In another study, Abdulnabi [7] develops a transfer function based vehicle model equipped with a CC algorithm. Here, author compares the performance of a PID controller that is tuned with particle swarm optimization (PSO) technique to other predesigned controllers, and observes that the former provides better response in terms of maximum overshoot, peak time, rise time, settling time. In yet another paper, antlion optimization technique is used for PID controller parameter tuning of a CC system [8]. Again, the results of the proposed approach are compared to others (conventional PID, state space method, fuzzy logic and genetic algorithm), and it is claimed that the controller with parameters tuned with antlion optimizer provides superior performance. As evident from the literature review, algorithms used for controller parameter tuning increase the accuracy of the controller and optimum controller parameters can be obtained promptly.

The chief objective of this study is to implement PSO algorithm for the tuning of PID parameters of an ACC system while meeting the requirements related to safety and performance. Furthermore, several voids in the related literature [6-8] are aimed to overcome. First, a single degree of freedom vehicle model incorporating a gearbox subsystem is developed as opposed to the transfer function based modeling approach. Second, the ADAS system is assumed as ACC, not CC as in the prior papers [6-8]. Consequently, the objectives of this study are: 1) to build a one-mass longitudinal vehicle dynamics model integrated with an ACC controller; 2) to develop an ACC controller parameter tuning algorithm based on PSO method; and 3) to compare the performance of PSO method with conventional tuning techniques.

2. OPTIMIZATION OF ADAPTIVE CRUISE CONTROL PARAMETERS

In order to optimize ACC parameters, a vehicle model and an ACC controller algorithm are developed. In this section, the ACC algorithm is first explained. Then, the details of the one-mass vehicle model are given. Subsequently, PSO algorithm is integrated into the model which includes ACC algorithm and vehicle model.

2.1. Adaptive Cruise Control Algorithm

Adaptive Cruise Control structure has two main layers which can be named as high and low level. High level loop calculates the desired wheel torque to keep the vehicle at desired conditions, while the low level loop is assumed as a first order system [9]. The high level loop is controlled by a PD controller with a time gap. The time gap that is set by the driver is employed to calculate the desired distance d_{des} as shown in Equation (1), where τ_{set} refers to the time gap between the lead and ego vehicles, and V_{ego} represents the speed of the ego vehicle. Here, it should be mentioned that the ego vehicle is the one equipped with ACC.

$$d_{des} = V_{ego}\tau_{set} \ . \tag{1}$$

Desired distance error and vehicle speed error are the main control inputs of the system. The main equation of the ACC controller is shown in Equation (2):

$$T_{wheel,des} = K_v\left(V_{lead} - V_{ego}\right) + K_t(d_{act} - d_{des}) \ , \tag{2}$$

where K_v and K_t are the main controller parameters, V_{lead} refers to the lead vehicle speed and d_{act} is the actual relative distance between the lead and ego vehicle that is calculated with Equation (3):

$$d_{act} = \int V_{lead}dt - \int V_{ego}dt \ . \tag{3}$$

The simulations are initialized with 0- and 40-meter positions for the ego and lead vehicles, respectively. Thus, the initial relative distance between lead and ego vehicles is assumed as 40 meters.

2.2. Vehicle Modeling

The ego vehicle is assumed as a passenger car with a 1.4-liter diesel engine and a 6-speed transmission, whose related data is adopted from a mass-produced road vehicle. Based on the data available for this vehicle (engine torque curve at full load, transmission ratios at all gears, final gear ratio and wheel radius), corresponding traction force curves at each gear are evaluated and depicted in Figure 2.

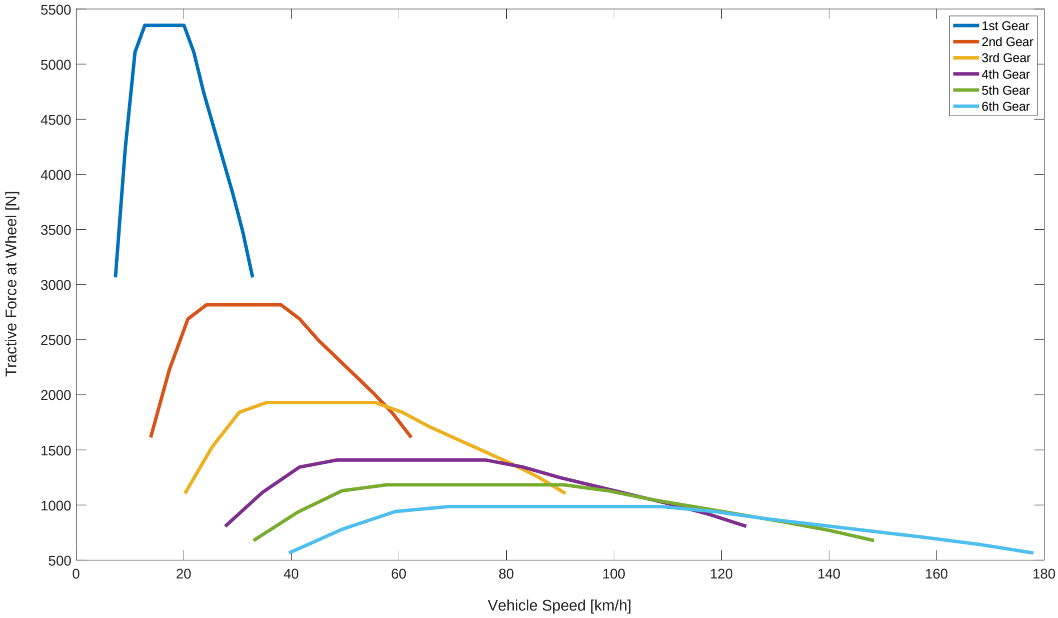


Figure 2. *Traction force curves for the forward speeds of the vehicle*

The road-tire interaction is defined with the Pacejka model, where the longitudinal traction forces are assumed as the function of longitudinal slip. Longitudinal slip for braking and traction cases are expressed below in Equations (4) and (5), respectively:

$$s = 1 - \frac{R_0 \omega_{wheel}}{V} \; , \tag{4}$$

$$s = 1 - \frac{V}{R_0 \omega_{wheel}} \; . \tag{5}$$

Here, R_0 and ω_{wheel} corresponds to the dynamic radius of the tire and its angular velocity, respectively. The Pacejka model for longitudinal tractive forces is given in Equation (6):

$$F_x = A \sin\left(B \tan^{-1}\left(Cs - D\left(Cs - \tan^{-1}(Cs)\right)\right)\right) , \tag{6}$$

where the parameters B, C and D are referred to as shape factors and these parameters are functions of slip angle, slip ratio, camber angle and wheel force. Furthermore, A is the maximum value of the longitudinal force. These parameters are all determined experimentally [10]. Resistive forces are integrated into model as a combination of aerodynamic resistance, rolling resistance and grade resistance. Finally, the dynamics of the vehicle in longitudinal direction is defined as in Equation (7), where ρ is the air density, C_d is the aerodynamic drag coefficient, f_r is the rolling resistance coefficient, θ is the angle of inclination of the road and A_f and m are the cross sectional area and the mass of the vehicle, respectively. Furthermore, the term F_{Accel} in Equation (7) is the inertial resistance which is due to the longitudinal acceleration of the vehicle.

$$F_{Accel} = F_x - \frac{1}{2}\rho C_d A_f V^2 - mgf_r - mg\sin(\theta) \; . \tag{7}$$

The dynamic model of the vehicle with ACC algorithm is built and depicted in Figure 3 in block diagram form.

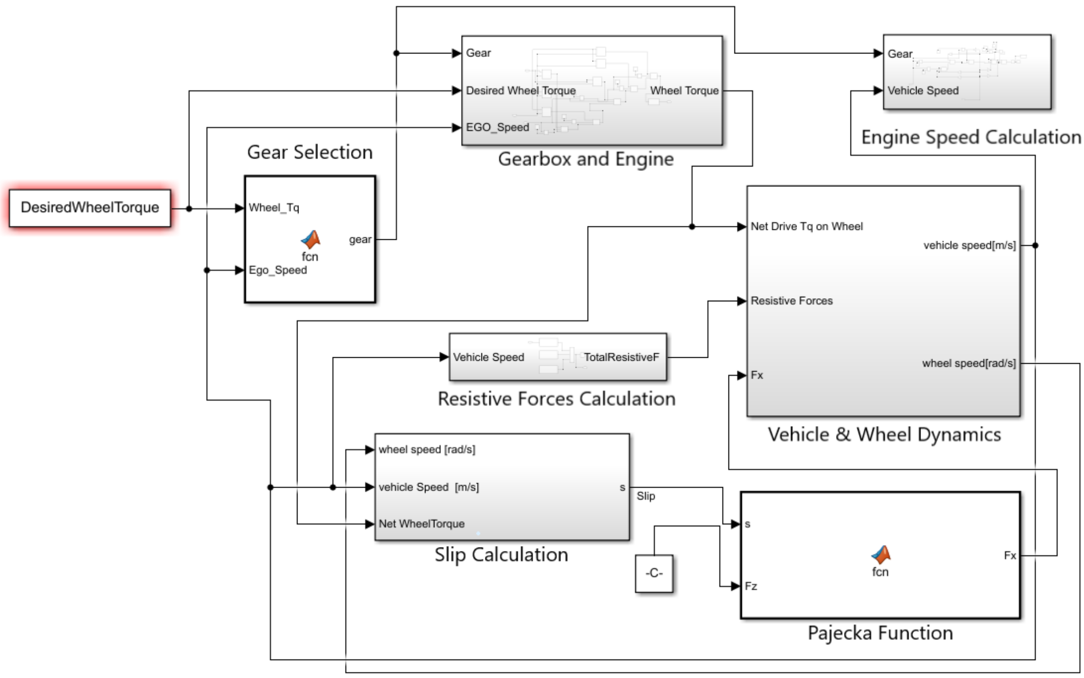


Figure 3. *Vehicle model and its subsystems*

2.3. Controller Parameter Tuning with Particle Swarm Optimization

Most of the optimization methods are inspired from a natural event such as foraging. Species such as insects, birds, etc. have to forage to survive. During the foraging behavior, the members of the swarm should communicate with each other and the basic objective is to reach the food as quickly as possible [11]. Thus, the objective in the optimization process is to reach a desired value as fast as possible. The discipline which employs the collective behavior of a social group is referred to as swarm intelligence. The Particle Swarm Optimization technique is a member of swarm intelligence-based optimization methods and developed by Kennedy and Eberhart in 1995 [12]. The main principle behind PSO is that every member of the swarm sets its position according to the best position in the group. During this process, there are two significant elements, which are position and velocity. The position of the particle is labeled as x_s^i, where the superscript i and subscript s refer to the iteration and particle numbers, respectively. Furthermore, the velocity of a particle is shown with $V_s{}^i$. The initial positions of each particle in the swarm are determined randomly as shown in Equation (8):

$$x_s^i = rand(1, n_{var})\ . \tag{8}$$

The velocity of each particle in the swarm are set to zero initially, and they are evaluated at each iteration according to Equation (9). In this equation the parameter $P_{best_s}{}^{pos}$ represents the best position of each swarm particle at an iteration and $G_{best}{}^{pos}$ is the global best position of the entire swarm. Furthermore, the coefficients c_1, c_2 and β determines the characteristics of the algorithm. Here, the coefficients c_1 and c_2 are known as acceleration constants and the parameter β refers to as an inertia factor. In this study, linearly decreasing inertia weight method is used, and the lower and upper bounds for β are set to as 0.4 and 0.9, respectively.

$$V_s^{i+1} = \beta V_s^i + c_1 rand(1, n_{var})\left(P_{best_s}{}^{pos} - x_s^i\right) + c_2 rand(1, n_{var})(G_{best}{}^{pos} - x_s^i)\ . \tag{9}$$

The position of each swarm particle is updated at each iteration according to $x_s^{i+1} = x_s^i + V_s^{i+1}$. Furthermore, the flowchart of the PSO algorithm that is implemented in this study is shown in Figure 4.

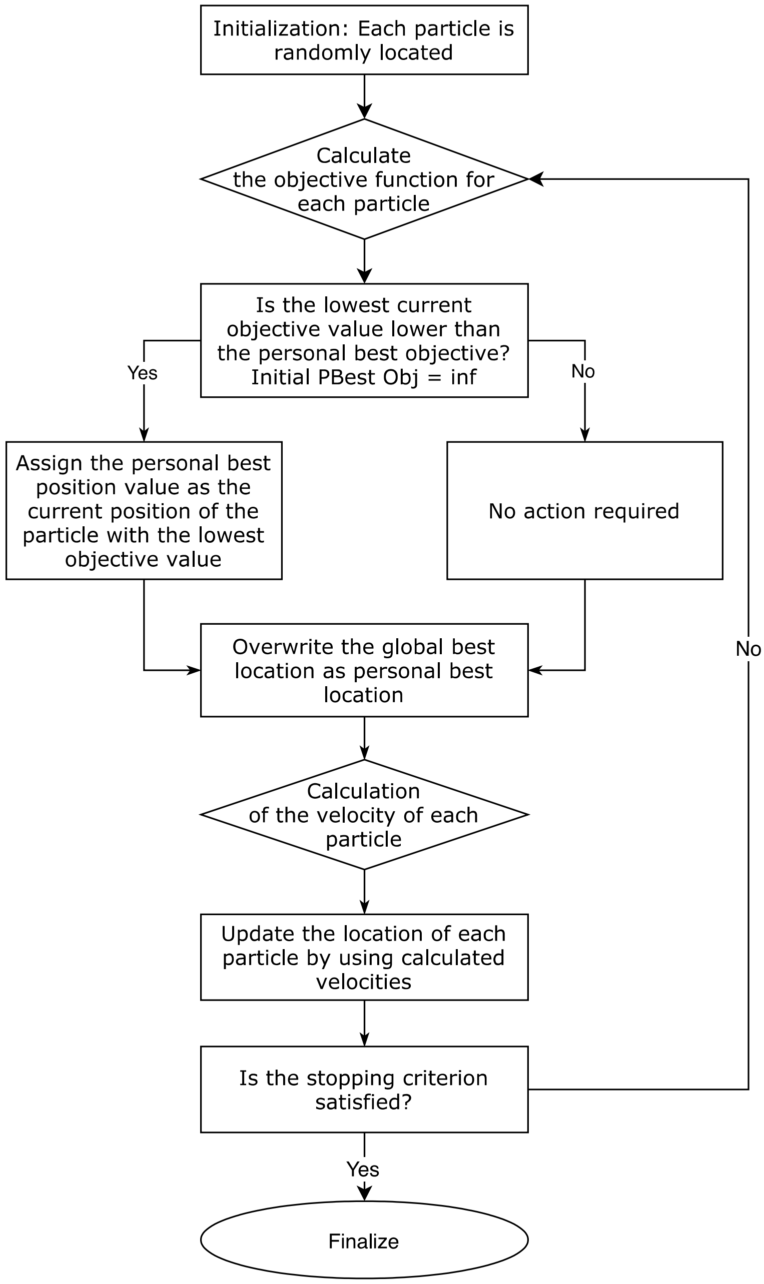


Figure 4. *Flowchart of the controller parameter tuning process with particle swarm optimization method*

Personal best and global best values are determined by comparing the objective values of each swarm particle. With the initialization of the algorithm, each particle is assigned a personal best value that is essentially its best objective value, and the personal best value is updated at each iteration. In terms of the global best value, it is also calculated at each iteration step and it is essentially the best objective value of all particles at a given iteration step. It is again updated at all iterations accordingly, until the maximum number of iterations is reached.

The speed trace of the lead vehicle is generated computationally with AVL VSM [13], which is a comprehensive tool for simulating activities of complex vehicle dynamics. This tool requires a detailed information about the lead vehicle such as tire size, coast down parameters, suspension parameters, driveline, engine and gearbox characteristics.

In order to tune the controller parameters, the vehicle model, PSO and ACC algorithms are combined. Though, the objective function is first determined. In this study, the objective function is chosen as integrated absolute error (IAE), which can be described as the area under the absolute error curve. The objective function is shown in Equation (10):

$$IAE_{ACC} = \int_0^{t_{tot}} |t_{set} - t_{act}| dt \ . \tag{10}$$

The model flow of the system where the vehicle model, PSO and ACC subsystems are combined is shown below in Figure 5.

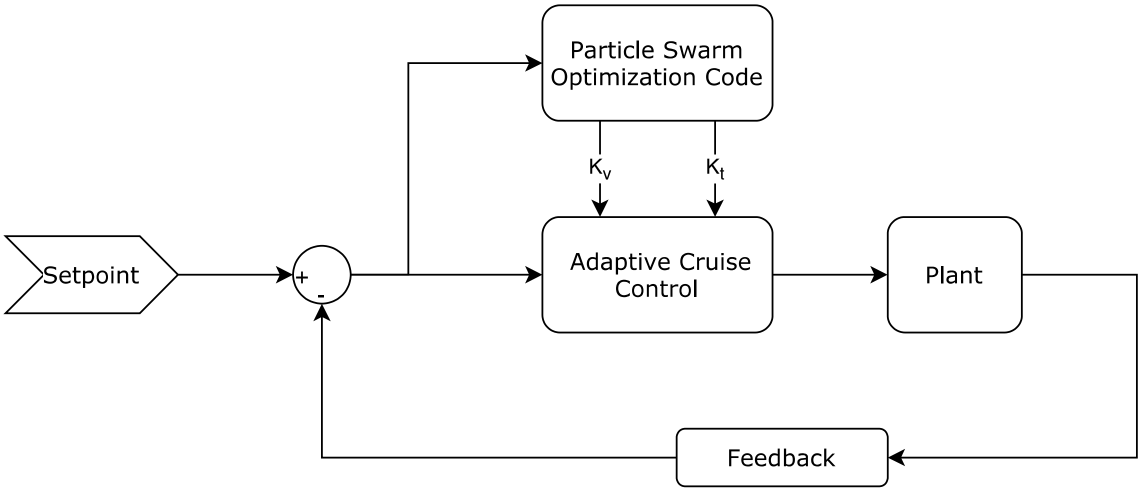


Figure 5. *Block diagram of the entire system with corresponding subsystems: vehicle model (shown as Plant), adaptive cruise control algorithm, and particle swarm optimization process*

3. SIMULATION RESULTS

The results of the simulations are depicted in figures given below. Notice that the initial inter-vehicle distance in simulations is set as 40 m as mentioned before, and the lead vehicle accelerates from 0 to 60 km/h, while the ego vehicle follows the lead vehicle based on the operation of its ACC algorithm. These operational parameters are selected arbitrarily, and they do not affect the performance of the proposed controller parameter tuning algorithm. Simulations are run at two different controller parameter sets. In the first parameter set, the controller parameters are assumed as $K_v = 10$ and $K_t = 10$, and these are referred to as base parameters. In the second set, the parameters K_v and K_t are optimized with the PSO method. Note that the velocity time histories for lead vehicle, ego vehicle with base controller parameters and ego vehicle with optimized controller parameters are shown in Figure 6. Observe that the ego vehicle with optimized controller parameters ($K_v = 286.3$ and $K_t = 27.2$) approaches the objective, which is a constant time gap, faster than the ego vehicle with base controller parameters. Furthermore, the speed response of the ego vehicle with base controller parameters is oscillatory, which would not be a desired response.

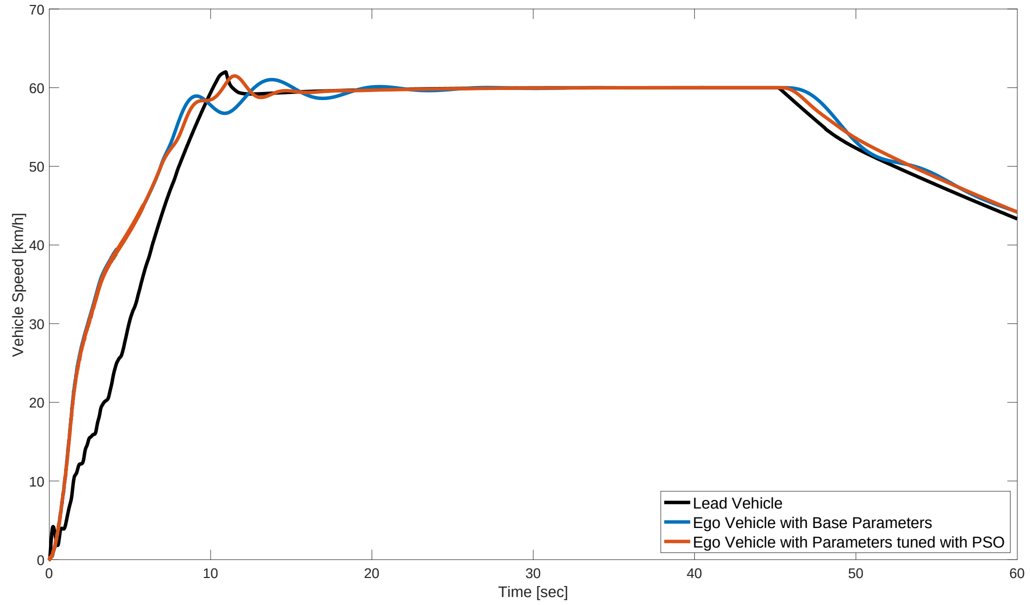
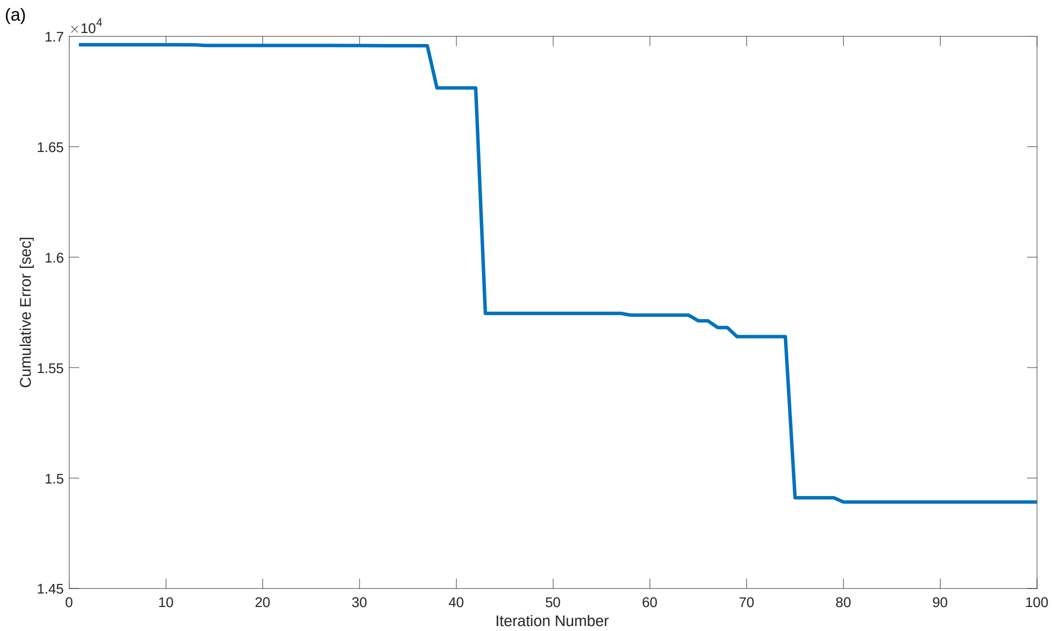


Figure 6. *Simulation results in terms of vehicle speed time histories*

Figure 7(a) demonstrates the change of cumulative error (difference between the desired and actual time gaps) with respect to iteration number of 100 iterations and a swarm of 3 particles. As seen, the solution converges to the desired solution in about 80 iterations. The time history of the time gap between lead and ego vehicles is shown in Figure 7(b). Notice that the time gap quickly reaches to the objective value, which was set to be 1 second, and remains at this value for the entire event. Furthermore, the time gap does not go below the 0.8 second value, which is assumed as a critical limit according to ISO 15622:2018 standard [14].



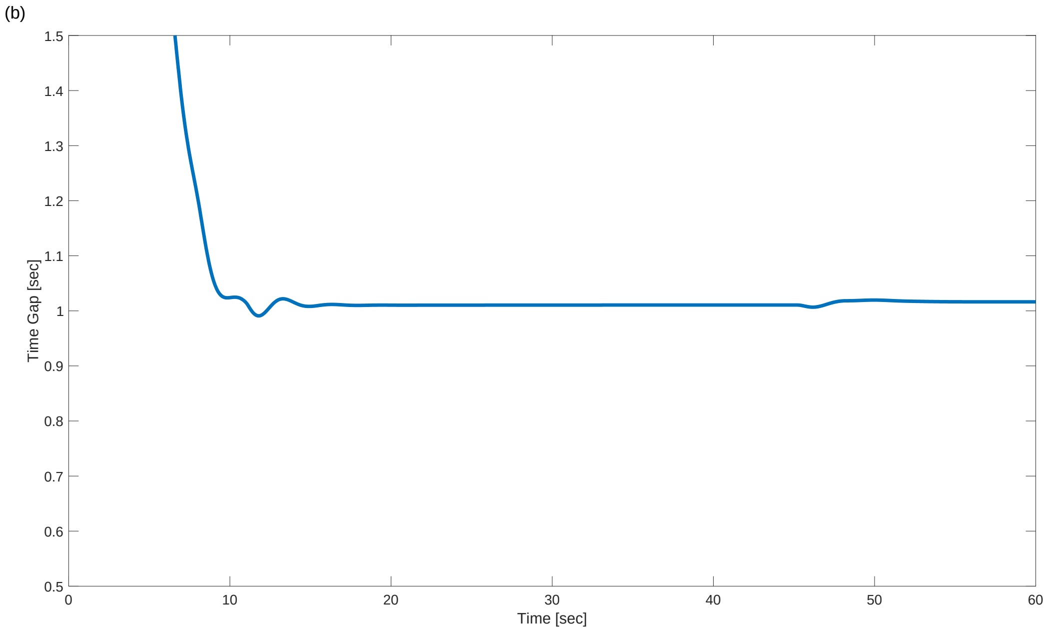


Figure 7. *a) Cumulative error with respect to iteration number b) Actual relative time gap between lead and ego vehicles*

The time histories for the actual and desired inter-vehicle distances are depicted in Figure 8. Observe that the desired inter-vehicle distance is not constant due to the time varying speed profile of the lead vehicle, and it is calculated based on the 1 second time gap objective. As in Figure 7(b), the ego vehicle reaches the desired inter-vehicle distance in about 10 seconds and successfully follow the lead vehicle.

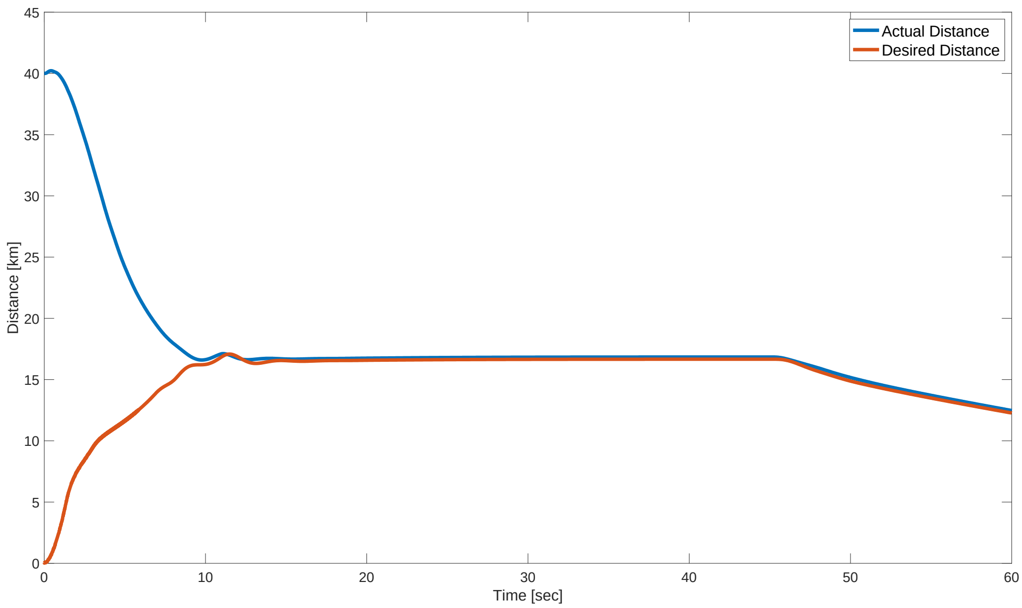


Figure 8. *Actual and desired relative distances between lead and ego vehicles*

4. CONCLUSION AND FUTURE RECOMMENDATIONS

In this study, an adaptive cruise control system is designed and combined with a one-mass longitudinal vehicle model that includes all resistive forces (aerodynamic, rolling, and grade resistances), a tire model (Pacejka magic formula), and a powertrain model. Furthermore, the PID controller parameter tuning is performed by the particle swarm optimization method. Finally, the performance of the adaptive cruise control system is compared for optimized and initial controller parameter sets. According to the results, it is observed that the parameters obtained with particle swarm optimization method leads to a better performance in terms of keeping constant time gap between lead and ego vehicles. Also, it is observed that the optimum parameter set for the PID controller can be obtained rapidly with the particle swarm optimization method. Hence the ego vehicle, whose controller parameters are optimized with particle swarm optimization method starts to cruise at the desired speed profile promptly. This increases the robustness of the controller system, which is a key requirement for calibration activities. Therefore, the capability of the controller expands for more complex structures.

ACKNOWLEDGEMENT

Support from AVL Turkey is gratefully acknowledged by the first author.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Yaprak, Ş., Akbulut, A. M., “Trafik kaza ve denetim istatistikleri”, Polis Akademisi Yayınları, Rapor No: 27, (2019).

[2] Xiao, L., Gao, F., “A comprehensive review of the development of adaptive cruise control systems”, Vehicle Systems Dynamics, 48(10): 1167-1192, (2010).

[3] Gürbüz, H., Buyruk, S., “Improvement of safe stopping distance and accident risk coefficient based on active driver sight field on real road conditions”, Intelligent Transport Systems, 13(12): 1843-1850, (2019).

[4] Vedam, N., Diaz-Rodriguez, I., Bhattacharyya, S.P., “A novel approach to the design of controllers in an automotive cruise-control system”, 40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, Texas, USA, (2014).

[5] Jiang, J., Ding, F., Zhou, Y., Wu, J., Tan, H., “A personalized human drivers' risk sensitive characteristics depicting stochastic optimal control algorithm for adaptive cruise control”, IEEE Access, 8: 145056-145066, (2020).

[6] Rout, M. K., Sain, D., Swain, S. K., Mishra, S. K., “PID controller design for cruise control system using genetic algorithm”, International Conference on Electrical, Electronics and Optimization Techniques, Chennai, India, (2016).

[7] Abdulnabi, A. R., “PID controller design for cruise control system using particle swarm optimization”, Iraqi Journal for Computers and Informatics, 43(2): 29-34, (2017).

[8] Pradhan, R., Majhi, S. K., Pradhan, J. K., Pati, B. B., “Antlion optimizer tuned PID controller based on bode ideal transfer function for automobile cruise control system”, Journal of Industrial Information Integration, 9: 45-52, (2018).

[9] Luo, L. H., Liu, H., Li, P., Wang, H., “Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following”, Journal of Zhejiang University, 11(3): 191-201, (2010).

[10] Jazar, R. N., “Vehicle dynamics: theory and application”, Springer, New York, NY, (2008).

[11] Erdoğmuş, P., “Particle swarm optimization with applications”, IntechOpen, London, (2018).

[12] Özsağlam, M. Y., Çunkaş, M., “Optimizasyon problemlerinin çözümü için parçacık sürü optimizasyonu algoritması”, Journal of Polytechnic, 11(4): 299-305, (2008).

[13] Anstalt für Verbrennungskraftmaschinen List. https://www.avl.com/-/avl-vsm-4-, 10.11.2019.

[14] BS ISO 15622:2018., “Intelligent transport systems – adaptive cruise control systems – performance requirements and test procedures”, (2018).