

The Adaptive Cruise Control for Curved Roads Using Archived Crow Search Algorithm

Dhidik Prastyanto^{*}, Esa Apriaskar, Subiyanto Subiyanto, Imam Khoirul Akbar, Ilham Ari Prastyo, Viyola Lokahita Bilqis, Dimas Alfarizky Ilham



Department of Electrical Engineering, Universitas Negeri Semarang, Semarang 50229, Indonesia

Corresponding Author Email: dhidik.prastyanto@mail.unnes.ac.id

Copyright: ©2024 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijtdi.080117>

Received: 6 January 2024

Revised: 25 February 2024

Accepted: 18 March 2024

Available online: 31 March 2024

Keywords:

automated vehicle, adaptive cruise control, crow search algorithm, curved road, steering control

ABSTRACT

One of the advanced driver assistance systems (ADAS) technologies that can address the issue of high-traffic accidents is adaptive cruise control (ACC). However, a challenge arises due to the lack of control algorithm development in ACC technology that accommodates curved road conditions. This paper proposes a comprehensive solution by introducing ACC for curved roads through the utilization of a multidimensional control system model. This paper aims to implement the crow search algorithm (CSA) into the ACC technology: (1) Our objective is to apply the original crow search algorithm (OCSA) to find the most optimal values for the parameters v_{err} , x_{err} , v_x of ACC, and k_p and k_i of lateral displacement control; (2) We also implement the archived crow search algorithm (ACSA) into the control system, which is considered to have faster computation time than OCSA. Based on the obtained results, ACSA demonstrates faster computation time. The optimal values for achieving enhanced performance are found to be k_p at 0.7492, k_i at 0.6506, v_{err} at 0.9716, x_{err} at 0.9778, and v_x at 0.7012. This model was developed using MATLAB and compared to the non-optimized version. The research aims to contribute to ADAS development by addressing the optimization challenges of control algorithms for ACC parameters on curved roads. Ultimately, this solution enhances driver safety by providing more effective control in challenging road conditions.

1. INTRODUCTION

Safety of the vehicle is an important matter that merits global discussion since the rates of traffic accidents are quite high. According to WHO [1], each year road accidents cause over 1 million fatalities and 20-50 million injuries [2], which makes traffic accidents a significant issue. The solution to reducing accident rates is by developing the features of ADAS technology to support drivers while driving a vehicle [3]. One of the ADAS technologies that can solve the issue is ACC which can support vehicles in maintaining a safe distance and reducing accidents [4]. By utilizing a combination of sensors such as radar, lidar, cameras, and advanced algorithms, the system can intelligently detect and monitor the movement of vehicles in front of it, automatically adjusting the car's acceleration and deceleration [5]. The implementation of ACC which is typically used in Automated Vehicle (AV) or semi-automated vehicles (SAV), enhances driver and passengers' comfort, as well as avoids accidents [6].

With such usefulness, ACC was developed by many researchers to be implemented in vehicles to ensure the safety and convenience of the drivers. Drivers will not feel excessively fatigued while driving their vehicles if ACC is applied. The utilization of ACC has shown sophisticated results in maintaining a safe distance for drivers, as demonstrated by the study [7, 8]. This feature significantly

amplifies the level of safety and convenience experienced by the driver. A broadly similar point has also been demonstrated in the study [9] using PID control.

Some studies suggest the use of artificial intelligence in ACC control, showing promising results comparing fuzzy and neural network-based controls as mentioned in the study [10]. When compared to conventional PID control, AI-based control outperforms as presented in the study [11]. Other evidence to show the promising results of using AI-based technique for the case of ACC have also been conducted by the study [12, 13], utilizing metaheuristic optimization techniques genetic algorithm (GA) and particle swarm optimization (PSO), respectively. Unfortunately, those reported works are still limited to straight-road case studies, while in fact, there is a possibility to vehicles experiencing a curved road.

With these limitations, several researchers have begun developing controllers to address curved roads, employing integrated vehicle dynamics control (IVDC) in the study [14]. In other work, researchers explored the same case but adopted AI in control; Symonds et al. [15] harnessed artificial intelligence to control steering, devising an effective design to track road dynamics. Fuzzy logic was employed by Dahmani et al. [16] to address curved roads using Takagi-Suseno fuzzy model. Implementing a different concept, Zhang et al. [17] incorporated driving behavior to tackle curved roads through

adaptive artificial intelligence. Meanwhile, Wang et al. [18] employed lane detection to provide warnings for curved road segments, aiming to prevent potential accidents.

None of the above literatures dealing with the curved road control problem utilized any metaheuristic optimization techniques. As previously mentioned, and highlighted in the [12, 13] about the potential use of metaheuristic optimization for straight-road case of ACC control, it can be a new research direction to develop a metaheuristic optimization technique but for the case of curved road control problem. It emerges as a non-trivial problem since ACC for the curved road will not only need to adjust the vehicles' speed, but also to synchronize with the steering control [7].

The paper is structured as follows. Section 2 presents a description of the ACC system model, exploring controller design and parameter optimization methods using the ACSA on automated vehicles for curved road scenarios. Additionally, this section provides an explanation of steering control. Section 3 discusses simulation results, including graphical comparisons between the optimization results using ACSA, original crow search algorithm (OCSA), and default settings. Finally, Section 4 concludes the paper by summarizing the experimental findings, implications, and outlining directions for future research.

2. OVERVIEW OF SIMULATION DESIGN

Based on Figure 1, the system is composed of two different control systems, adaptive cruise control (ACC) and steering control, which are continuously used to regulate the vehicle's behavior in order to provide comfort for the driver and enhance safety. The first subsystem, ACC, has the function of controlling the longitudinal speed of the vehicle on the road. Its role is to observe and respond to the movements of the vehicles in front of it to adjust the longitudinal speed to maintain a safe following distance [19]. This helps reduce the risk of potential accidents and relieves the driver's workload. On the other hand. The second subsystem, steering control, is responsible for controlling the lateral displacement of the vehicle when navigating curved roads. In this situation, steering control calculates the optimal steering angle based on the road curvature and the vehicle's speed [20]. This ensures that the vehicle can navigate curves smoothly and safely. The combination of these two subsystems allows the vehicle to operate more safely and efficiently [14].

As illustrated in Figure 1, the vehicle equipped with ACC and steering control is referred to as the ego car, while the vehicle in front is named the lead car. With steering control, the ego car can follow the centerline of the road, whereas the lead car, without steering control, still stays within the curved lane. CSA aims to optimize both ACC and steering control concurrently, seeking the global optimal solution, thereby

enhancing the overall system performance. For the test, the curve radius of curvature configuration used is 760 meters with linewidth 4 meters.

To simulate vehicle movement on a curved road used vehicle dynamics on MATLAB to create a mathematical model of the vehicle that includes dynamic properties such as mass, mass distribution, inertia, suspension, and braking. The equations of motion for the vehicle, encompassing forces like gravity, friction, and aerodynamics, are numerically integrated within the MATLAB environment. Key parameters of the vehicle, such as mass and friction coefficient, play a crucial role in determining its dynamic characteristics. In this paper, the vehicle dynamics used come from the vehicle dynamics subsystem models the vehicle dynamics with the bicycle model - force input block from the automated driving toolbox.

The dynamics of the lead car and ego car equations transform the vehicle's acceleration and steering into its actual position, yaw angle, lateral velocity, and longitudinal velocity. The dynamic equation for both cars can be expressed in Eq. (3). As written in Eq. (3), respectively, V_y is vehicle lateral velocity, ψ is vehicle yaw angle, ψ' is vehicle yaw angle rate, V_x is vehicle longitudinal velocity, V'_x is vehicle longitudinal acceleration, m is mass of the vehicle, l_f is longitudinal distance from center of gravity to front tires, I_z is yaw moment of inertia of vehicle, I_r is longitudinal distance from center of gravity to rear tires, τ is longitudinal time constant, C_f is cornering stiffness of front tires, and C_r is cornering stiffness of rear tires. The values for those vehicle parameters are written in Table 1.

Table 1. Vehicle parameter

Parameter	Value
m	1575 kg
I_z	2875 m×N×s ²
l_f	1.2 m
I_r	1.6 m
C_f	19000 N/rad
C_r	33000 N/rad
τ	0.5 N/A

The results obtained from the vehicle dynamics, including quantities such as longitudinal velocity V_x and lateral velocity V_y , are originally represented in a reference frame tied to the vehicle's body. To determine the path taken by the vehicle, these body-fixed coordinates are transformed into global coordinates using the following relationships, given by Eq. (1) and Eq. (2):

$$X = V_x \cos(Y) - V_y \sin(Y) \quad (1)$$

$$Y = V_x \sin(Y) + V_y \cos(Y) \quad (2)$$

$$\frac{d}{dt} \begin{bmatrix} V_y \\ \psi \\ \psi' \\ V_x \\ V'_x \end{bmatrix} = \begin{bmatrix} -\frac{2C_f + 2C_r}{mV_x} & 0 & -V_x - \frac{2C_f l_f - 2C_r l_r}{mV_x} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\frac{2C_f l_f - 2C_r l_r}{I_z V_x} & 0 & -\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z V_x} & \frac{2C_f}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \delta + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

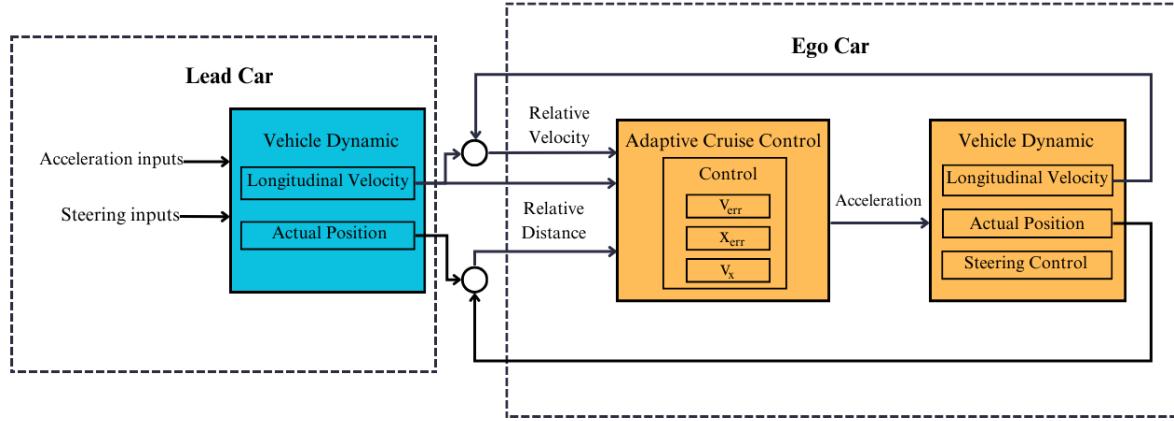


Figure 1. Schematic model of the system

2.1 System model and ACC

The ego vehicle is equipped with a sophisticated ACC system. This ego vehicle is also equipped with various sensors that can measure and monitor its distance from the vehicle in front of it, which is referred to as the main vehicle. The ACC system aims to ensure a safe distance from the vehicle in front by controlling the speed of the ego vehicle with a fast response time. It utilized sensor data to constantly gauge the gap between the lead vehicle and the ego vehicle, making necessary adjustments to ensure a safe distance. Figure 2 illustrates the case of ACC for a curved road discussed in this research.

As illustrated in Figure 3, ACC has two modes, namely speed control and distance control. These two modes change based on the relative distance (D_R) and safe distance (D_{safe}). Relative distance is the distance between the ACC vehicle and the car in front. Meanwhile, the safe distance is the safe distance between two vehicles. Speed control mode is active when the relative distance is greater than or equal to the safe distance because the goal of this mode is to match the maximum speed set by the user (V_{set}). Typically, the setpoint for ACC speed ranges from 30 km/h to 100 km/h. Conversely, distance control mode is active when the relative distance is below the safe distance with the aim of achieving a safe distance between the two vehicles. Generally, ACC will enter the distance control mode first until it approaches the safe distance, then enter the speed control mode to match the setpoint.

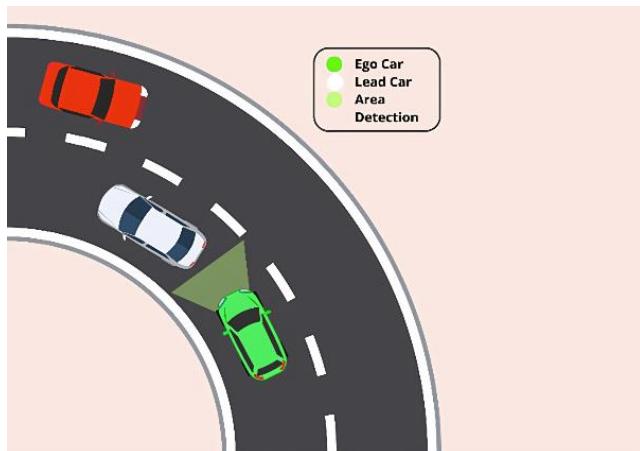


Figure 2. Illustration of ACC in curved road

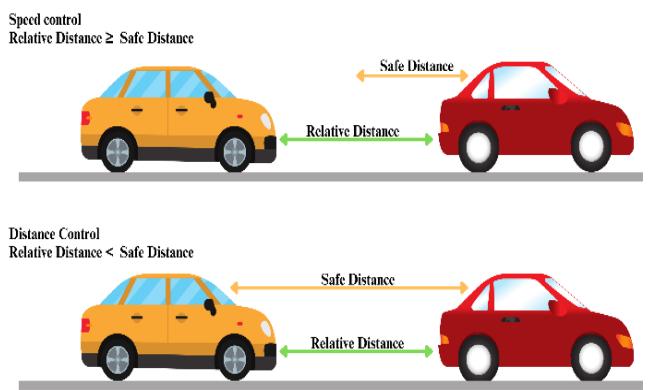


Figure 3. Two control mode in ACC

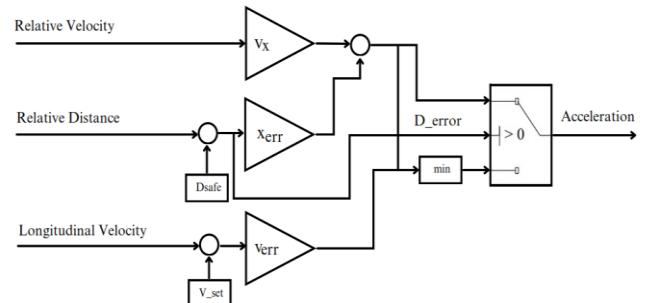


Figure 4. Control system diagram of ACC

The controller design is carried out to integrate the CSA into ACC so that it can operate in a previously prepared simulation. This controller design stage involves several important steps such as determining the controller structure, determining the parameters to be optimized, establishing the objective function, and integrating CSA with the simulation platform. During this stage, the dynamics equations of the ACC system are implemented in the simulation platform. Figure 4 illustrates the elements related to ACC control, which consists of six main variables: (1) The set velocity (V_{set}), denoting the maximum speed determined by the driver, results in the ego car's speed being set at the specified velocity in m/s^2 when there is no other vehicle in front; (2) The time gap (T_{gap}) represents the preferred safe time interval between the ego car and the front vehicle, measured in seconds; (3) Longitudinal velocity (V_L) indicates the linear speed of the ego car in m/s ; (4) Relative distance (D_R) signifies the gap between the ego

car and the lead vehicle in meters; (5) Relative velocity (V_R) corresponds to the difference in speed between the ego car and the lead car. Finally, (6) Default spacing (D_d) refers to the minimum distance when the ego car is stationary, measured in meters. V_L , D_R , and V_R serve as independent variables in the calculation of the control signal, taking the form of acceleration from the ego car.

To adjust the program based on the distance between the lead car and the ego car, it can be demonstrated in Eq. (4).

$$D_{safe} = D_d + T_{gap}D_{ego} \quad (4)$$

In the classical ACC, there are three key parameters that have a significant impact on the system's outcomes. These three parameters are affecting two types of acceleration equation as follows:

1. The first parameter is known as v_{err} in Eq. (5). It is a variable multiplied by the gap between the desired velocity and the longitudinal velocity, producing the acceleration of the ego car, u_1 (first acceleration equation).

$$u_1 = (V_{set} - V_L) v_{err} \quad (5)$$

The second equation (u_2) comprises two components: x_{err} and V_x . Within this second type, the acceleration of the ego car is determined by multiplying V_x and the relative velocity, subtracted by x_{err} multiplied by the error distance, as formulated in Eqs. (6)-(7).

$$u_2 = V_R V_x - (D_{safe} - D_R) x_{err} \quad (6)$$

$$e_d = D_{safe} - D_R \quad (7)$$

When the speed control mode is active (D_R) value greater than D_{safe}), the applied control signal will be selected from the smaller of the first and second parts, as indicated in Eq. (8). In contrast to speed control, the distance control mode only uses the second part, with the condition that the D_R value is less than D_{safe} , as shown in Eq. (9).

$$a = \min(u_1, u_2) \quad (8)$$

$$a_E = u_2 \quad (9)$$

2.2 Steering control

A steering wheel is a central component in a motorized vehicle that gives the driver the ability to control the direction

of movement of the vehicle. When a vehicle crosses a curved road, the role of the steering wheel becomes very significant to ensuring that the vehicle can navigate the curve safely and in accordance with applicable traffic rules. Meanwhile, ACC technology has successfully integrated automation functions that help vehicles maintain a safe distance from vehicles in front. However, ACC tends to focus more on aspects of controlling speed and distance to the vehicle in front, rather than on the ability to control steering functions, especially when negotiating corners.

The objective of the steering control is to ensure that the vehicle remains within its designated lane and effectively tracks the curved path of the road. This objective is accomplished by controlling the front steering angle. The lateral displacement error (e_1) towards zero shown in Eq. (10), illustrated in Figure 5, and the primary objective is to minimize the yaw angle error (e_2), as shown in Eq. (11). This involves adjusting the steering angle to maintain proper alignment with the desired path and lane position.

$$e_1 = V_x + e_2 + V_y \quad (10)$$

$$e_2 = \psi - \psi_{des} \quad (11)$$

Figure 6 illustrates the use of CSA to overcome problems that arise in efforts to integrate ACC with the steering wheel control function when a vehicle is crossing a curve. CSA, which takes inspiration from the social behavior of crows, is applied to optimize lateral displacement settings when the vehicle is moving through curves, with the aim of increasing efficiency and safety in dealing with such situations. The implementation of CSA in ACC has the potential to expand the role of this technology in controlling the steering wheel, which will ultimately have a positive impact on the level of safety and convenience when negotiating curves on the road.

In addition to CSA, a Proportional-Integral (PI) controller is employed to further enhance steering control. The PI controller has a general equation as written in Eq. (12).

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau \quad (12)$$

where, $u(t)$ is the control input, $e(t)$ is the error signal, k_p is the proportional gain, and k_i is the integral gain. These parameters play a crucial role in optimizing the steering angle. The CSA can be employed to fine-tune k_p and k_i for optimal performance, ensuring the minimization of yaw angle and lateral displacement errors.

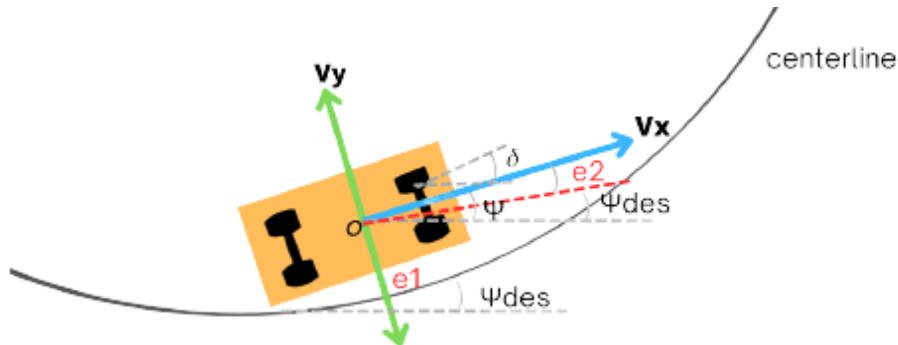


Figure 5. Vehicle dynamic model for steering control

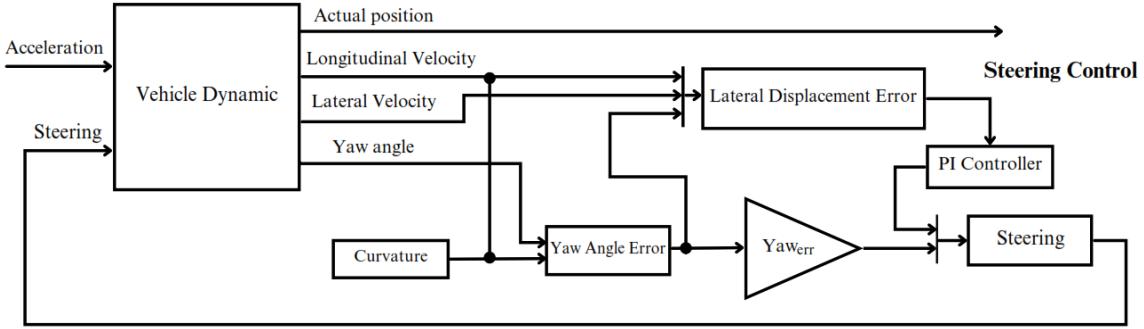


Figure 6. Steering control system diagram

2.3 Crow search algorithm tuning method

CSA describes a group of crows working together in search of food sources. Each bird sets a position that represents a potential solution in the search space [21]. The flight length process begins by calculating the fitness value for each solution based on the optimization objective. Through iteration, facilitated by the interchange of positions and the adoption of superior solutions, the algorithm encapsulates the essence of social dynamics within the crow population. The birds interact by switching between positions and updating a better solution. This concept reflects social behavior in the crow population.

Algorithm 1. Original Crow Search Algorithm

Input: n Flock (population) size for the first iteration
 Ap Awareness probability
 fl Flight length
 $iter$ number of iterations
 $iter_{max}$ Maximum number of iterations
Output: Optimal crow position

- 1 Initialize the position of n crows in the search space
- 2 Evaluate the position of the crows
- 3 $mem = x \rightarrow$ memory initialization
- 4 $fit_mem = ft \rightarrow$ fitness memory
- 5 **while** $iter < iter_{max}$ **do**
- 6 **for** $i=1 : n$
- 7 Choose a random crow to follow
- 8 Define value of awareness probability
- 9 **if** $r_i \geq Ap^{i, iter}$
- 10 $xn^{i, iter+1} = x^{i, iter} + r_i \times fl^{i, iter} \times (m^{i, iter} + x^{i, iter})$
- 11 **else**
- 12 $xn^{i, iter+1} =$ a random position of search space
- 13 **end if**
- 14 **end for**
- 15 Check the feasibility of new positions
- 16 Determine the fitness of crows
- 17 Memory updates for each crow
- 18 $mem = xn \rightarrow$ memory initialization
- 19 $fit_mem = ft \rightarrow$ fitness memory
- 20 **end while**

CSA algorithm is a flock-based optimization algorithm inspired by the behavior of a flock of crows in searching for food [22]. In the context of controller parameter optimization, as explained in this paper, CSA used as a methodology for tuning crucial parameters such as v_{err} , x_{err} , and V_x parameters for straight roads as well as k_p and k_i for steering on ACC controller. The values for those CSA parameters are presented in Table 2.

After adjusting the parameters, determining the parameters in CSA, such as flock size, flight length or fl , iteration, and awareness probability or AP , becomes an important step. The value of fl affects the scope of the solution, where a larger value broadens the scope of the solution in general, while a smaller value narrows the scope of the solution to a local region. Determination of a smaller fl value is needed to obtain the optimal solution. A smaller AP determination drives more new positions on the crows, ultimately providing new solutions to increase the percentage of better solutions. The algorithm runs by making as many individuals as the population, then calculates using Integral Absolute Error (IAE). IAE measures the error between the setpoint (target) and the system output over the entire observation time period by calculating the total of the absolute value of the difference between the setpoint and the system output.

$$IAE = 10^5 / \int_0^\infty |e(t)| dt \quad (13)$$

Table 2. Parameter for CSA

Parameter	Value
Flock size	20
Iteration	100
v_{err}	0-1
x_{err}	0-1
V_x	0-1
k_p	0-1
k_i	0-1

2.4 Archived CSA algorithm

The pseudo-code of ACSA is presented to succinctly summarize the conceptual steps undertaken in the algorithm.

Algorithm 2. Archived Crow Search Algorithm

Input: n Flock (population) size for the first iteration
 Ap Awareness probability
 fl Flight length
 $iter$ number of iterations
 $iter_{max}$ Maximum number of iterations
Output: Optimal crow position

- 1 Initialize the position of n crows in the search space
- 2 Evaluate the position of the crows
- 3 $mem = x \rightarrow$ memory initialization
- 4 $fit_mem = ft \rightarrow$ fitness memory
- 5 **while** $iter < iter_{max}$ **do**
- 6 **for** $i=1 : n$
- 7 Choose a random crow to follow
- 8 Define value of awareness probability
- 9 **if** $r_i \geq Ap^{i, iter}$

```

10       $xn^{i,iter+1} = x^{i,iter} + r_i \times f^{i,iter} \times (m^{i,iter} + x^{i,iter})$ 
11  else
12       $xn^{i,iter+1}$  = a random position of search space
13  end if
14 end for
15 Check the feasibility of new positions
16 Determine the fitness of crows
17 Memory updates for each crow
18 Memory update for archived flock
19  $mem = xn \rightarrow$  memory new crow
20  $fit\_mem = ft \rightarrow$  fitness memory
21 Sort  $fit\_mem, mem, x$ , and  $ft$  from best to worst
22  $xn, x, ft, mem$ , and  $fit\_mem$  take only the first 4 rows.
23 end while

```

In stochastic traffic optimization, the computational time needed for CSA operators is typically insignificant compared to the time required for performance evaluation using simulation models. Thus, there's a critical need to improve the original CSA by reducing the overall computational time for assessing fitness functions. Recent advancements in computational intelligence have introduced an archive crow search algorithm, which employs a selection process utilizing a small population size alongside a large external archive. This external archive stores the best solutions found so far to effectively approximate the optimal solution in various applications. Leveraging the search history stored in the external archive, the selection process aims to minimize the number of function evaluations necessary for achieving convergence. ACSA has been implemented in our application and plays a pivotal role in finding optimal solutions.

In ACSA after updating the crow positions, the algorithm checks the feasibility of the new positions and determines the fitness values of these crows. Subsequently, the memory is updated for each crow, and information regarding fitness, memory, positions, and fitness values is sorted from best to worst. Only the first four rows of xn, x, ft, mem , and fit_mem are retained. In the optimization phase, the population size is reset to four in the first iteration, ensuring that only the top four individuals continue the search.

Overall, the computational efficiency of ACSA makes it a preferable algorithm for specific applications. The implementation of ACSA in this particular application context plays a central role in discovering optimal solutions within a complex search space.

3. EXPERIMENT RESULT

3.1 Performance of archived crow search algorithm

In this paper, both ACSA and OCSA using parameters k_p , k_i , v_{err} , x_{err} and V_x with the objective function combining ACC and steering control were tested to determine whether it could perform well or not. One of the key indicators was the reduction in IAE from each test. Before optimization, IAE reflected the shortcomings of ACC in maintaining speed and distance between the ego car and the lead car. This indicates that the designed CSA is capable of functioning effectively in finding optimal control parameters. Figure 7 illustrates the decrease in IAE values obtained from each test, both from ACSA and OCSA.

In Figure 7, display graphical representations are observed between the ACSA and OCSA methods, indicating variations

in their performance trajectories. The ACSA method exhibits a notable reduction in Integral of Absolute Error (IAE) values, reaching its lowest point in the 39th iteration. This achievement is underscored by two significant drops in IAE during the 5th and 39th iterations. Conversely, the OCSA method showcases a more fluctuating trend, characterized by multiple decreases in IAE values. It requires a longer series of iterations before attaining its optimal IAE value, which occurs in the 47th iteration compared to the optimal trial outcomes. However, it's too early to conclusively assert that ACSA consistently outperforms OCSA in swiftly reaching the optimal point. To validate the consistent performance of CSA, it's imperative to extend beyond single-test conclusions. Therefore, ensuring a robust presentation of CSA's optimal performance necessitates a thorough examination involving the repetition of tests at least 20 times. This iterative testing approach provides a more comprehensive and statistically significant evaluation of the algorithm's capabilities. The result is that after 20 tests OCSA has an average computational time of 103.35 minutes, while ACSA has an average computational time of 17.5. The confidence interval for the computational time ACSA has (16.03,18.17) and for OCSA has (141.48,144.22). In comparison, the efficiency is around 87.41-88.64%.

As illustrated in Figure 8, This repetitive testing approach provides a more comprehensive and statistically significant evaluation of the algorithm's capabilities. The result is that after 20 tests OCSA has an average computational time of 103.35 minutes, while ACSA has an average computational time of 17.5. The confidence interval for the computational time ACSA has (16.03,18.17) and for OCSA has (141.48,144.22). In comparison, the efficiency is around 87.41-88.64%.

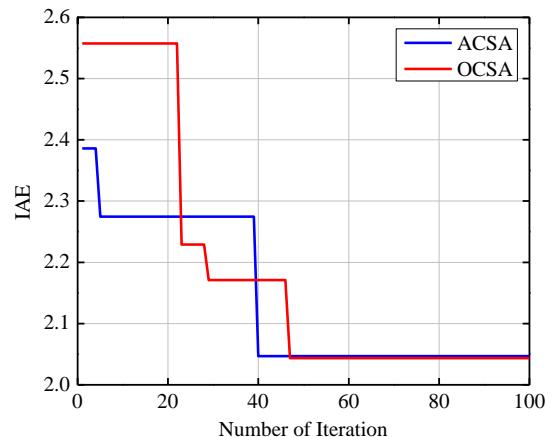


Figure 7. Performance comparison between ACSA and OCSA

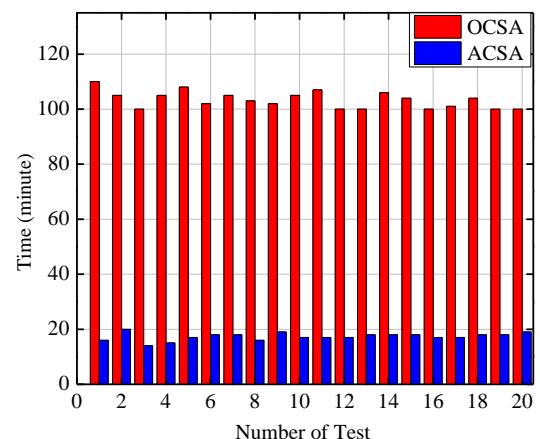


Figure 8. Computational time comparison

Table 3. IAE change after optimization

	Integral Absolute Error		
	Steering	ACC	Addition
Before optimization	0.3591	18.5967	18.9558
Optimization using ACSA	0.0562	1.9952	2.0514
Optimization using OCSA	0.0327	1.9674	2.0002
			2.0830

Analyzing the results depicted in Table 3, it becomes evident that the IAE values derived from optimization in the best generation for the Optimal OCSA are slightly better than those for the ACSA. Nevertheless, it is imperative to underscore the remarkable consistency observed in the mean IAE across both ACC and steering control for two controller, which 2.0830 for OCSA and 2.1464 for ACSA minute with an average result of IAE is only different 0.0634. This stability in the average IAE across diverse control aspects strongly indicates that CSA maintains a dependable performance level consistently. Also, confidence interval for the performance seen from IAE OCSA has (2.0708, 2.1072) and ACSA has (2.0765, 2.1315). In performance comparison between before optimization with OCSA has improvement around 88.91-89.12% and for ACSA has 88.71-89.07%

Table 4. Optimized parameter after tuning 20 time with CSA

Criteria	Optimized Parameter	Optimal Solution	Standard Deviation
ACSA	K_p	0.7492	0.2382
	K_i	0.6506	0.2784
	V_{err}	0.9716	0.2698
	X_{err}	0.9778	0.2903
	V_x	0.7012	0.0572
OCSA	K_p	0.9547	0.2244
	K_i	0.8985	0.2107
	V_{err}	0.3015	0.1364
	X_{err}	0.9819	0.0147
	V_x	0.7088	0.0130

The standard deviation plays a crucial role in providing insights into the dispersion or clustering of optimized parameters around the optimal solution. Examining the values presented in Table 4, it becomes apparent that the standard deviation associated with both the ACC and steering control is consistently observed to be relatively small and often below zero. This finding implies that the optimized parameters, critical for achieving efficient performance in both the ACC and steering control systems, exhibit a tendency to be stable and converge toward the best possible solution. The smaller standard deviation values mean reduced variability and a more reliable optimization process, indicating a higher level of precision and confidence in the model's performance. This stability in the optimized parameters is paramount for ensuring the effectiveness and reliability of the control mechanisms, contributing to a more robust and dependable overall system.

3.2 Performance of ACC and steering control

Figure 9 shows promising results for both CSA. This carefully designed system showcases a level of optimization that has a profound impact on both ACC and steering control. Its efficacy becomes evident in its ability to judiciously manage the relative distance (D_R) between the ego car and the

lead car, always diligently approaching the designated safe distance (D_{safe}) without ever subsiding it. CSA's profound contributions extend to the ACC system, which gains the capability to not only accurately measure the speed and position of the vehicle in front but also provide road users with an elevated level of safety and comfort during their journeys.

To show that the test was carried out on a curved road scenario, Figure 10 shows the parameters used by CSA as an optimization objective. The graph shows a very small lateral displacement error, especially below 0.002, reflecting the high level of accuracy and stability of the car system in maintaining its lateral position. Lateral displacement error measures how far the car is from the desired position on the lateral axis, which is parallel to the direction of the car's movement.

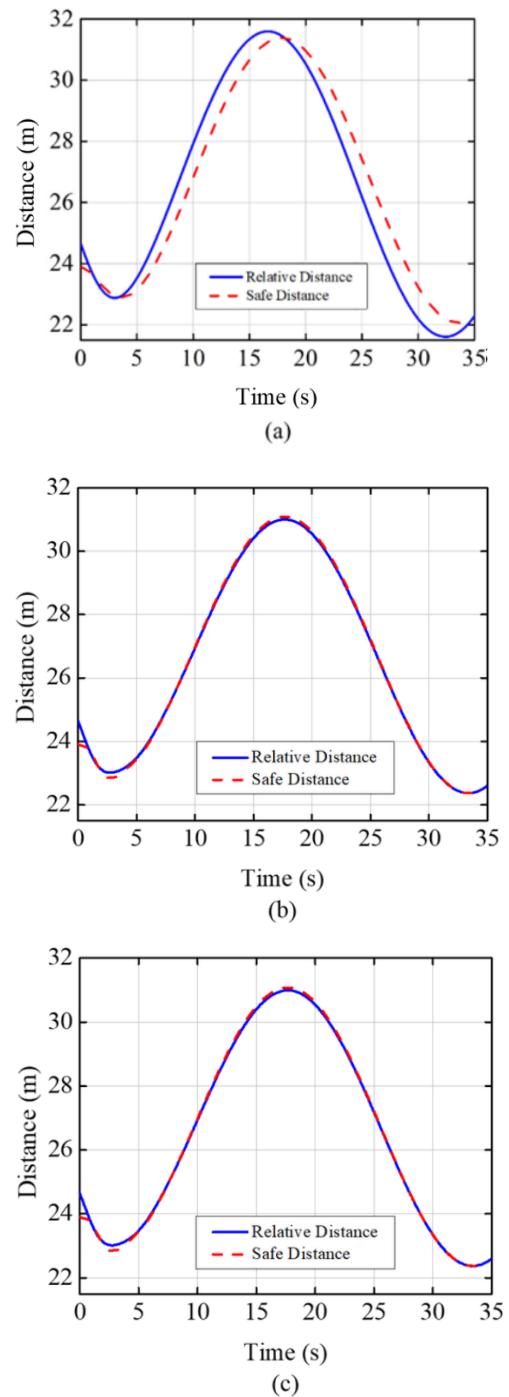


Figure 9. Performance (a) before optimization, (b) using OCSA and (c) using ACSA

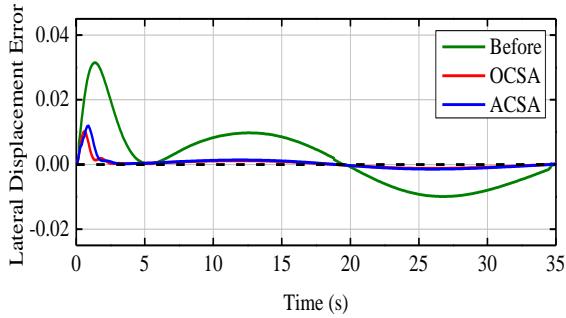
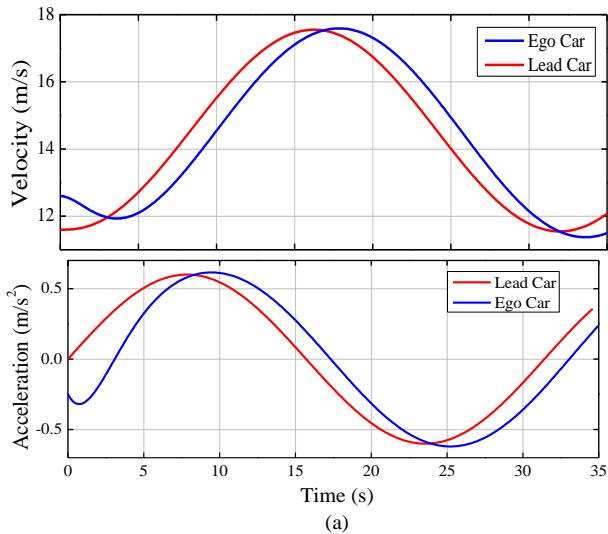
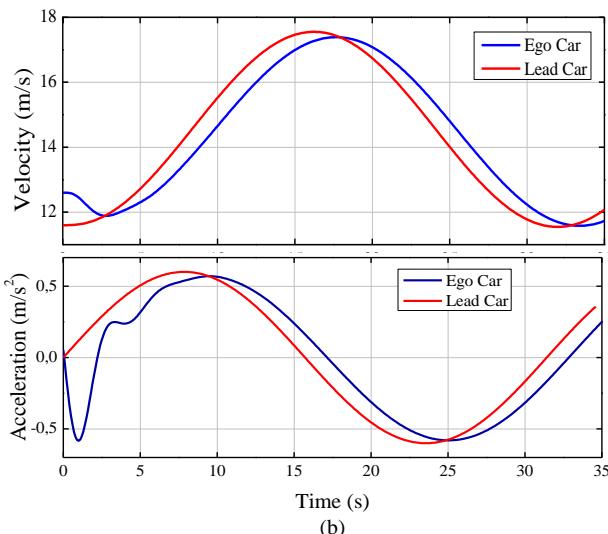


Figure 10. Lateral displacement error before and after optimization using OCSA and ACSA

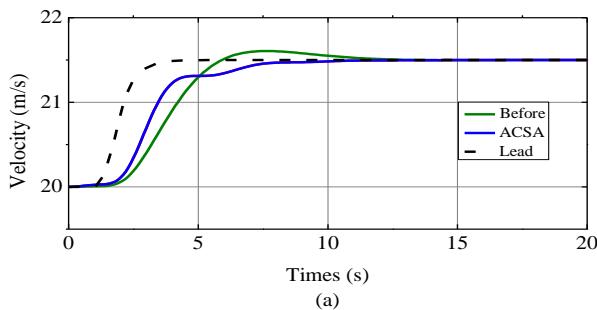


(a)

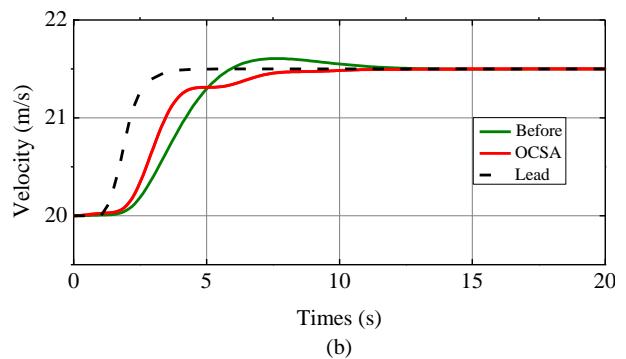


(b)

Figure 11. Velocity and acceleration (a) before and (b) after optimization using ACSA



(a)



(b)

Figure 12. Response to disturbance before and after optimization using (a) OCSA and (b) ACSA

To show that the test was carried out on a curved road scenario, Figure 10 shows the parameters used by CSA as an optimization objective. The graph shows a very small lateral displacement error, especially below 0.002, reflecting the high level of accuracy and stability of the car system in maintaining its lateral position. Lateral displacement error measures how far the car is from the desired position on the lateral axis, which is parallel to the direction of the car's movement.

The implementation of this algorithm in ACC is a resounding success, one that speaks volumes about its potential to significantly enhance traffic safety and vehicle efficiency. The algorithm empowers ACC to expertly oversee the maximum acceleration within a range spanning from -3 m/s^2 to 2 m/s^2 [23]. This control ensures that not only are road users safe from abrupt and uncomfortable speed changes, but the overall efficiency and functionality of vehicles are maximized. In a world increasingly driven by the pursuit of safer and more efficient transportation solutions. Figure 11 shows that CSA can be used for optimization and in the future is not only for simulation but can be used for real hardware implementation.

The initial speed of the ego and lead car is same, but suddenly the lead car has an increased velocity. Figure 12 shows how the controller on the ego car responds to changes in the speed of the lead vehicle. The ego car will be increasing speed to match the speed to lead car. The speed response generated before optimized has overshoot 0.49%, settling time 13 second and overshoot 6 seconds. As for OCSA has 9 seconds settling time and ACSA has 9.1 seconds settling time. It shows that there is no significant difference in performance between OCSA and ACSA.

4. CONCLUSION

This article describes quite promising results in the implementation of adaptive cruise control in curving road conditions using steering control, along with the optimization of its parameters through the use of the crow search algorithm as the optimization method. The results of this research indicate that the OCSA and ACSA have successfully improved the performance of ACC in accurately controlling distance and speed, thus providing a more comfortable and safe driving experience. Performance of both methods is similar. Moreover, ACC can now maintain a more stable distance in curved roads due to its integration with steering control, both of which have been optimized simultaneously using CSA.

However, significant challenges exist in translating simulation results into real-world applications. The implementation of ACC requires careful consideration of vehicle hardware requirements and calibration processes to ensure optimal performance and safety in various driving situations. Future research should focus more on overcoming these challenges to facilitate the deployment of optimized ACC in real-world driving scenarios.

ACKNOWLEDGMENT

We wish to acknowledge the Ministry of Education, Culture, Research, and Technology, Indonesia, for supporting our research under a research grant with contract number: 144/E5/PG.02.00.PL/2023. We would also like to express our sincere gratitude to our discussion partners, Akhyar Abdillah Manaf and Ilya Amelia, whose valuable and thoughtful discussions lead to the development and refinement of our experimental works throughout this project. We are truly fortunate to have had such a kind and knowledgeable partner.

REFERENCES

- [1] World Health Organization. (2023). Road safety. https://www.who.int/health-topics/road-safety#tab=tab_1, accessed on Sep. 8, 2023.
- [2] Mohammed, A.A., Ambak, K., Mosa, A.M., Syamsunur, D. (2019). A review of the traffic accidents and related practices worldwide. *The Open Transportation Journal*, 13(1): 65-83. <https://doi.org/10.2174/1874447801913010065>
- [3] Li, X.P. (2022). Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method. *Transportation Research Part B: Methodological*, 166: 1-18. <https://doi.org/10.1016/j.trb.2022.09.003>
- [4] Jiménez, F., Naranjo, J.E., Anaya, J.J., García, F., Ponz, A., Armingol, J.M. (2016). Advanced driver assistance system for road environments to improve safety and efficiency. *Transportation Research Procedia*, 14: 2245-2254. <https://doi.org/10.1016/j.trpro.2016.05.240>
- [5] Kukkala, V.K., Tunnell, J., Pasricha, S., Bradley, T. (2018). Advanced driver-assistance systems: A path toward autonomous vehicles. *IEEE Consumer Electronics Magazine*, 7(5): 18-25. <https://doi.org/10.1109/mce.2018.2828440>
- [6] Vahidi, A., Eskandarian, A. (2003). Research advances in intelligent collision avoidance and adaptive cruise control. *IEEE Intelligent Transportation Systems Magazine*, 4(3): 143-153. <https://doi.org/10.1109/its.2003.821292>
- [7] Chen, Y.H., Feng, G.D., Wu, S.F., Tan, X.J. (2021). A new hybrid model predictive controller design for adaptive cruise of autonomous electric vehicles. *Journal of Advanced Transportation*, 2021: 6626243. <https://doi.org/10.1155/2021/6626243>
- [8] Kim, S.G., Tomizuka, M., Cheng, K.H. (2012). Smooth motion control of the adaptive cruise control system by a virtual lead vehicle. *International Journal of Automotive Technology*, 13: 77-85. <https://doi.org/10.1007/s12239-012-0007-6>
- [9] Nuri, N.R.M., Hazli, H.H., Tofrowaih, K.A. (2022). Adaptive cruise control using PID and model predictive controllers in low traffic condition. In Singh, A.S.P., Abdollah, M.F.B., Amiruddin, H., Taha, M.M. (eds.) *Proceedings of Mechanical Engineering Research Day 2022*, pp. 152-153. UTeM Press.
- [10] Singh, A., Satsangi, C.S., Panse, P. (2015). Adaptive cruise control using fuzzy logic. *International Journal of Digital Application & Contemporary Research*, 3(8): 1-7.
- [11] Alomari, K., Mendoza, R.C., Sundermann, S., Goehring, D., Rojas, R. (2021). Fuzzy logic-based adaptive cruise control for autonomous model car. In: Galambos, P., Madani, K. (eds) *Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems ROBOVIS*, 1: 121-130. SciTePress. <https://doi.org/10.5220/0010175101210130>
- [12] Prastiyanto, D., Apriaskar, E., Ibrahim, R.F., Rumanda, A., Manaf, A.A., Amelia, I. (2023). Adaptive cruise control tuned by genetic algorithm for safe distance of automated vehicle. *IOP Conference Series: Earth and Environmental Science*, 1203(1): 012026. <https://doi.org/10.1088/1755-1315/1203/1/012026>
- [13] Ozkaya, E., Arslan, H., Sen, O.T. (2021). Particle swarm optimization method based controller tuning for adaptive cruise control application. *Gazi University Journal of Science*, 34(2): 517-527. <https://doi.org/10.35378/gujs.762103>
- [14] Idriz, A.F., Rachman, A.S.A., Baldi, S. (2017). Integration of auto-steering with adaptive cruise control for improved cornering behaviour. *IET Intelligent Transport Systems*, 11(10): 667-675. <https://doi.org/10.1049/iet-its.2017.0089>
- [15] Sharp, R.S., Casanova, D., Symonds, P. (2000). A mathematical model for driver steering control, with design, tuning and performance results. *Vehicle System Dynamics*, 33(5): 289-326. [https://doi.org/10.1076/0042-3114\(200005\)33:5;1-Q;FT289](https://doi.org/10.1076/0042-3114(200005)33:5;1-Q;FT289)
- [16] Dahmani, H., Chadli, M., Rabhi, A., El Hajjaji, A. (2015). Vehicle dynamics and road curvature estimation for lane departure warning system using robust fuzzy observers: Experimental validation. *Vehicle System Dynamics*, 53(8): 1135-1149. <https://doi.org/10.1080/00423114.2015.1026609>
- [17] Zhang, D., Xiao, Q., Wang, J., Li, K. (2013). Driver curve speed model and its application to ACC speed control in curved roads. *International Journal of Automotive Technology*, 14: 241-247. <https://doi.org/10.1007/s12239-013-0027-x>
- [18] Wang, H.F., Wang, Y.F., Zhao, X.M., Wang, G.P., Huang, H., Zhang, J.J. (2019). Lane detection of curving road for structural highway with straight-curve model on vision. *IEEE Transactions on Vehicular Technology*, 68(6): 5321-5330. <https://doi.org/10.1109/TVT.2019.2913187>
- [19] He, Y.L., Ciuffo, B., Zhou, Q., Makridis, M., Mattas, K., Li, J., Li, Z.Y., Yan, F.W., Xu, H.M. (2019). Adaptive cruise control strategies implemented on experimental vehicles: A review. *IFAC-PapersOnLine*, 52(5): 21-27. <https://doi.org/10.1016/J.IFACOL.2019.09.004>
- [20] Arifin, B., Suprapto, B.Y., Prasetyowati, S.A.D., Nawawi, Z. (2022). Steering control in electric power steering autonomous vehicle using type-2 fuzzy logic

- control and PI control. World Electric Vehicle Journal, 13(3): 53. <https://doi.org/10.3390/wevj13030053>
- [21] Alkrwy, A., Hussein, A.A., Atyia, T.H., Khamees, M. (2021). Adaptive tuning of PID controller using crow search algorithm for DC motor. IOP Conference Series: Materials Science and Engineering, 1076(1): 012001. <https://doi.org/10.1088/1757-899X/1076/1/012001>
- [22] Hussien, A.G., Amin, M., Wang, M.J., Liang, G.X., Alsanad, A., Gumaei, A., Chen, H.L. (2020). Crow search algorithm: Theory, recent advances, and applications. IEEE Access, 8: 173548-173565. <https://doi.org/10.1109/ACCESS.2020.3024108>
- [23] Karjanto, J., Wils, H., Yusof, N.M., Terken, J., Delbressine, F., Rauterberg, M. (2022). Measuring the perception of comfort in acceleration variation using electrocardiogram and self-rating measurement for the passengers of the automated vehicle. Journal of Engineering Science and Technology, 17(1): 180-196.