

Article

Performance Evaluation of PID Controller for an Automobile Cruise Control System using Ant Lion Optimizer

Rosy Pradhan^{1,a}, Santosh Kumar Majhi^{2,b,*}, Jatin Ku Pradhan^{1,c}, and Bibhuti Bhusan Pati^{1,d}

¹ Department of Electrical Engineering, Veer Surendra Sai University of Technology, Burla, India

² Department of Computer Science & Engineering, Veer Surendra Sai University of Technology, Burla, India

E-mail: ^arosy.pradhan88@gmail.com, ^bsmajhi_cse@vssut.ac.in (Corresponding author), ^cjkp10@iitbbs.ac.in, ^dpati_bibhuti@rediffmail.com

Abstract. This paper considers the design and performance evaluation of PID controller for an automobile cruise control system (ACCS). A linearized model of the cruise control system has been studied as per the dominant characteristics in closed loop system. The design problem is recast into an optimization problem which is solved using Ant Lion Optimization (ALO). The transient performance of proposed ACCS i.e., settling time, rise time, maximum overshoot, peak time and steady state error are investigated by step input response and root locus analysis. To show the efficacy of the proposed algorithm over a state space method, classical PID, fuzzy logic, genetic algorithm, a comparison study is presented by using MATLAB/SIMULINK. Furthermore, the robustness of the system is evaluated by using bode analysis, sensitivity, complimentary sensitivity and controller sensitivity. The results indicate that the designed ALO based PID controller for ACCS achieves better performance than other recent methods reported in the literature.

Keywords: PID controller, ant lion optimizer, cruise control system, time domain analysis, frequency domain analysis, robustness analysis.

ENGINEERING JOURNAL Volume 21 Issue 5

Received 22 February 2017

Accepted 22 April 2017

Published 29 September 2017

Online at <http://www.engj.org/>

DOI:10.4186/ej.2017.21.5.347

1. Introduction

Automobile cruise control system (ACCS) is designed to reduce the drivers fatigue in the long run drive. In addition, now-a-days traffic safety is given a priority and primary concern in the automotive research area. In that perspective, cruise control permits the driver to manage the vehicle speed and when ACCS is turn on the speed of the automobile is upheld automatically without the application of the accelerator pedal. In turn the probability of potential crash is minimized and the safety is maintained. The throttle is attuned as per the calculated velocity of the automobile. The reaction time of the ACCS is considered very critical due to the velocity variation of the system. Currently, a number of features are added due to the revolution of recent technologies e.g., all the control operations such as the speed control, the attainment of last run speed, deactivate the speed after brakes etc. by pressing the buttons [1]. Moreover these features are common in automotive vehicles. The automobile vendors are moving towards designing of automatic vehicles. ACCS is used to provide safety, traffic fluency and also reduces fuel consumption [2, 3]. Moreover, cruise control system helps in avoiding collisions between vehicles, better traffic management, reducing travel time and lower consumption of fuel by maintaining desired speed [4, 5].

Various control methods applied to cruise control system is found in literature [6-16] including conventional PID, state space, fuzzy logic, genetic algorithm (GA). In these control methods the objective is to find the values of proportional (K_p), integral (K_i) and derivatives (K_d) by optimizing the unit feedback function.

This work is focused to optimize the PID controller parameters to obtain the best result for the cruise control system. Additionally, an unstable cruise control system is made stable by using the concept of Taylor series expansion. Owing to many advantage offered, the ALO is used to optimize the regulatory parameter as per the specification of the system. The ALO tuned PID controller, the first of its use for ACCS performs comparatively better than other reported methods in literature [17-19].

The rest of the paper is organized as follows. Section 2 presents the description of the non linear cruise control model and its linearized version. Section 3 presents the brief overview of the ant lion optimizer optimization to make it self-content. In Section 4, result and discussion are presented considering application of ALO algorithm, effect of objective function, transient analysis, robustness analysis (bode analysis, complimentary sensitivity and controller sensitivity) and root locus analysis. Finally, this paper concludes briefly in Section 5.

2. Description of Automobile Cruise Control System (ACCS)

The ACCS is used to regulate the vehicle's speed according to the driver's reference command. The schematic block diagram of cruise control system considered in this paper is shown in Fig. 1. Here the cruise control system generates the desired amount of throttle input u so as to maintain the constant velocity (V) i.e. to follow the reference velocity V_{ref} . In order to maintain the constant velocity of the vehicle, the pedal actuator generates the desired amount of gas pedal depression (δ) when the road inclination angle (θ) increases. The system dynamics is presented in Fig. 2. The nonlinear longitudinal dynamics of the vehicle may be written as [11].

$$F_d = M \frac{dv}{dt} + F_a + F_g \quad (1)$$

where M is mass of vehicle and passenger, V is velocity of the vehicle, $F_a = C_a(V - V_w)^2$ is an aerodynamic drag, C_a is aerodynamic drag co-efficient, V_w is wind gust speed, $F_g = Mg \sin \theta$ is climbing resistance or downgrade force, θ is road inclination, F_d is engine drive force and g is the gravitational acceleration.

Furthermore, the actuator of cruise control system is modeled as first order lag system i.e. $\frac{C_1 e^{\tau s}}{Ts+1}$ with saturation block having saturation limit F_{dmin} and F_{dmax} , where C_1 is the actuator constant, T is the time of observation and τ is the reaction time of the driver. The engine drive force can be written as

$$F_d = \frac{C_1 e^{\tau s}}{Ts+1} \quad (2)$$

Linearized model of cruise control system:

For making the design of controller easy, one must have linearized model for the vehicle system to start with. By considering all initial conditions and disturbances i.e. $V_w = 0$ and $\theta = 0$, the nonlinear Eq. (1) and (2) can be converted to

$$\dot{V} = \frac{1}{M}(F_d - C_a V^2) \quad (3)$$

$$\dot{F}_d = \frac{1}{T}(C_1 u(t - T) - F_d) \quad (4)$$

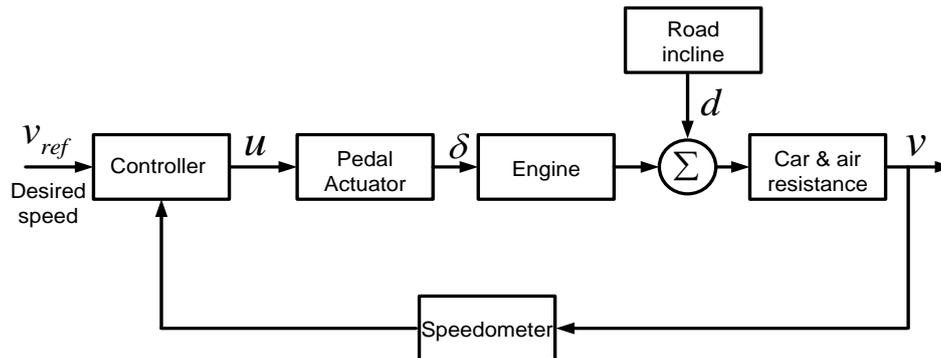


Fig. 1. Block diagram of cruise control system [11].

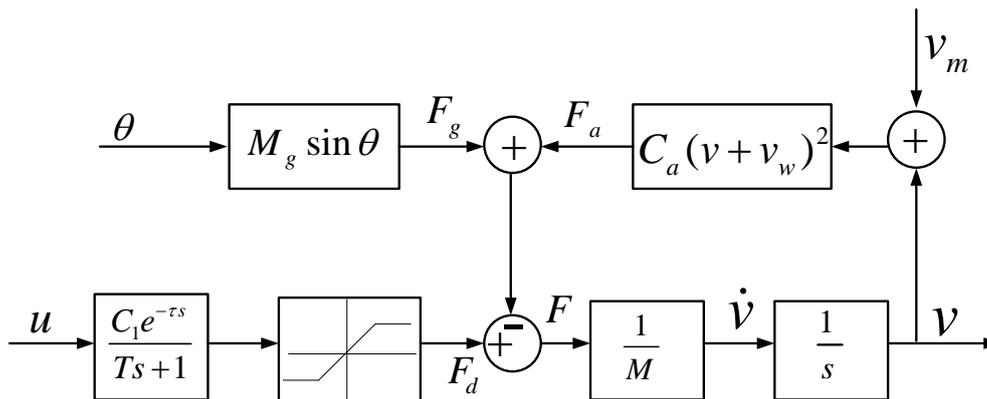


Fig. 2. Dynamic model of automobile [11].

However, from the above two equations, it is observed that the non-linearity still exists due to the quadratic term in Eq. (3). The non-linearity is eliminated by using Taylor's series expansion. Let the non-linear Eq. (3) and (4) be represented as

$$\dot{x} = f(x, u), \quad y = h(x, u) \quad (5)$$

where x is the state vector defined as $x = [V \ F_d]^T$, u is the control input vector, y is the output vector which is equal to the velocity. Let $x = x^0 + \Delta x$, $u = u^0 + \Delta u$, where (x^0, u^0) is the equilibrium or operating point and Δx and Δu is the small deviation around the operating point. The linearized system then becomes

$$\Delta \dot{x} = A \Delta x + B \Delta u(t - T) \quad (6)$$

$$y = C \Delta x \quad (7)$$

where

$$A = \left. \frac{\partial f}{\partial x} \right|_{x^0, u^0} = \begin{bmatrix} -\frac{2C_a V^0}{M} & \frac{1}{M} \\ 0 & -\frac{1}{T} \end{bmatrix}, \quad B = \left. \frac{\partial f}{\partial u} \right|_{x^0, u^0} = \begin{bmatrix} 0 \\ \frac{C_1}{T} \end{bmatrix} \quad \text{and} \quad C = \left. \frac{\partial h}{\partial x} \right|_{x^0, u^0} = [1 \ 0].$$

The transfer function $G(s)$ can be directly obtained from the above state space matrices (6) and (7) as

$$G(s) = \frac{\Delta V(s)}{\Delta U(s)} = \frac{\frac{c_1 e^{-\tau s}}{MT}}{\left(s + \frac{2C_a v}{M}\right)\left(s + \frac{1}{T}\right)} \quad (8)$$

Now, with the operating point $v = 30$ km/hr and from Table 1, the value of system matrices and the plant transfer function $G(s)$ are

$$A = \begin{bmatrix} -0.0476 & 0.00067 \\ 0 & -1 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 743 \end{bmatrix}; C = [1 \ 0].$$

$$G(s) = \frac{\Delta V(s)}{\Delta U(s)} = \frac{2.4767}{(s+0.0476)(s+1)(s+5)} \quad (9)$$

Table 1. Model parameters considered [11].

| Symbol | Value | Units |
|------------|-------|------------------------|
| C_1 | 743 | - |
| C_a | 1.19 | N/(m/sec) ² |
| M | 1500 | Kg |
| τ | 0.2 | sec |
| T | 1 | Sec |
| F_{dmax} | 3500 | N |
| F_{dmin} | -3500 | N |
| g | 9.8 | m/sec ² |

3. Overview of Ant Lion Optimizer

Ant Lion Optimization technique is the one of the nature inspired optimization algorithm for solving the uni-dimensional as well as multidimensional optimization problem. This algorithm was proposed by Mirjalili (2015) [20]. This algorithm is encouraged by the hunting activities of the grey antlions in nature and basically their favorite preys are ants shown in Fig. 3. A cone shaped pit is created by the antlions and they hides its larvae under beneath the bottom of cone shaped pit to trap the ant. The edge of cone is made sharp such that the ant can easily falls into the bottom. Once the preys trapped into the cone, then the antlion throws the sand towards the edge of the cone which makes the prey incapable to escape from the trap. After that the antlion consumes the prey and prepares another pit to trap next prey. The above described hunting nature of antlion is explained in five stages such as the random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps. As ALO mimics the hunting activity of antlion, so this optimization governed by the following certain conditions [20].

1. Ants move in random walks about the search space.
2. The traps of the antlions are influenced by random walks.
3. Antlions can construct their pit according to their fitness value. The fitness value determines the pit size.
4. Antlion with higher fitness value i.e larger pit have maximum chance to catch the ants.
5. In each iteration, an ant is caught by a fittest antlion.
6. In order to trigger the sliding behavior of ants towards antlions, the assortment of random walk of ant is adjusted in decreasing order.
7. If the fitness value of ant is more than that of an antlion, which means that it is captured by the antlion.
8. After each hunt step, the position is changed by antlion and they build a new improved pit for trapping another prey.

In addition, the antlion optimization has four matrices which store the position and fitness of ant and ant lions. The position of ants is saved in a matrix during the optimization process. The position of ants is shown in the position matrix (M_{ant}), where n is the number of ants and d is the number of variables or dimensions. In addition, by considering the position matrix, the fitness (objective) function of each ant is evaluated and stored in a fitness matrix named as M_{FA} , where n represents the populations of ants, f is the fitness or objective function and d denotes the number of variables.

$$M_{ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,d} \end{bmatrix}$$

$$M_{FA} = \begin{bmatrix} f(A_{1,1} & A_{1,2} & \dots & A_{1,d}) \\ f(A_{2,1} & A_{2,2} & \dots & A_{2,d}) \\ \vdots & \vdots & \ddots & \vdots \\ f(A_{n,1} & A_{n,2} & \dots & A_{n,d}) \end{bmatrix}$$

Moreover, the antlions are hiding somewhere in the search space, so the position and the fitness value are also stored in the matrices $M_{antlion}$ and M_{FAL} respectively. Here n refers to the number of antlions and d refers to the number of variables.

$$M_{antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \dots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \dots & AL_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \dots & AL_{n,d} \end{bmatrix}$$

$$M_{FAL} = \begin{bmatrix} f(AL_{1,1} & AL_{1,2} & \dots & AL_{1,d}) \\ f(AL_{2,1} & AL_{2,2} & \dots & AL_{2,d}) \\ \vdots & \vdots & \ddots & \vdots \\ f(AL_{n,1} & AL_{n,2} & \dots & AL_{n,d}) \end{bmatrix}$$

The five major steps for the optimization method and their mathematical descriptions are mentioned below [20].

1. Random walks of ants

For modelling purpose of the hunting behavior of the antlion, the antlion and ant should have interaction with each other. For this the ants are required to move in the search space for food and shelter and antlions are hunted the ants by using their traps. Because of the stochastic movement of ants for searching of foods, a random walk is chosen for modeling of ant movements as follows,

$$X(t) = [0, \text{cusum}(2r(t_1) - 1), \dots, \text{cusum}(2r(t_n) - 1)] \quad (10)$$

where cusum calculates the cumulative sum, n indicates the maximum numbers of iteration; t is the steps of random walk and $r(t)$ is the stochastic function defined as below:

$$r(t) = \begin{cases} 1 & \text{if } \text{rand} > 0.5 \\ 0 & \text{if } \text{rand} \leq 0.5 \end{cases} \quad (11)$$

where rand is a random number generated with uniform distribution in the interval 0 to 1.

The mathematical representation of normalized random walk of ant is given by following equation [20]:

$$X_i(t) = \frac{(x_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + C_i^t \quad (12)$$

where a_i is the minimum of random walk of i^{th} variable, b_i indicates the maximum of random walk of the i^{th} variable, C_i^t is the minimum of the i^{th} variable at t^{th} iteration and similarly d_i^t is the maximum of the i^{th} variable at t^{th} iteration.

2. Building trap

The mathematical modeling of antlion's hunting capability is influenced by a roulette wheel. The ALO method uses the roulette wheel function to search the fittest antlion during the optimization process. This process filters the best antlion with higher probability for catching the prey.

3. Entrapments of ants in traps

As discussed above, the random walk of ants are influenced by the traps of the antlions. Therefore the mathematical relationship for this assumption is expressed by the following equation

$$C_i^t = \text{Antlion}_j^t + C^t \quad (13)$$

$$d_i^t = \text{Antlion}_j^t + d^t \quad (14)$$

where vector C and d are the hypersphere of randomly walked ant and around the selected antlion respectively. C^t and d^t indicates the minimum and maximum value of all the variable at t^{th} iteration respectively. Similarly C_i^t and d_i^t are minimum and maximum of the i^{th} variable at t^{th} iteration respectively.

4. Sliding ants towards the antlion

Once the antlion realized that the ant is in the trap, it throws the sand towards the edge of the pit through its mouth which makes the ant trap inside the pit. This behavior of antlion is modeled by the following equation

$$C^t = \frac{c^t}{I} \quad (15)$$

$$d^t = \frac{d^t}{I} \quad (16)$$

where I indicates the ratio described in Eq. (13), C^t and d^t are the minimum and maximum values of all the variables at t^{th} iteration respectively.

$$I = 10^w \frac{t}{T} \quad (17)$$

where t is the current iteration, T is maximum number of iteration and w is the constant depending on the current iteration. The above two mathematical equations represent the sliding process of ant into the pit.

5. Prey catching and rebuilding the trap

When the ant is caught by the antlion, it is considered as the final stage of hunt. This behavior of antlion is described by the Eq. (18) where the fitness value of ant is more than the fitness value of antlion. In this situation ant is consumed by the antlion. Then the antlion is updated with its position or build a new trap to catch a new prey.

$$\text{Antlion}_j^t = \text{Ant}_i^t \text{ if } f(\text{Ant}_i^t) > f(\text{Antlion}_j^t) \quad (18)$$

where t is the current iteration, Antlion_j^t indicates the position of the selected j^{th} antlion at t^{th} iteration and Ant_i^t shows the position of i^{th} ant at t^{th} iteration. Function $f()$ indicates the fitness value of ant and antlion.

6. Elitism

Elitism is the significant feature which allows the algorithm to obtain best solution at every step of the optimization procedure. During the ALO algorithm iteration, the preeminent antlion is obtained and saved as an elite. Since the elite one influences the movement of all the ants during the iterations, consequently, it is considered that every ant moved randomly around the selected antlion and elite simultaneously and is given by the given equations

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (19)$$

where R_A^t is the random walk around the selected antlion at t^{th} iteration, R_E^t indicates the random walk around the elite t^{th} iteration and Ant_i^t shows the position of i^{th} ant at t^{th} iteration. The steps for the algorithm are given in Table 2.

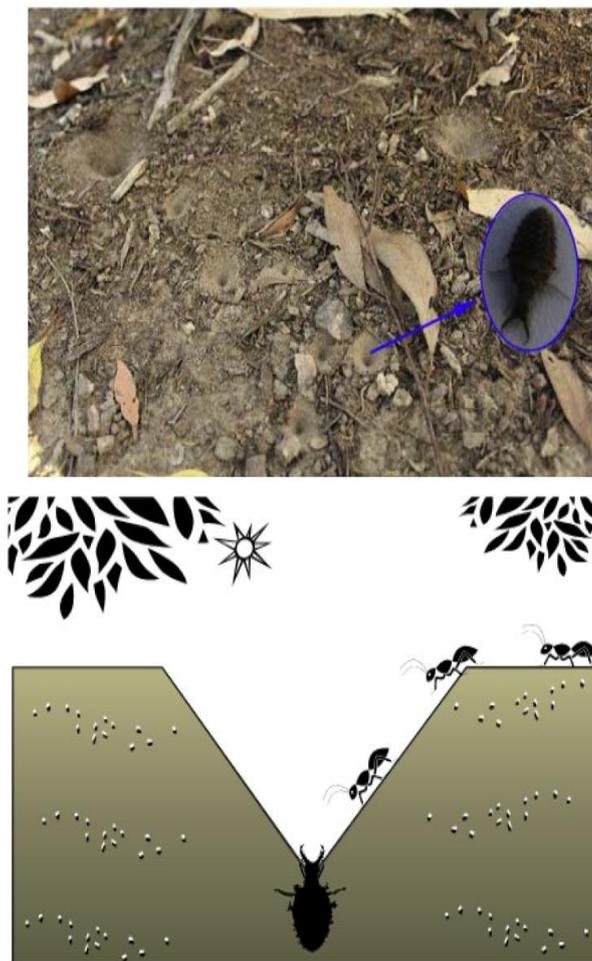


Fig. 3. Hunting nature of antlion [20].

The ALO has shown high performance in solving the classical optimization problem. ALO converges rapidly towards the optimum with the help of exploitation. It has algorithm that has a high intensity of exploration which helps it to explore the capable regions of the search space. Furthermore, the local optima avoidance of this algorithm is satisfactory since it is able to avoid all of the local optima and approach the global optima. The algorithm provides better results in comparison with PSO, DE and other meta-heuristic algorithms [17-19].

Table 2. Main steps of ALO algorithm.

| |
|---|
| Input: number of search agents, maximum iteration Output: Best Score and Best Positions Initialize the ant and antlion population randomly Calculation of fitness of ant and antlion and determination of elite (E) While (t< maximum iteration) For each ant(search agent) Select an antlion based on Roulette wheel Update the minimum value and maximum value of t th iteration Random walk creation and normalization based on Min-Max normalization Update the position of ants End for Calculate the fitness of all search agents Replace an fitter ant lion Update elite if antlion is better than elite t=t+1 End While Return E |
|---|

4. Implementation and Analysis

4.1. Problem Formulation

ALO algorithm tuned PID Controller for an ACCS system is shown in Fig. 4. The self tuning PID is considered as controller for the ACCS system because of its proven advantages. The PID controller parameters are tuned by classical PID, state space, GA, Fuzzy Logic and ALO algorithms. During the evolutionary algorithms, the upper and lower bounds of different parameters are chosen as $3 < K_p < 4$, $0.1 < K_i < 0.25$, $3 < K_d < 4$ [10]. The number of iterations and population size for ALO algorithm are 1000 and 100 respectively. For all objective functions, the numbers of iteration and population size remain same. The objective functions for these algorithms are the various time domain integral performance indices which are represented by Eq. (20)-(24). Optimum values of the controller can be calculated by minimizing the indices functions. The objective function is chosen for minimizing the time response characteristics due to the dependency of error on time.

- (a) Integral Absolute Error (IAE)

$$J_1 = \text{IAE} = \int_0^{\infty} |e(t)| dt \quad (20)$$

- (b) Integral Square Error (ISE)

$$J_2 = \text{ISE} = \int_0^{\infty} |e^2(t)| dt \quad (21)$$

- (c) Integral of Time multiplied Absolute Error (ITAE)

$$J_3 = \text{ITAE} = \int_0^{\infty} t |e(t)| dt \quad (22)$$

- (d) Integral of time multiplied Square Error (ITSE)

$$J_4 = \text{ITSE} = \int_0^{\infty} t |e^2(t)| dt \quad (23)$$

- (e) Integral Square Time Multiplied Square Error (ISTSE)

$$J_5 = \text{ISTSE} = \int_0^{\infty} t^2 |e^2(t)| dt \quad (24)$$

The problem can be represented as

$$\text{Minimize } J \quad (25)$$

Subjected to

$$\begin{aligned} K_{pmin} < K_p < K_{pmax} \\ K_{imin} < K_i < K_{imax} \\ K_{dmin} < K_d < K_{dmax} \end{aligned}$$

Here, J is the objective function (J_1, J_2, J_3, J_4 and J_5) and $e(t)$ is the error $\Delta V_{ref} - \Delta V$

From the various performance indices, the optimal gains of the controller obtained are presented next section.

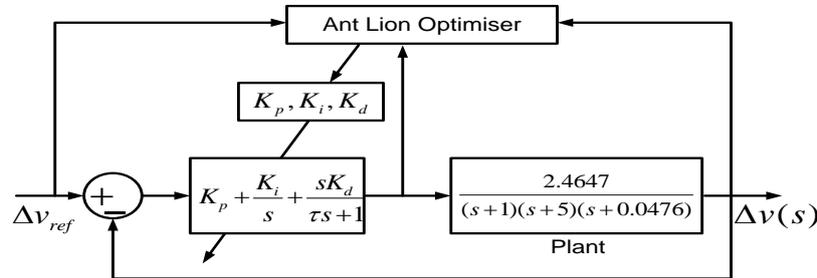


Fig. 4. The block representation of ALO tuned PID controller for ACCS system.

4.2. Proposed ALO-Based PID Parameter Design

In order to design an optimal PID gains, here different objective functions such as ISE, IAE, ITSE, ITAE and ISTSE are used. The best parameters of ALO tuned PID controller for above defined different objective functions are shown in Table 3. The step responses of linearized compensated system (shown in Fig. 4) for different optimal objective functions is shown in Fig. 5. From Fig. 5, it is evident that the controller parameters obtained from optimal ISE objective function yield better transient performance than other objective function. Thus the closed loop transfer function of the ACC system with the optimal PID parameter is given by

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.03855s^3 + 3.944s^2 + 8.989s + 4.668}{0.000375s^5 + 0.046s^4 + 0.8705s^3 + 5.915s^2 + 9.942s + 4.668} \quad (26)$$

Comparison with existing result

A comparative analysis of proposed ALO-PID algorithm with other recent published methods such as state space, fuzzy logic [7], genetic algorithm (GA) [10] and conventional PID techniques [6] are shown in Table 4. The comparative analysis are based on the transient performance, e.g., rise time (T_r), settling time (T_s), peak time (T_p), maximum overshoot ($\%M_p$) and steady state performance i.e. steady state error (E_{ss}). It is clear from Table 4 that the proposed ALO optimized PID controller shows better result as compared to GA, Fuzzy Logic, state space, and conventional PID methods in terms of transient and steady state performance. The step response for the cruise control system by using conventional, state space, fuzzy logic, genetic algorithm and the proposed ALO tuned PID controller is shown in Fig. 6. It is evident from Fig. 6 that the fuzzy-PID controller and GA tuned PID controller achieve better performance than the conventional and state space based PID controller. It is verified through simulation that, the fuzzy-PID and GA tuned PID achieve those performance by using high gain feedback which increases the control input. On the other hand, the proposed ALO tuned PID controller further improves the transient and steady state performance by even satisfying low gain feedback which makes the control input within the safe limit.

Table 3. ALO-PID controller ACCS parameters for various objective functions.

| Parameters/Objective Function | IAE | ISE | ITAE | ITSE | ISTSE |
|-------------------------------|---------|---------------|---------|---------|---------|
| K_p | 3.6112 | 3.8996 | 3.3498 | 3.7284 | 3.5603 |
| K_i | 0.12005 | 0.1705 | 0.21902 | 0.17362 | 0.13408 |
| K_d | 3.4743 | 3.9551 | 3.6732 | 3.5161 | 3.4155 |

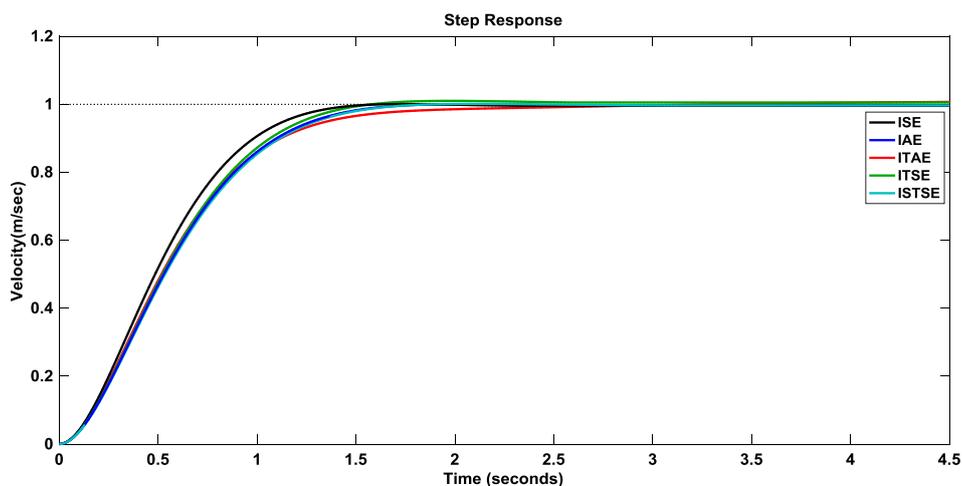


Fig. 5. Step response of the PID- Cruise Control System with different objective functions.

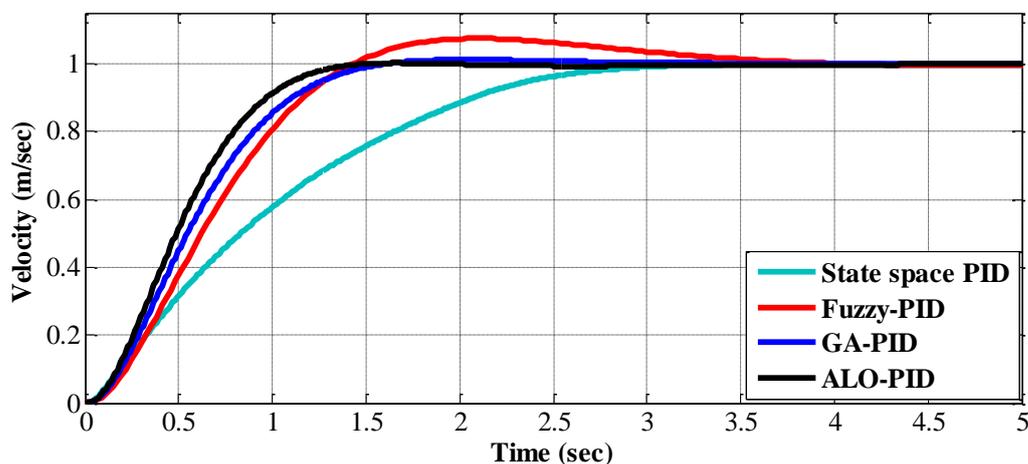


Fig. 6. The step response of the ACCS for different algorithms.

Table 4. Various tuning methods with performance parameter values.

| Control Methods in Cruise Control System | T_r (sec) | T_s (sec) | $\%M_p$ | E_{ss} |
|--|---------------|--------------|---------------|----------|
| PID [8] | 5.5 | 1.7 | 10.2 | 0.01 |
| State Space | 2.1 | 2.8 | 0 | 0.01 |
| Fuzzy Logic | 1.167 | 3.2 | 5 | 0.01 |
| GA [8] | 0.945 | 1.46 | 1.14 | 0 |
| ALO-IAE | 0.9060 | 1.4269 | 0.544 | 0 |
| ALO-ISE | 0.8161 | 1.258 | 0.1093 | 0 |
| ALO-ITAE | 0.9695 | 2.21 | 0.1228 | 0 |
| ALO-ITSE | 0.8895 | 1.3828 | 0.9442 | 0 |
| ALO-ISTSE | 0.925 | 1.46 | 0.4569 | 0 |

4.3. Root Locus Analysis

The root locus curves for cruise control system tuned by ALO algorithm is shown in Fig. 7. The values for closed loop and damping ratio of ALO-PID ACC system are given in Table 5. It has been observed that the closed loop poles reside to the left half of the s-plane. Hence, we can say the closed loop system is

stable. It is understandable from the results depicted in Table 6 that the conjugate poles obtained from ALO algorithm are more to the left on the s-plane and has highest damping ratio.

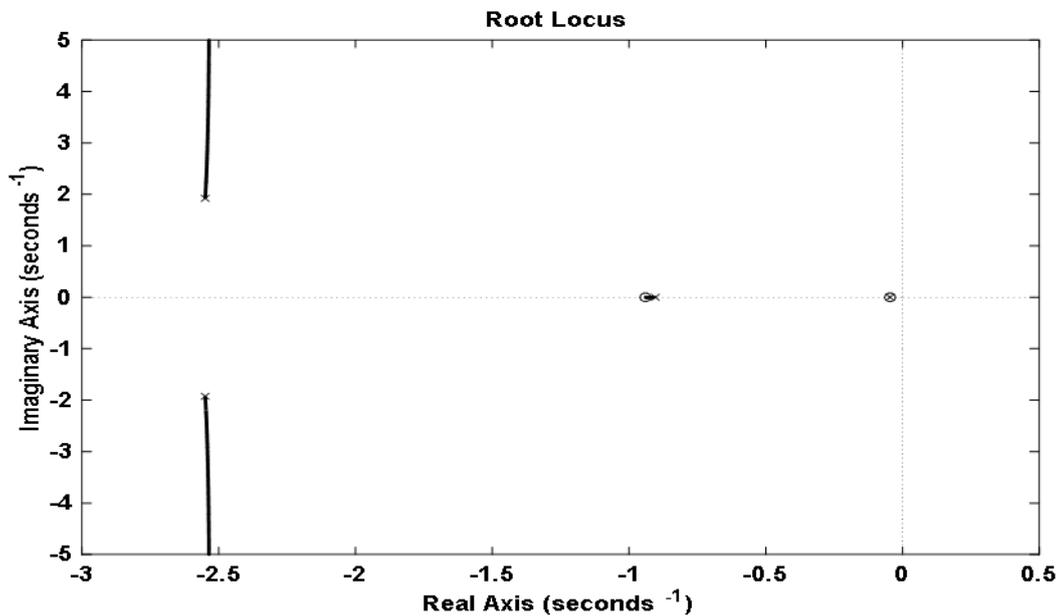


Fig. 7. Root locus curve of the ALO-PID cruise control system.

Table 5. Root locus analysis of ALO-PID ACCS.

| ALO-PID | |
|-------------------|---------------|
| Closed Loop Poles | Damping Ratio |
| -2.55+1.92i | 0.798 |
| -2.55-1.92i | 0.798 |
| -0.0458 | 1 |
| -0.904 | 1 |

4.4. Bode Analysis

The frequency response analysis by using bode plot for ALO tuned PID for the ACC system is shown in Fig. 8. In Table 6, the bandwidth, delay margin, phase margin and peak gain are presented for the ALO algorithm. In addition, the values have been compared with GA [10]. From bode plot, the minimum peak gain, maximum phase margin, delay margin and bandwidth are obtained. Therefore we conclude that, the ALO algorithm results the best frequency response.

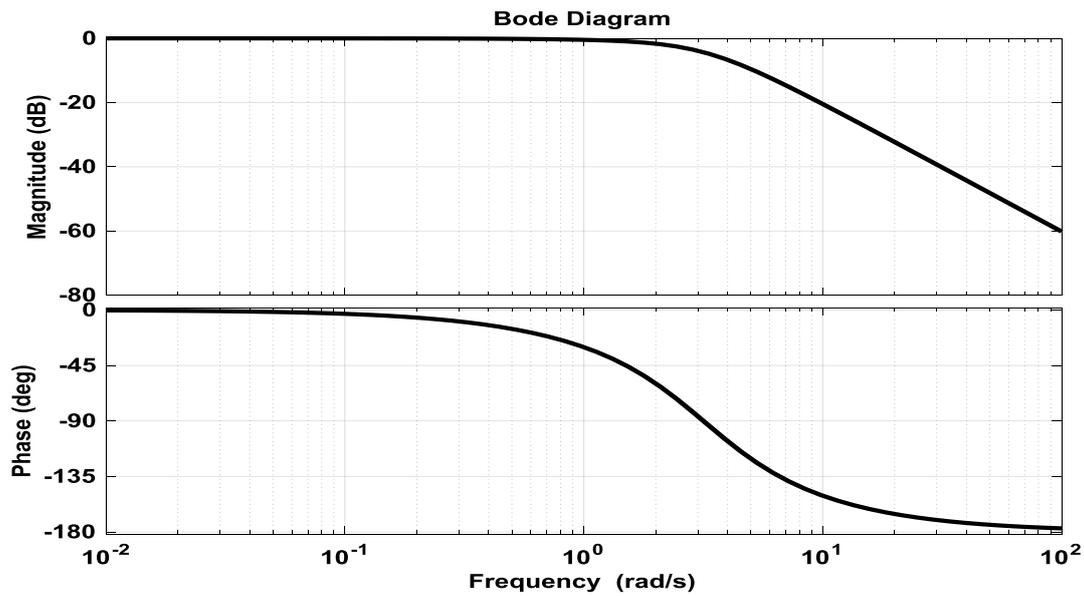


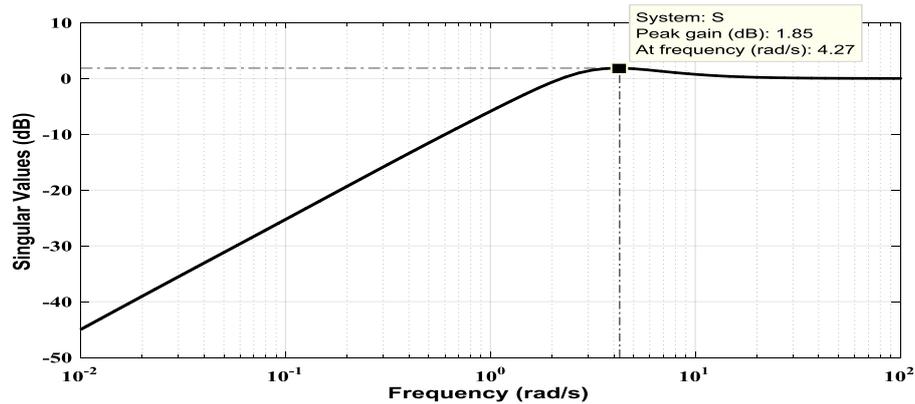
Fig. 8. Bode plot of the ALO-PID ACCS.

Table 6. Bode analysis.

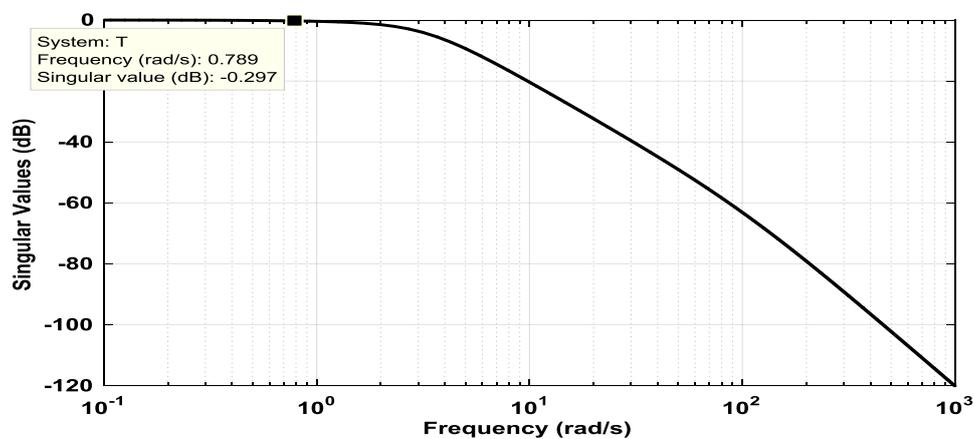
| Algorithms | Peak Gain (dB) | Phase Margin (deg) | Delay Margin (s) | Bandwidth |
|------------|----------------|--------------------|------------------|-----------|
| ALO | 0 | 180 | Inf | 2.63 |
| GA [10] | NA | 71.5 | Inf | NA |

4.5. Robustness Analysis

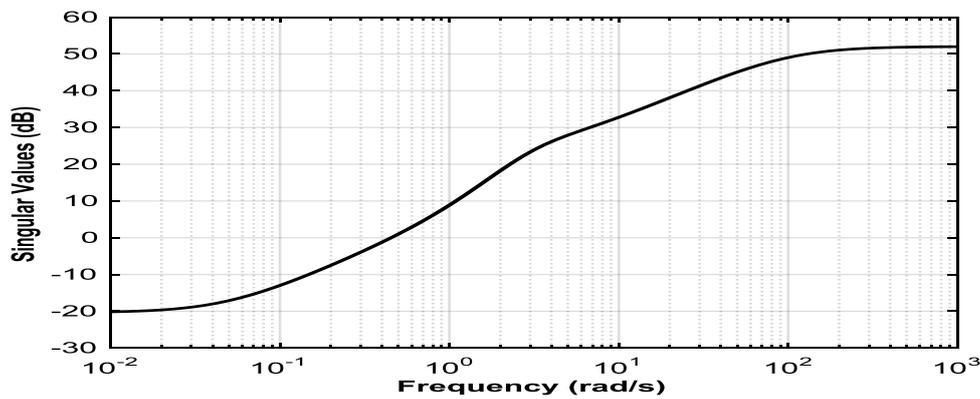
The robustness of a system is better captured by the singular value plots of the following transfer function: Sensitivity function $S(s) = \frac{1}{1+L(s)}$, Complementary function $T(s) = \frac{L(s)}{1+L(s)}$, Controller sensitivity $K(s)S(s)$, where $L(s) = G(s)K(s)$ is the loop transfer function and $K(s)$ is the PID controller. For the better robustness, the peak of the above transfer functions i.e. $\|S\|_\infty, \|T\|_\infty$ should be as small as possible (less than 2 or 6 dB) and at the same time, the gain of the sensitivity function should be less in low frequency region and for complementary sensitivity function, the gain should be less at high frequency region. As shown in Fig. 9 (a)-(c) the peak of the sensitivity is **1.23**, complementary sensitivity is **0.96** and there is no peak (within the frequency range 10¹ to 10³ rad/sec) in controller sensitivity plot. Therefore from the above parameter values, we can conclude that the close loop system is robust against any disturbances such as input output disturbances, parametric uncertainty etc.



(a) Sensitivity analysis of ACCS tuned by ALO



(b) Complementary sensitivity analysis of ACCS tuned by ALO



(c) Controller sensitivity analysis of ACCS tuned by ALO

Fig. 9. Different sensitivity plots for cruise control system.

5. Conclusion

This paper introduces a use of a recent optimization method named Ant Lion Optimizer to regulate the performance indices for automobile cruise control system. The ALO is used to evaluate the optimal tuning parameters for PID controlled cruise control system. The important contribution of the work includes:

- (i) Comparison of application of six objective functions such as IAE, ISE, ITAE and ITSE, ISTSE in the process to obtain the control parameters. From the results, it is evident that the ISE is a better choice for PID to optimize the control parameters.
- (ii) In terms of convergence characteristics, ALO exhibits promisingly better results for PID in terms of rise time, settling time, overshoot and steady state error.
- (iii) The time domain analysis, frequency domain analysis and robustness analysis has been carried out for the proposed system to show the supremacy in performance of the proposed algorithm over other recently reported methods.

The future scope of this work is the extension of the proposed method for design of a fractional order PID controller for an automobile cruise control system.

Acknowledgments

The authors would like to thank for using the facilities created in VSSUT out of AICTE sponsored RPS Project entitled “Transient Stability Analysis and Control of Power Systems with Excitation Control”-2012.

References

- [1] R. C Zhao, P. K. Wong, Z. C. Xie, and J. Zhao, “Real-time weighted multi-objective model predictive controller for adaptive cruise control systems,” *International Journal of Automotive Technology*, vol. 18, no. 2, pp. 279-292, April 2017.
- [2] M. H. Lee, H. G. Park, S. H. Lee, K. S. Yoon, and K. S. Lee, “An adaptive cruise control system for autonomous vehicles,” *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 3, pp. 373-380, March 2013.
- [3] Z. Situm D. Pavkovic, and B. Novakovic. “Servo pneumatic position control using fuzzy PID gain scheduling,” *J. Dyn. Sys., Meas., Control*, vol. 126, no. 2, pp. 376-387, 2004.
- [4] D. Corona and B. De Schutter, “Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 365-372, March 2008.
- [5] H. Fukuoka, Y. Shirai, and K. Kihei, “Driving support system adaptive to the driver state of surrounding vehicles: Simulation study on a rear-end precrash safety system,” *Transactions of the Society of Automotive Engineering of Japan*, vol. 40, no. 3, pp. 933-938, 2009.
- [6] N. Vedam, I. Diaz-Rodriguez, and S. P. Bhattacharya, “A novel approach to the design of controllers in an automotive cruise-control system,” in *Proceeding of the 40th Annual Conference of the IEEE on Industrial Electronics Society (IECON’14)*, USA, 29th Oct–1st Nov, 2014, pp. 2927-2932.
- [7] R. Muller and G. Nocker, “Intelligent cruise control with fuzzy logic,” in *Proceedings of the Intelligent Vehicles’92 Symposium*, Detroit, MI, 1992, pp. 173-178.
- [8] A. Morand, X. Moreau, P. Melchior, M. Moze, and F. Guillemard, “CRONE cruise control system,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 15-17, Jan. 2016.
- [9] H. Suzuki and T. Nakatsuji, “Effect of adaptive cruise control on traffic throughput: Numerical example on actual freeway corridor,” *JSAE Review*, vol. 24, no. 4, pp. 403-410, Oct. 2003.
- [10] M. K. Rout, D. Sain, S. K. Swain, and S. K. Mishra, “PID controller design for cruise control system using genetic algorithm,” in *Proceeding of International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT’16)*, India, 2016, pp. 4170-4174.
- [11] K. Osman, M. F. Rahmat, and M. A. Ahmad, “Modelling and controller design for a cruise control system,” in *Proceeding of 5th International Colloquium on Signal Processing & Its Applications*, Kuala Lumpur, 2009, pp. 254-258.
- [12] S. Miyata, T. Nakagami, S. Kobayashi, T. Izumi, H. Naito, A. Yanou, H. Nakamura, and S. Takehara, “Improvement of adaptive cruise control performance,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1-8, Sept. 2010. doi:10.1155/2010/295016
- [13] K. Osman, M. F. Rahmat, and M. A. Ahmad, “Modelling and controller design for a cruise control system,” presented at *5th International Colloquium on Signal Processing & Its Applications (CSPA)*, 2009. doi:10.1109/CSPA.2009.5069228.

- [14] C. Qiu, "A design of automobile cruise control system based on fuzzy PID," in *Proc. International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, 26th-28th April 2014, Sapporo, Japan, pp. 450-453. doi:10.1109/InfoSEEE.2014.6948151
- [15] R. Rajamani and C. Zhu, "Semi-autonomous adaptive cruise control systems," *IEEE Trans. on Vehicular Technology*, vol. 51, no. 5, pp. 1186-1192, September 2002.
- [16] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296-305, Feb. 2014.
- [17] E. S. Ali, S. A. Elazim, and A. Y. Abdelaziz, "Ant lion optimization algorithm for optimal location and sizing of renewable distributed generations," *Renewable Energy*, vol. 101, pp. 1311-1324, Feb. 2017.
- [18] A. E. Hassanien, H. Hefny, and P.W. Tsai, "Antlion optimization based segmentation for MRI liver images," in *Proceedings of the Tenth International Conference on Genetic and Evolutionary Computing*, Springer, Fuzhou City, Fujian Province, China, November 7-9, 2016, vol. 536, p.265-272.
- [19] M. Raju, L. C. Saikia, and N Sinha, "Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 52-63, Sept. 2016.
- [20] S. Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software*, vol. 83, pp. 80-98, May 2015.