

# Combination of particle swarm optimization algorithm and artificial neural network to propose an efficient controller for vehicle handling in uncertain road conditions

B. Aalizadeh<sup>1</sup> · A. Asnafi<sup>2</sup>

Received: 2 October 2015 / Accepted: 25 October 2016 / Published online: 19 November 2016  
© The Natural Computing Applications Forum 2016

**Abstract** Due to fast variation of desired variables in vehicle handling problem, design of an accurate applicable economic and considerably quick responsible controller for steering control of such systems has attracted much attention in the literature. The problem becomes more complicated, if the variation of road condition comes into play also. In this study, by combination of a simple PID and an optimized adaptive neural network controller, an arrangement for active front steering control of vehicles in different road frictions is proposed. A general PID controller is picked up and then optimized using the particle swarm optimization algorithm. After that, a neural network is added consecutively and trained by the outputs of PID controller and neural network toolbox of MATLAB software. The proposed controller fulfills both the applicability and efficiency due to dual use of PID and neural network controllers. Simulation results confirm the rightness of suggested controller in active steering control of vehicles even for unpredictable road friction.

**Keywords** Particle swarm optimization algorithm · Artificial neural networks · Vehicle handling · Active front steering control

## 1 Introduction

The steering system plays the substantial role in the dynamics, handling and specially the yaw stability control of all road vehicles. Due to possible sudden unpredictable and unexpected driving condition changing, having a sense of efficient steering control is unavoidable. As an applicable index, any change in the road coefficient of friction and its effect on the yaw stability control has been considered in some articles [1, 2]. Fast and accurate response to changes in road conditions enforces engineers to design and introduce efficient controls so that the answer is minimized for the probably late drivers. In the literature, several approaches such as direct yaw moment control, four-wheel steering system, active differential braking and active front steering control have been proposed to control the yaw rate of vehicles.

The active front steering (AFS) [2] arrangement is one of the efficient ones that works by adding an additional angle to that previously made by conventional steering system [3]. In the literature, several studies have been done using simple arrangement controllers to reach economical applicable solutions. For example, in [4], a fractional-order PID controller was designed as the AFS to control the yaw stability of a vehicle. Boada et al. [5] designed a fuzzy logic controller to control both the yaw rate and sideslip angle of a vehicle. Hajjaji and Bentalba [6] designed a Sugeno-type fuzzy logic controller for their control scheme. Masao et al. [7] designed a steering controller using the state feedback of both yaw rate and sideslip angle and utilizing active rear wheel steering and direct yaw moment control systems jointly. Zeyada et al. [8] developed a fuzzy logic controller to improve the handling of a vehicle by using both differential braking and active front steering control systems.

---

✉ A. Asnafi  
asnafi@shirazu.ac.ir

B. Aalizadeh  
b.aalizadeh@yahoo.com

<sup>1</sup> Department of Mechanical Engineering, Sama Technical and Vocational Training College, Islamic Azad University, Gachsaran Branch, Gachsaran, Iran

<sup>2</sup> Hydro-Aeronautical Research Center, Shiraz University, Shiraz 71348-13668, Iran

Others used almost simplified models for vehicle dynamics. Rezaei and Heidari Shirazi [9] designed a QFT (quantitative feedback theory) robust controller to control the yaw rate of a 7-degree-of-freedom vehicle by means of active front steering system. Their controller was adjusted for a range of offline road coefficients of friction; thus, it failed to modify the vehicle behavior completely and properly. Ma et al. [10] designed a fuzzy logic AFS controller for an eight-degree-of-freedom car. Although the degrees of freedom grew up, it did not lead to a precise behavior. Wu et al. in [11], used the back propagation of error as a learning method for a neural networks AFS controller. Then, an optimization was done using genetics algorithm, but the road condition was assumed unchanged during the maneuvers.

It can be found that some studies employed combined algorithms and methods to overcome the resultant errors. Masao et al. [12] designed an optimal control for a vehicle to enhance the handling quality of the front steering angle by controlled distribution of moment on all four wheels. Elbeheiry et al. [13] developed a sliding mode control that combines both active front steering and active roll moment control (ARMC) system in order to enhance vehicle stability and handling of vehicle in emergency maneuvers. Yihu et al. [14] designed a fuzzy logic controller to improve yaw stability of a vehicle by integrated yaw moment control and active front steering. Their study did not consider unpredictable situation, for example, changes in road friction.

Although these combination methods can improve the vehicle handling, they are not economical controllers and usually increase the overall cost of manufacturing and production.

Some works have been focused on improving the controller performances using superimposed methods. Zhanga et al. [15] optimized a fuzzy logic controller for a four-wheel steering vehicle by a combination of genetic algorithm, nonlinear programming quadratic line search (NPQL) and response surface model (RSM) in a fixed road condition.

Motoki and Masao designed a controller by using direct yaw moment control to manipulate the yaw rate of a vehicle in different road surface conditions. They used sideslip angle regulating as the feed forward and both the sideslip angle and yaw rate as the feedback signal [16]. Halvai et al. developed an adaptive neuro-fuzzy controller and studied the effect of sudden changes of road friction coefficient for right side wheels of the vehicle [17]. It is noted that the proposed controller can steer to desired conditions for almost limited variation.

Most of the controllers, which have been recommended in the literature, were based on the simplified vehicle dynamic models or assumed constant road and parameter conditions. Due to unpredictable and unexpected parameter changes, an

applicable and efficient controller must be developed in order to adjust itself with these various conditions. It is necessary to design a controller that adapts its parameters during maneuvers to improve the handling of vehicle.

In this study, an applicable and economic but simultaneously efficient and adaptive neural network controller was proposed. In order to design an efficient neural network controller, some problems must first be noticed and solved. First, it is noted that an efficient network has many neuron interconnections and unknown parameters that must be optimized. Second, the ranges of neural network's parameters are not initially specified. Third, during the training of the network, the parameters (layer weights) may be trapped in local optimums.

For not dealing with these problems, first a general PID controller with only three unknown parameters but almost specified ranges of variation is picked up. Then, utilizing the PSO (particle swarm optimization) algorithm, the PID controller parameters are optimized. PSO algorithm works with both the local and global solutions; therefore, the final global optimal solution can be reached. By means of the neural network toolbox in MATLAB software and data obtained from optimized PID controller, the neural network is trained. This integrated controller is applied to the vehicle, and again by using back propagation of error as a learning algorithm, the controller can modify its weights to improve the handling for different road conditions. The proposed controller satisfies both the applicability and effectiveness due to joint use of PID and adaptive neural network controllers.

## 2 Vehicle model description

In this section, both the reference and actual models for our system will be introduced. With respect to Fig. 1, the reference model is a 2D reduction of a real 3D vehicle whose two rear and front wheels were replaced by a set of wheels. In a real vehicle, the best condition for minimized slipping and optimal handling occurs when the wheels rotate about a point. This introduced model can satisfy these optimal conditions for desired handling simply. One advantage of this simple model is that the trajectory response of vehicle under step input of steering is a circle or the yaw rate is a constant value [18, 19].

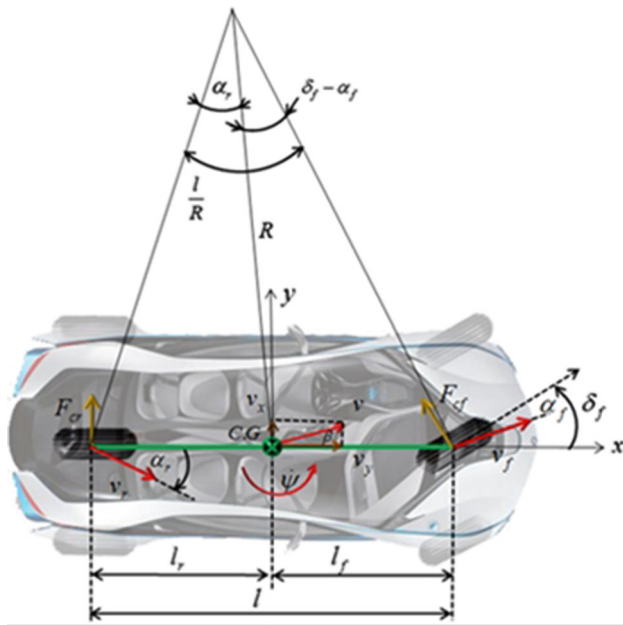
With reference to Fig. 1, the 2D governing equations of motion become:

$$m(\dot{v}_y + \dot{\psi}v_x) = F_{cr} + F_{cf} \cos \delta_f \quad (1)$$

$$m(-\dot{\psi}v_y) = -F_{cf} \sin \delta_f \quad (2)$$

$$I_z \ddot{\psi} = l_f F_{cf} - l_r F_{cr} \quad (3)$$

where  $m$  is the total mass of car,  $v_y$  and  $v_x$  are the velocities along  $y$ -direction and  $x$ -direction, respectively,  $F_{cr}$  and  $F_{cf}$



**Fig. 1** The schematic diagram of the reference model

are the lateral forces exerted to the rear and front wheels, \$\delta\_f\$ is the steering angle of front wheel, \$I\_z\$ is the mass moment of inertia, and \$\psi\$ is yaw angle.

\$l\_f\$ and \$l\_r\$ are the interspaces between front/rear wheels and the center of mass, \$l\$ is the distance between rear and front axles, and \$\alpha\_f\$ and \$\alpha\_r\$ are the lateral slip angle of front rear wheels, respectively.

Assuming the constant cornering stiffness for rear and front tire leads someone to rewrite the lateral forces as:

$$F_{cf} = C_f \alpha_f \Rightarrow F_{cf} = C_f \left( \beta + \frac{\dot{\psi} l_f}{v_x} - \delta_f \right) \tag{4}$$

$$F_{cr} = C_r \alpha_r \Rightarrow F_{cr} = -C_r \left( \beta - \frac{\dot{\psi} l_r}{v_x} \right) \tag{5}$$

where \$C\_f\$, \$C\_r\$ and \$\beta\$ are the cornering stiffness of front and rear wheels and sideslip angle of the car, respectively. By introducing \$\beta\$ and \$d\psi/dt\$ as the state variables, one can simply reach to such following state space equation between \$\delta\_f\$ and defined state variables:

$$\begin{bmatrix} \dot{\beta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{(C_f + C_r)}{mv_x} & -1 - \frac{(l_f C_f - l_r C_r)}{mv_x^2} \\ -\frac{(l_f C_f - l_r C_r)}{I_z} & -\frac{(l_f^2 C_f + l_r^2 C_r)}{mv_x^2} \end{bmatrix} \begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} C_f \\ \frac{mv_x}{l_f C_f} \\ I_z \end{bmatrix} \delta_f \tag{6}$$

A simple calculation results in the transfer function between the yaw rate and the steer angle of the front wheel as:

$$\frac{\dot{\psi}}{\delta_f} = \frac{A_1 s + A_2}{B_1 s^2 + B_2 s + B_3} \tag{7}$$

where \$s\$ is the Laplace operator and the coefficients \$A\_1\$, \$A\_2\$, \$B\_1\$, \$B\_2\$ and \$B\_3\$ are:

$$\begin{aligned} A_1 &= mv_x l_f C_f \\ A_2 &= LC_f C_r \\ B_1 &= mv_x I_z \\ B_2 &= I_z (C_f + C_r) + m(l_f^2 C_f + l_r^2 C_r) \\ B_3 &= \frac{C_f C_r}{v_x} (l_f + l_r)^2 \left[ 1 + \frac{mv_x^2 (l_r C_r - l_f C_f)}{C_f C_r (l_f + l_r)^2} \right] \end{aligned} \tag{8}$$

With respect to Fig. 1, it can be shown that:

$$\delta_f - \alpha_f + \alpha_r = \frac{l}{R} \rightarrow \delta_f = \alpha_f - \alpha_r + \frac{l}{R} \tag{9}$$

Also, by rewriting Eqs. (4) and (5), we have:

$$\alpha_f = \frac{F_{cf}}{C_f} = \frac{m_f v^2}{C_f R} \tag{10}$$

$$\alpha_r = \frac{F_{cr}}{C_r} = \frac{m_r v^2}{C_r R} \tag{11}$$

where \$m\_f\$ and \$m\_r\$ show the distributed masses on the rear and front axles in nonoperating situation. Substituting Eqs. (10) and (11) to (9), one can reach to:

$$\begin{aligned} \delta_f &= \frac{l}{R} + \left( \frac{m_f}{C_f} - \frac{m_r}{C_r} \right) \frac{v^2}{R} \\ &= \frac{l}{R} + \left( \frac{m_f}{C_f} - \frac{m_r}{C_r} \right) a_y \\ &= \frac{l}{R} + K_u a_y \end{aligned} \tag{12}$$

where \$a\_y\$ is the lateral acceleration and \$K\_u\$ is known as the understeering coefficient, with such following relation:

$$K_u = \frac{m_f}{C_f} - \frac{m_r}{C_r} = (\alpha_f - \alpha_r) \frac{R}{v^2} \tag{13}$$

Finally, using Eqs. (10)–(13), one can rearrange the transfer function between the yaw rate of vehicle and steering angle (Eq. (7)) as:

$$\dot{\psi} = \frac{v \delta_f}{l + k_u v^2} \tag{14}$$

Equation (14) prepares the reference signal for our control simulation arrangement, while, in this paper, a virtual but completely efficient software is used to model the actual plant. CarSim software as one of the MATLAB Simulink toolbox is used to model an actual vehicle with

**Table 1** The parameters of the vehicle used in our simulations

$m$	1530 (kg)	$l_f$	1.14 (m)	$R_w$	0.31 (m)
$I_z$	4607.5 (kg m <sup>2</sup> )	$l_r$	1.64 (m)	$I_w$	0.9 (kg m <sup>2</sup> )
$v$	72 (km/h)	$l_w$	1.55 (m)	$H$	0.52 (m)

real and different road conditions and other true situations. Many studies have used this software package to simulate the vehicle responses in their control schemes (see for example [12, 13, 20, 21]).

The parameters of the vehicle, we used in this study, are shown in Table 1. In this table,  $R_w$  is the nominal radius of the wheel,  $I_w$  is the mass moment of inertia of the wheel,  $h$  is the height of the center of gravity of the sprung mass, and  $l_w$  is the width between right and left wheels.

### 3 Controller design pattern

In this section, the logic and procedure of selecting and designing the proposed controller will be described. Since the whole controlling process happens in a very short time interval, it should be able to demonstrate more efficient and precise behaviors rather than simple conventional controllers.

Here, and due to simplicity, generality and widespread use of PID controllers, a general PID controller is picked up for a determined and fixed road condition. After using the particle swarm optimization method and an artificial neural network, the PID coefficients have been optimized and updated throughout such a smart planned process in real time. This advanced and smart combination can improve the handling of the vehicle in different road coefficients of friction in short intervals of time.

#### 3.1 Particle swarm optimization algorithm

The particle swarm optimization (PSO) is a random search algorithm, which was imitated from the social behavior of the bird flocks or fish schools, to find the best value of a cost function. This algorithm was first introduced by Kennedy and Eberhart [22] and can be used to optimize binary discrete optimization problems as well as the continuous ones [23]. The ability of algorithm in finding optimized candidate has been studied in some works (see for example [24–26]). In this optimization algorithm, first a set of population (swarm) of  $n$  particles was created (initialization step). Then, the fitness function for these particles was evaluated. The best position for each particle (personal best) and the best position among all particles (global best) were selected and be named as “*Pbest*” and “*Gbest*,” respectively [27, 28]. If the stopping criterion

was satisfied, the process stopped and the final *Gbest* was reported as the optimal population.

In our simulation, each particle includes three parameters, i.e., the coefficient of a general PID controller. They are named as  $x_{i1} = k_p$ ,  $x_{i2} = k_{int}$  and  $x_{i3} = k_d$  where  $i$  is the  $i$ th particle. First a set of 3D particles is created. Then, the fitness function (the integral of squared error) for these particles is evaluated and the “*Pbest*” and “*Gbest*” are computed. Each particle can be defined by its current velocity and position. The position of each particle changes according to the following update formula [28]:

$$x_{ij}(n+1) = x_{ij}(n) + v_{ij}(n+1) \quad (15)$$

where  $x_{ij}$  and  $v_{ij}$  are the position and the velocity of the parameter  $j$  of  $i$ th particle of the whole population. Also, the update rule for the velocity is [28]:

$$v_{ij}(n+1) = wv_{ij}(n) + c_1r_{1j}(n)[Pbest_{ij} - x_{ij}] + c_2r_{2j}(n)[Gbest_j - x_{ij}] \quad (16)$$

The first term at the right side of Eq. 16 is known as the inertia term and makes the particle moves in the same direction and with the same velocity. The second term which is known as “the personal influence” makes the particle come back to a possibly better previous position, while the third term which is known as “the social influence” makes the particle monitor the best neighbor direction. In these terms,  $r_{1j}$  and  $r_{2j}$  are some random values in the range of [0, 1], while  $w$ ,  $c_1$  and  $c_2$  are the weighting factors. These weighting factors enable one to manipulate the rate of convergence and also the magnitude of each effect in the searching process. See for example [29, 30] that presented some methods to select optimal values of above weighting factors. The flowchart of the PSO algorithm is schematically drawn in Fig. 2.

The constant values that are used in our PSO algorithm are also tabulated in Table 2.

#### 3.2 Proposed neural network controller

As it can be seen in many works, an artificial neural network is an imitation of making a decision with respect to previous trained behaviors and data in the biological human brain [31–33]. After an ANN has been trained by some defined and determined input/output signals, it may decide about a suitable response due to an unknown input [33]. The efficiency and correctness of this method has been discussed in many applications including control systems. See for example [34–36].

Figure 3 shows the architecture of a two-layer feed-forward neural network controller that is used in this study. In this model, all the activation functions ( $f$ ) of first layer are hyperbolic tangent sigmoid functions, while that of

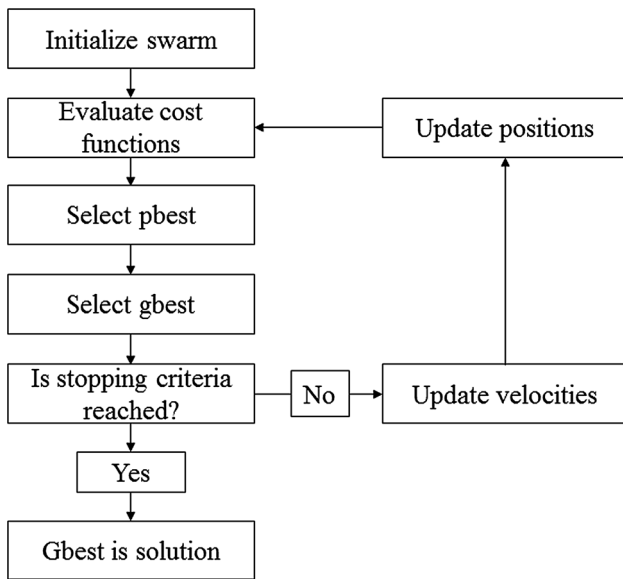


Fig. 2 The schematic flowchart of the PSO algorithm

Table 2 The values of parameters that are used in our PSO algorithm

Description	Notation (parameter)	Value
The initial population	$n$	100
Cognitive position acceleration	$c_1$	1.5
Social position acceleration	$c_2$	1
Weight of inertia	$w$	0.8
The stopping criterion (number of iteration)	–	1000

second layer are all linear. With reference to this architecture, the outlets of first layer are:

$$z_i = f \left( \sum_{j=1}^2 w_{ij} x_j \right) \quad (i = 1, 2, \dots, 30) \tag{17}$$

While the output signals of the second layer become:

$$u = f \left( \sum_{l=1}^{30} v_l z_l \right) \tag{18}$$

Here  $w_{ij}$ 's and  $v_l$ 's are the layer weights between the input, intermediate and output neurons.

During the simulation/training and in each iteration, the error signal is computed via the following relation:

$$e(n) = Y_{ref} - y(n) \tag{19}$$

and the cost function is defined as:

$$E = \frac{1}{2} e^2(n) \tag{20}$$

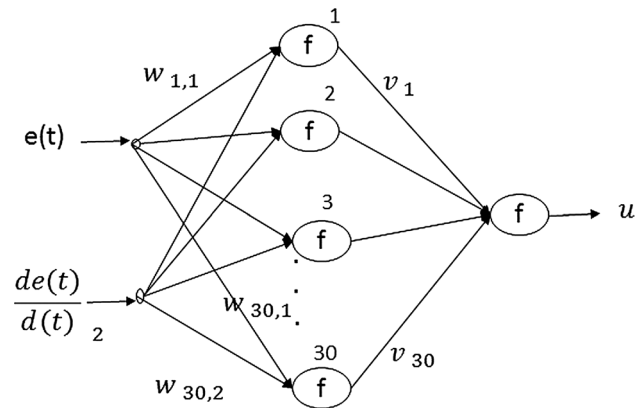


Fig. 3 The architecture of the network used in our simulation

To minimize the cost function (Eq. 20), the back-propagation method is used. The method was first introduced by Werbos [37] and then used in neural networks as a learning algorithm in [38]. Here, this method is used to update the weights iteratively to reach an optimal value for the desired cost function. In fact, the back-propagation algorithm together with the delta learning rule [39, 40] is used to compute layer weights for minimizing the cost function  $E$  (Eq. 20) in this study. Based on above methods, the weight variation in each iteration must be proportional to the minus gradient of the cost function with respect to that layer weight, i.e.,

$$\begin{aligned} \Delta w_{ij} &\propto - \frac{\partial E}{\partial w_{ij}} \stackrel{\text{Chain rule}}{=} - \frac{\partial E}{\partial e(n)} \frac{\partial e(n)}{\partial y(n)} \frac{\partial y}{\partial u} \frac{\partial u}{\partial w_{ij}} \\ \Delta v_i &\propto - \frac{\partial E}{\partial v_i} \stackrel{\text{Chain rule}}{=} - \frac{\partial E}{\partial e(n)} \frac{\partial e(n)}{\partial y(n)} \frac{\partial y}{\partial u} \frac{\partial u}{\partial v_i} \end{aligned} \tag{21}$$

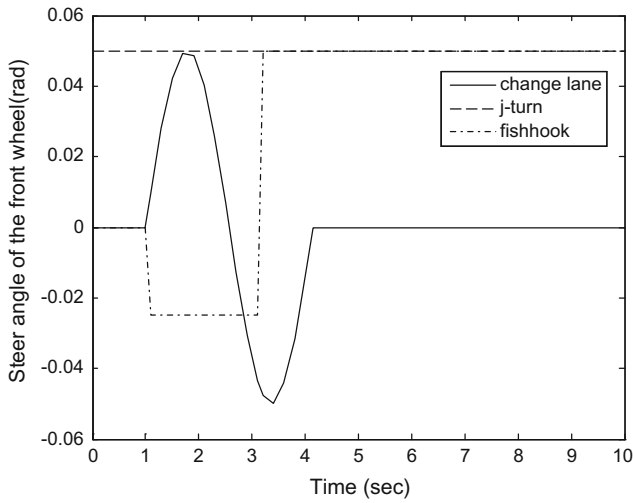
A simple calculation leads someone to reach such following relation for layer weight changing rule:

$$\begin{aligned} \Delta w_{ij} &= \eta e(n) \frac{\partial u}{\partial w_{ij}} \\ \Delta v_i &= \eta e(n) \frac{\partial u}{\partial v_i} \end{aligned} \tag{22}$$

where  $\eta$  is the coefficient of training. This coefficient must be selected such that both the stability of iterations and the speed of training will be satisfied. In this study,  $\eta = 0.0002$  satisfies both conditions; therefore, the updating law of layer weights becomes:

$$\begin{aligned} w_{ij}(n+1) &= w_{ij}(n) + \eta e(n) \frac{\partial u}{\partial w_{ij}} \\ v_i(n+1) &= v_i(n) + \eta e(n) \frac{\partial u}{\partial v_i} \end{aligned} \tag{23}$$

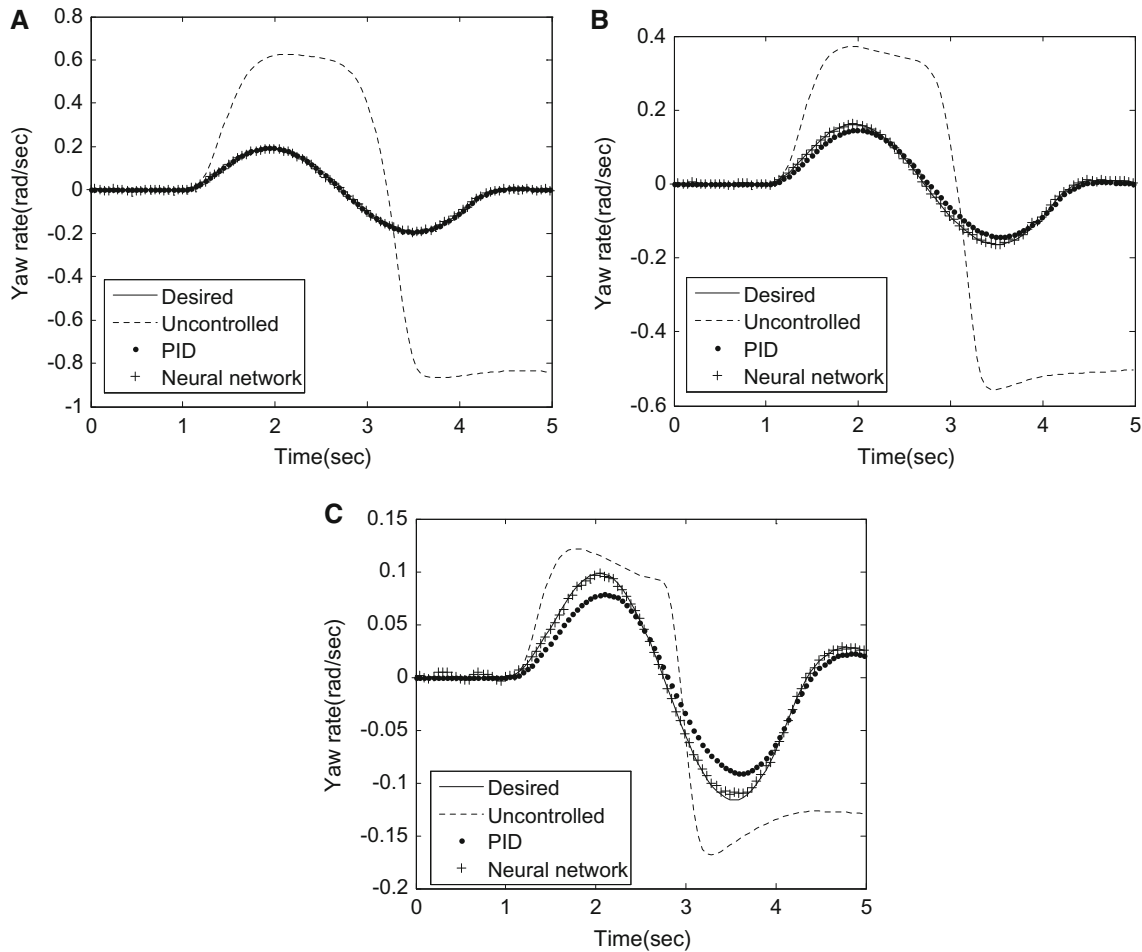
More specifically, first the parameters of a conventional PID controller, i.e.,  $K_p$ ,  $K_d$  and  $K_i$ , are optimized for a



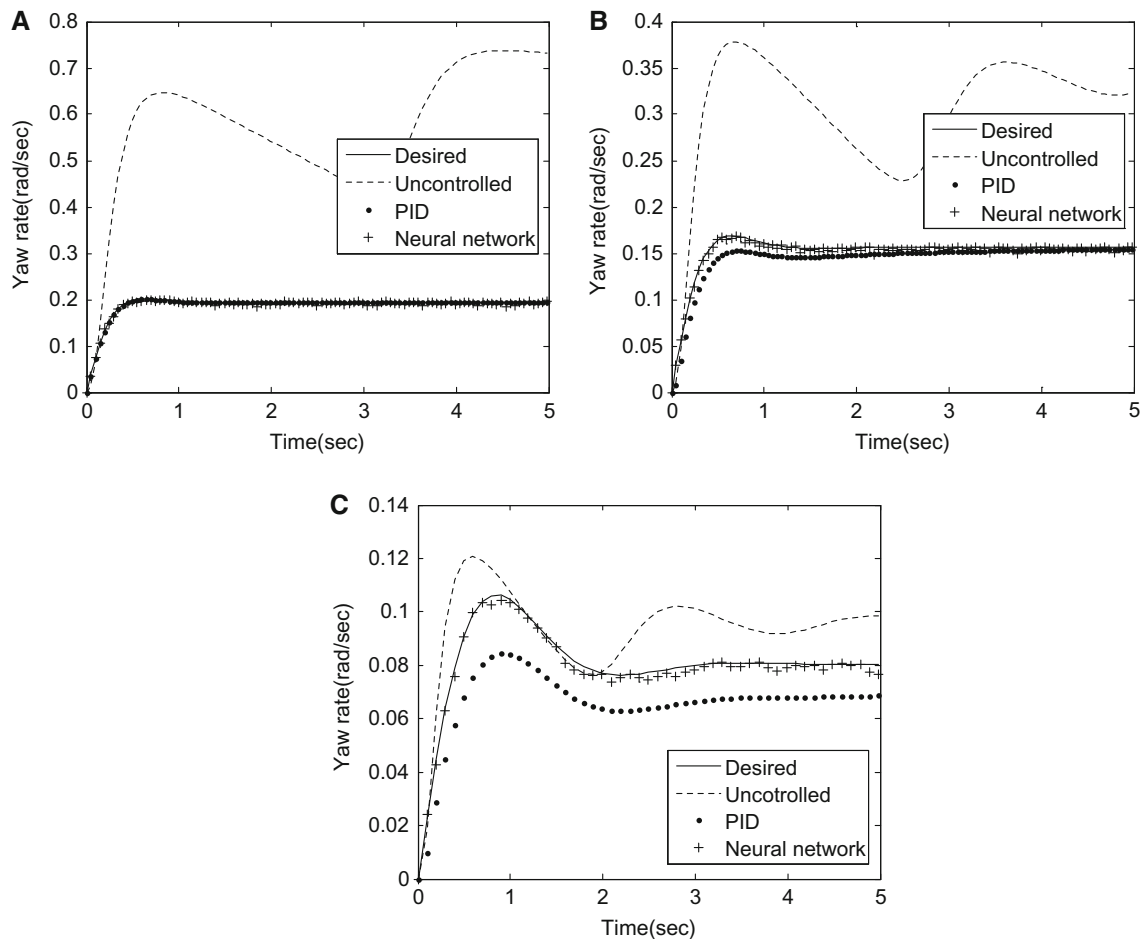
**Fig. 4** Three famous maneuvers that must be tracked in our simulations

specified road condition using particle swarm optimization algorithm. After that, train the NN controller to set the weights of NN controller, and then, the weights of NN are adjusted during the control process. Here the PSO (particle swarm optimization) algorithm has utilized to optimize the PID controller that must train the NN controller, and ANN (artificial neural network) has used to control the yaw rate of vehicle.

Briefly, the parameters of a conventional PID controller, i.e.,  $K_p$ ,  $K_d$  and  $K_i$ , are optimized when the road coefficient of friction is equal to one by using particle swarm optimization algorithm. After that, a two-layer neural network is designed and trained. The trained layer weights of the network can tune the controller signal for other road conditions. In the next section, the efficiency and effectiveness of the controller for three types of road conditions will be justified in real time for some famous vehicle maneuvers.



**Fig. 5** The desired, uncontrolled and controlled yaw rate of the vehicle for the change lane maneuver when the road coefficient of friction is **a** 1, **b** 0.6 and **c** 0.2



**Fig. 6** The desired, uncontrolled and controlled yaw rate of the vehicle for the *J*-turn maneuver when the road coefficient of friction is **a** 1, **b** 0.6 and **c** 0.2

### 4 Simulation results

In what follows, the adeptness and the ability of the designed controller for three famous maneuvers, i.e., *J*-turn, fishhook and change lane in three different road friction conditions, are investigated. Figure 4 shows these three famous maneuvers.

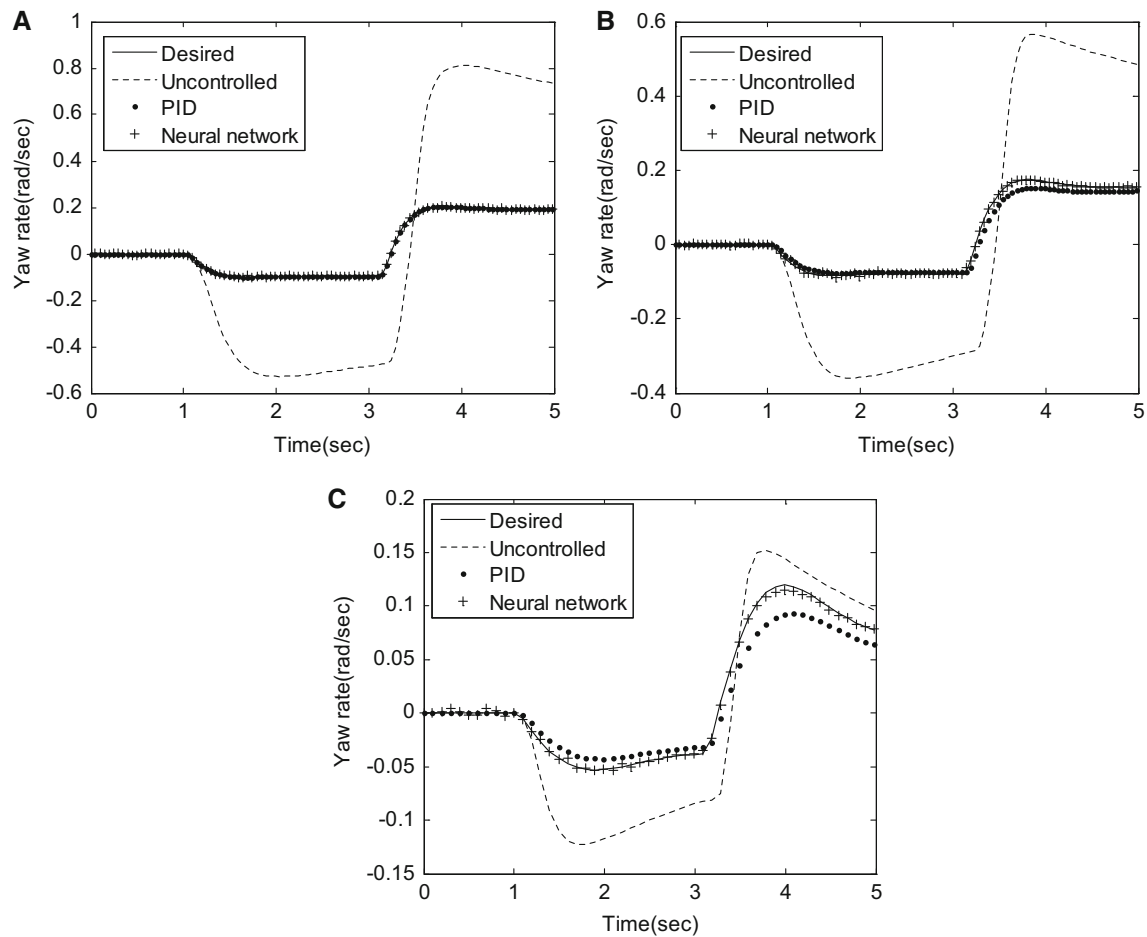
In this study, the coefficient of friction as an unpredictable condition was selected and varied from a road with coefficient of friction equal to one to a road with coefficient of friction equal to 0.2. As an analogical study, the behaviors of the system under the PID that was optimized by PSO and when it was integrated to ANN are compared. In the following figure legends, “PID” is the abbreviation of PID controller optimized by SFO, while “neural network” is the abbreviation of PID controller optimized by SFO and integrated by the ANN.

Figure 5 shows the desired, uncontrolled and controlled yaw rate of the vehicle for the change lane maneuver at different road conditions under mentioned controllers.

According to this figure, in first road coefficient of friction, i.e., the case (a) of Fig. 5, both the controllers can track the desired path, properly. It is completely expectable since the PSO algorithm is used to find optimized controller for this road condition only and the neural network has been trained at this road condition. Of course, when road coefficient of friction decreases, the optimized PID controller cannot control the vehicle alone and employing an auxiliary adaptive ANN is unavoidable as Fig. 5b, c indicates this matter.

The simulation is repeated for other two maneuvers, i.e., the *J*-turn and fishhook. Figures 6 and 7 show the superiority of the integrated optimized PID plus ANN controller with respect to optimized PID controller alone in tracking the desired trajectories.

In Table 3, one can compare the performance of the mentioned controllers of previous cases quantitatively and entirely.



**Fig. 7** The desired, uncontrolled and controlled yaw rate of the vehicle for the fishhook maneuver when the road coefficient of friction is **a** 1, **b** 0.6 and **c** 0.2

**Table 3** The integral of squared error between the desired and actual trajectories for our case studies

Maneuver/controller	Adaptive NN	PID + PSO
Coefficient of friction = 1		
Change lane	4.998 e(-5)	1.438 e(-6)
J-turn	8.182 e(-5)	3.859 e(-6)
Fishhook	5.081 e(-5)	2.004 e(-6)
Coefficient of friction = 0.6		
Change lane	3.857 e(-5)	0.0007666
J-turn	3.432 e(-5)	0.0004785
Fishhook	4.575 e(-5)	0.0008475
Coefficient of friction = 0.3		
Change lane	3.084 e(-5)	0.0009413
J-turn	2.741 e(-5)	0.001162
Fishhook	2.696 e(-5)	0.0009974

## 5 Conclusion

It was tried to introduce an active steering control scheme for vehicle handling that satisfies both the applicability and effectiveness criteria. An integrated optimized PID and artificial neural network controller was designed, and its performance was investigated for three famous yaw rate maneuvers in different road frictions. An analogical study was also done between the PID that was optimized by particle swarm algorithm and the entire represented controller. It was shown that in regular road friction, both the controllers can steer the vehicle well, but for rainy and icy roads, only the integrated PID plus adaptive ANN can follow the desired trajectory precisely.

### Compliance with ethical standards

**Conflict of interest** We declare that we have no conflicts of interest.



## References

1. Tseng HE, Madau D, Ashrafi B, Brown T, Recke D (1999) Technical challenges in the development of vehicle stability control system. In: Proceedings of the 1999 DEE international conference on cantml applications Kohala Coast-Island of Hawai'i, Hawsi'i
2. Rajamani R (2011) Vehicle dynamics and control. Springer, Berlin
3. Ackermann J, Bunte T, Odenthal D (1999) Advantages of active steering for vehicle dynamics control. In: 32nd international symposium on automotive technology and automation, Vienna, Austria
4. Li Q, Shi G, Lin Y, Wei J (2009) Yaw stability control of active front steering with fractional-order PID controller. In: International conference on information engineering and computer science. ICIECS 2009, pp 1–4
5. Boada BL, Boada MJL, Díaz V (2005) Fuzzy-logic Applied to Yaw Moment Control for Vehicle Stability, vol 43 (10). Taylor & Francis, London, pp 753–770
6. El Hajjaji A, Bentalba S (2003) Fuzzy path tracking control for automatic steering of vehicles. Robot Auton Syst 43(4):203–213
7. Nagai M, Hirano Y, Yamanaka S (1997) Integrated control of active rear wheel steering and direct yaw moment control. Veh Syst Dyn 27(5–6):357–370
8. Zeyada Y, Karnopp D, El-Arabi M, El-Behiry E (1998) A combined active-steering differential-braking yaw rate control strategy for emergency maneuvers. SAE Technical Paper 980230. doi:10.4271/980230
9. Rezaei M, Shirazi KH (2014) Robust handling enhancement on a slippery road using quantitative feedback theory. Proc Inst Mech Eng Part D J Automob Eng. doi:10.1177/0954407013518037
10. Ma J, Liu G, Wang J (2010) An AFS control based on fuzzy logic for vehicle yaw stability. In: 2010 International conference on computer application and system modeling (ICCSM), vol 5. IEEE
11. Wu S, Zhu E, Qin M, Ren H, Lei Z (2008) Control of four-wheel-steering vehicle using GA fuzzy neural network. In: 2008 International conference on intelligent computation technology and automation (ICICTA), vol 1. IEEE, pp 869–873
12. Nagai M, Shino M, Gao F (2002) Study on integrated control of active front steer angle and direct yaw moment. JSAE Rev 23(3):309–315
13. Elbeheiry EM, Zeyada YF, Elaraby ME (2001) Handling capabilities of vehicles in emergencies using coordinated AFS and ARMC systems. Veh Syst Dyn 35(3):195–215
14. Yihu W et al (2007) A fuzzy control method to improve vehicle yaw stability based on integrated yaw moment control and active front steering. In: International conference on mechatronics and automation, 2007. ICMA 2007. IEEE
15. Zhang J et al (2007) A fuzzy control strategy and optimization for four wheel steering system. In: IEEE international conference on vehicular electronics and safety, 2007. ICVES. IEEE
16. Shino M, Nagai M (2001) Yaw-moment control of electric vehicle for improving handling and stability. JSAE Rev 22(4):473–480
17. Niasar AH, Moghbeli H, Kazemi R (2003) Yaw moment control via emotional adaptive neuro-fuzzy controller for independent rear wheel drives of an electric vehicle. In: Proceedings of 2003 IEEE conference on control applications, 2003. CCA 2003, vol 1. IEEE, pp 380–385
18. Jazar RN (2013) Vehicle dynamics: theory and application. Springer, Berlin
19. Wong JY (2001) Theory of ground vehicles. Wiley, London
20. Benekohal RF, Treiterer J (1988) CARSIM: car-following model for simulation of traffic in normal and stop-and-go conditions. Transp Res Rec 1194:99–111
21. Kinjawadekar T et al (2009) Vehicle dynamics modeling and validation of the 2003 ford expedition with esc using carsim. No. 2009-01-0452. SAE Technical Paper
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference neural networks, vol IV. Perth, Australia, pp 1942–1948
23. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: Proceedings of the conference on systems, man, and cybernetics SMC97, pp 4104–4109
24. Shi Y, Eberhart, RC (2001) Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 congress on evolutionary computation, 2001, vol 1. IEEE, pp 101–106
25. Robinson J, Rahmat-Samii Y (2004) Particle swarm optimization in electromagnetics. IEEE Trans Antennas Propag 52(2):397–407
26. Allaoua B, Laoufi A, Gasbaoui B, Abderahmani A (2009) Neuro-fuzzy DC motor speed control using particle swarm optimization. Leonardo Electron J Pract Technol 15:1–18
27. Clerc M (1999) The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation, 1999. CEC 99, vol 3. IEEE
28. Khalil TM, Youssef HKM, Abdel Aziz MM (2006) A binary particle swarm optimization for optimal placement and sizing of capacitor banks in radial distribution feeders with distorted substation voltages. AIML 06 international conference, Sharm El Sheikh, Egypt, pp 129–135
29. Shi Y, Eberhart R (1998) Parameter selection in particle swarm optimization. In: Proceedings of the 7th annual conference evolutionary program, pp 591–600
30. Zhang L-P, Yu H-J, Hu S-X (2005) Optimal choice of parameters for particle swarm optimization. J Zhejiang Univ Sci 6A(6):528–534
31. Fullér R (2000) Neural fuzzy systems. In: Advances in soft computing series. Springer, Berlin
32. McCulloch WS, Pitts WA (1943) A logical calculus of the ideas imminent in nervous activity. Bull Math Biophys 5:115–133
33. Hebb DO (1949) The organization of behavior. Wiley, New York
34. Noriega JR, Wang H (1998) A direct adaptive neural-network control for unknown nonlinear systems and its application. IEEE Trans Neural Netw 9(1):27–34
35. Ge SS, Hang CC, Zhang T (1999) Adaptive neural network control of nonlinear systems by state and output feedback. IEEE Trans Syst Man Cybern Part B Cybern 29(6):818–828
36. Gomi H, Kawato M (1993) Neural network control for a closed-loop system using feedback-error-learning. Neural Netw 6(7):933–946
37. Werbos P (1974) Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University
38. Rojas R (1996) Neural networks. Springer, Berlin
39. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
40. Kriesel D (2011) A brief introduction to neural networks. Available at <https://www.dkriesel.com>