Journal of the Brazilian Society of Mechanical Sciences and Engineering (2018) 40:104
https://doi.org/10.1007/s40430-017-0945-z

TECHNICAL PAPER

Path tracking controller of an autonomous armoured vehicle using modified Stanley controller optimized with particle swarm optimization

Noor Hafizah Amer[1,2] · Hairi Zamzuri[2] · Khisbullah Hudha[1] · Vimal Rau Aparow[1] · Zulkiffli Abd Kadir[1] · Amar Faiz Zainal Abidin[3]

Received: 22 June 2017 / Accepted: 5 December 2017 / Published online: 31 January 2018

Abstract

This study presents the development and optimization of a proposed path tracking controller for an autonomous armoured vehicle. A path tracking control is developed based on an established Stanley controller for autonomous vehicles. The basic controller is modified and applied on a non-linear, 7degree-of-freedom armoured vehicle model, and consists of various modules such as handling model, tire model, engine, and transmission model. The controller is then optimized using particle swarm optimization algorithm to select the optimum set of controller parameters. The main motivation of this study is that implementation of path tracking control on an autonomous armoured vehicle is still very limited and it is important to have a specific study on this field due to the different dynamics and properties of the armoured vehicle compared to normal passenger vehicles. Several road courses are considered and the performance of the developed controller in guiding the vehicle along these courses was compared against the original Stanley Controller. It was found that the optimized controller managed to improve the overall lateral error throughout the courses with 24–96% reduction in lateral error. Also, the optimization for the proposed controller was found to converge faster than its counterpart with up to 93% better solution.

Keywords Particle swarm optimization · Armoured Vehicle · Autonomous path tracking · Path tracking · Stanley controller

Nomenclature

CG	Vehicle's centre of gravity
X	Global vehicle position in X axis (m)
Y	Global vehicle position in Y axis (m)
x	Local vehicle position in x axis (m)
y	Local vehicle position in y axis (m)
A_x	Lateral acceleration in vehicle local coordinates (ms^{-2})
A_y	Longitudinal acceleration in vehicle local coordinates (ms^{-2})
F_x, F_y, F_z	Forces in vehicle local coordinates direction (N)
θ	Vehicle's inclination w.r.t ground as shown in Fig. 1 (rad)
C_R	Distance between gun and vehicle's CG (m)
φ	Firing angle w.r.t. vehicle's longitudinal axis (rad)
β	Vehicle's moving direction w.r.t. vehicle's longitudinal axis (rad)
R	Radius of wheel (m)
Alpha, α	Lateral slip angle (rad)
T_a	Engine acceleration torque (Nm)
Omega, ω	Wheel rotational speed (rad/s)
V_x	Longitudinal vehicle velocity(m/s)

Technical Editor: Victor Juliano De Negri.

Throughout this paper, all of the symbols used and its definitions were made consistent with authors' previous papers Amer et al. [9] and [38]. Main definitions can be referred to Figs. 1 and 3. Unless stated otherwise, definitions can be summarized in nomenclature.

✉ Noor Hafizah Amer
noorhafizah@upnm.edu.my

[1] Faculty of Engineering, Universiti Pertahanan Nasional Malaysia, 57000 Kuala Lumpur, Malaysia

[2] Malaysia-Japan International Institute of Technology, University Technology Malaysia, 54000 Kuala Lumpur, Malaysia

[3] Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka, 76100Durian Tunggal, Melaka, Malaysia

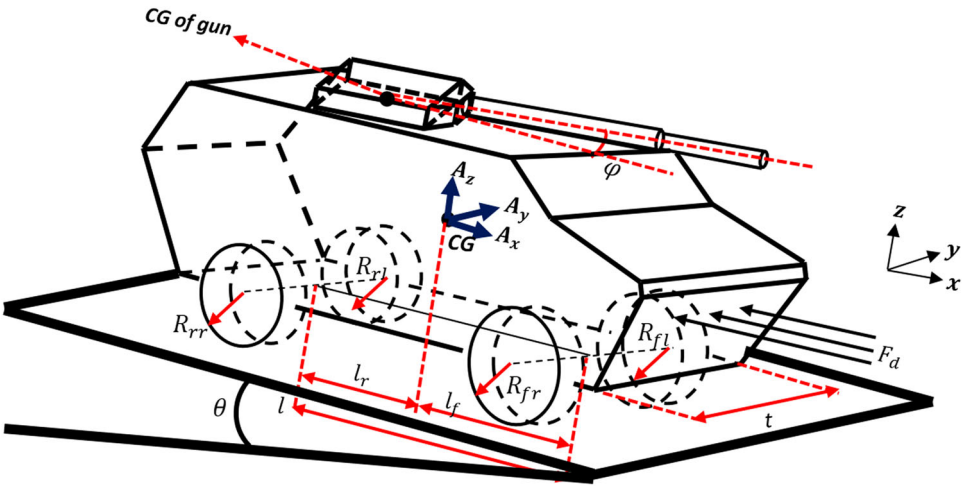

Fig. 1 Armoured vehicle model [9, 30, 31]

δ	Steered wheel angle (rad)
Y_{traj}	Y-position of a point on path nearest to the vehicle (m)
ψ	Vehicle's yaw angle (rad)
$\dot{\psi}$	Vehicle's yaw rate (rads^{-1})
ψ_{traj}	Path yaw angle (rad)
$\dot{\psi}_{traj}$	Path's yaw rate (rads^{-1})
ϕ	Heading error, $\psi - \psi_{traj}$ (rad)
ϕ	Heading error, $\psi - \psi_{traj}$ (rad)
e	Lateral error, $Y_{traj} - Y$ (m)
k_ϕ	Tuned controller gain
k_1	Gain for ϕ
k	Tuned controller gain (s^{-1})
k_ψ	Tuned controller gain (s)
v	Instantaneous vehicle velocity (m/s)

1 Introduction

In military application, usage of autonomous vehicle can be beneficial in terms of survivability of military personnel involved and accessibility of the vehicle to access hardly reached regions. Using autonomous vehicle, the possibilities of compromising personnel safety during missions are relatively low compared to the conventional ways where soldiers carrying out missions on foot. While autonomous technology for normal vehicle are easily found, the same technology for military vehicles have been very limited [1]. Therefore, this study is aimed to provide an academic insight on path tracking control for automatic steering of an armoured vehicle system.

In path tracking for autonomous vehicle, control strategies can be categorized into several types namely geometric/kinematic controller which developed based on the geometric/kinematic properties of the vehicle such as velocities and dimensions [2, 3]; dynamic controller which considers the dynamic properties of the vehicle such as forces and torques [4, 5]; classical controller such as sliding mode [6, 7] and PID approaches [8, 9]; and intelligent or adaptive controller which may include model predictive control (MPC) [10, 11]. Many studies proposed geometric or kinematic controller in their setup due to its simple mechanism and easy to implement.

Perhaps, one of the most popular geometric controllers is Pure Pursuit which has been proven to be efficient with low computational cost [12–14]. However, the controller requires careful selection of look-ahead distance since it will relate directly to the steering command and future path for vehicle. To overcome this, Shan et al. [15] employed a fuzzy controller to tune the look-ahead distance and clothoid curve fitting to create a smooth curvature for the vehicle to undertake to reach the desired path. Regardless, tuning the look-ahead distance is still a trivial issue for this type of controller and good balance between stability and tracking performance is usually not easy to achieve [3]. This controller was also observed to neglect the dynamics of vehicle [16].

Stanley controller is another effective geometric controller. Compared to Pure Pursuit, careful tuning of the look-ahead distance can be avoided and it considers the vehicle dynamics and instantaneous vehicle velocity. It has led the Stanford team and their autonomous vehicle, Stanley to win the DARPA Challenge 2005 with average 0.1 m cross track lateral error [17] and performed surprisingly well against other popular controllers [3, 18]. It is preferable mainly to avoid careful selection of look-ahead distance usually associated with other geometrical controllers such as Pure Pursuit. Also, it is easier to implement with simpler structure compared to other intelligent controllers such as model predictive control (MPC). However, going through the Stanley applications, one can notice that

most applications are applying the simplified, linear version of the controller as presented in Hoffmann et al. [17]. The original publication also presented an extended version of the controller that includes yaw rate term which can compensate the absence of look-ahead distance and provide future state of the path. In this study, the extended version is considered and modified to be applied on the armoured vehicle model.

To optimize controller parameters, particle swarm optimization (PSO) algorithm was chosen. It is one of the many nature inspired optimization algorithms available as research tools, such as Genetic Algorithm (GA), ant colony optimization (ACO), artificial bee colony algorithm (ABC), and gravitational search algorithm (GSA). A comprehensive study on these algorithms and their optimization performance can be found in [19] which stated that PSO performed relatively better in terms of convergence and consistencies compared to the other evolutionary algorithms stated above. A study by Zhang [20] also concluded that PSO was better in finding global solution with greater precision compared to conventional optimum search methods. Compared against genetic algorithm (GA), PSO was easier to implement with less parameters to adjust. Overall, the study concluded that PSO provides better accuracy with a fast convergence with a standard computing capability. In terms of controller parameters tuning, recent study by Merabti et al. [11] studies different meta-heuristic searching algorithm in tuning a model predictive controller using PSO, ACO, and GSA. The study proved the superiority of PSO which produced better controller performance compared to the others. PSO was shown to carry out optimization tasks within up to 90% quicker duration compared to ACO and GSA algorithms. Based on these advantages, this study adopts PSO in the optimization task of controller parameters. Other than this, the algorithm has been successfully utilized to optimize PID controller's parameters [9, 21, 22] as well as LQR controller's parameters [23] previously. In each case, the algorithm has managed to find the optimum parameters in improving the controller's performance.

In a nutshell, PSO is a meta-heuristic approach to solve optimization problems by emulating the motion of particles moving in swarms. Introduced in 1995 [24], the algorithm has seen various applications and advancements since then. Basically, the algorithm mimics the behavior of particles in swarm with randomly assigned initial position that moves together towards the most optimum position. Each of the particles will have the memory of its own best position, *pbest* and the overall swarm best position, *gbest* based on the optimum fitness value. The memories will be considered in determining particle's motion for its next position. This is to ensure that the agents are not too quick to move towards the new position which can avoid entrapment in local optimum solutions. Over several iterations, the whole swarm will find an optimum position (solution). The algorithm by Eberhart, Kennedy [24] was improved by Shi, Eberhart [25] by considering the inertial weight of each particle. To further ensure that the solution will not trap within a local optimum, near-neighbor interactionve has been introduced [26] by considering best position within a sub-swarm of neighbors "near" the particle.

The main aim of this study is to develop an automatic steering controller for an autonomous light armoured vehicle (LAV) to guide the vehicle along a pre-defined path. It is aimed to have a simple and robust controller with minimum computational cost due to the nature of armoured vehicle's environment with limited capability in storing high computational power on board. Due to the proven effectiveness and simpler mechanism [3, 18], Stanley Controller is chosen for implementation on the LAV. Further modifications are proposed to ensure the controller to operate properly with the armoured vehicle system. The controller is tuned using PSO algorithm in order to find the optimum set of parameters. Then, six trajectories are chosen to test the controller capabilities in guiding the autonomous LAV.

Besides the proposed controller and the optimization procedure, the main contribution of this paper is on the analysis of the controller on different type of trajectories. Most controllers catered large curvature roads. Previous studies have proven that controller might face problem with sharp corners and have to rely on path planner to provide smooth curvature [3, 18, 27]. On tracking controller, most studies have considered straight [28] and large curvature roads [18] where the sharpness of the path is still a problem to solve [29]. In this study, the road trajectories used are considering long roads with large curvature as well as roads with sharp maneuverings to evaluate the optimized controllers. Also, the controllers are applied on an armoured vehicle model, instead of a normal passenger vehicle which is a minor contribution of this paper.

This paper is organized based on the works undertaken in this study. First, introduction section covers brief background on previous works on the field. Then, Sect. 2 explains the vehicle model used to simulate the vehicle's behavior and the road courses used to test the controller. Then, the proposed controller is explained in Sect. 3. The next section describes the optimization procedure for all the controller's parameters followed by results and discussions in Sect. 5. Conclusion for this work and future recommendation are presented in the next section. The last few sections present all supplementary information for this paper such as acknowledgement and nomenclature.

2 Vehicle and path modeling

2.1 Vehicle model development

Previous work from authors [30] has demonstrated the mathematical derivation, development, and verification of a nonlinear full armoured vehicle model aimed mainly for lateral simulation studies. This model is equipped with 7DOF handling model, engine model, tyre model, kinematic model, calculation of lateral and longitudinal slips, and load distribution model, which has been used for active front steering study [31]. The same armoured vehicle model also has been used for path tracking control previously [9]. Figure 1 shows the armoured vehicle model where all symbols are explained in nomenclature section. It is worth noting that extra caution should be given on the axes notation, x, y, and z which represent the moving local coordinate for vehicle with origin at vehicle's centre of gravity. This should not be confused with the fixed global coordinate axes, X and Y which are usually associated with the Earth's latitude or longitude.

Figure 2 shows the basic configuration of the armoured vehicle model. In order to limit the length of this section, only main equations are shown herewith. Detailed descriptions and full vehicle parameters can be found in Aparow et al. [30].

2.1.1 Handling model (7DOF)

Handling model describes the dynamics of the vehicle in longitudinal plane as depicted in Fig. 3. Newton–Euler method was used to derive the equations of motion for longitudinal motions in x-direction, lateral motions in y-direction and yaw rotational motions about z-axis which shown in Eqs. (1), (2), and (3), respectively.

$$\sum F_x = ma_x F_{xrr} + F_{xrl} + F_{xfl}\cos\delta - F_{yfl}\sin\delta + F_{xfr}\cos\delta - F_{yfr}\sin\delta + mg\sin\theta - F_d - F_R\cos\varphi = m_b a_x \quad (1)$$

Here, F_d is the drag force which considers the air resistance, F_{air} with air density, ρ, frontal cross-sectional area, A, and vehicle's drag coefficient, C_d, as well as rolling resistance, F_d, with tire rolling resistance, C_d.

$$F_d = F_{air} + F_{roll} = \frac{1}{2}\rho A C_d\left(v_x^2\right) + mgC_r(v_x)$$

$$\sum F_y = ma_y F_{yrr} + F_{yrl} + F_{xfl}\sin\delta + F_{xfr}\sin\delta + F_{yfl}\cos\delta + F_{yfr}\cos\delta - F_R\sin\varphi = m_b a_y, \quad (2)$$

Next, considering yaw motion with yaw displacement, ψ about z-axis,

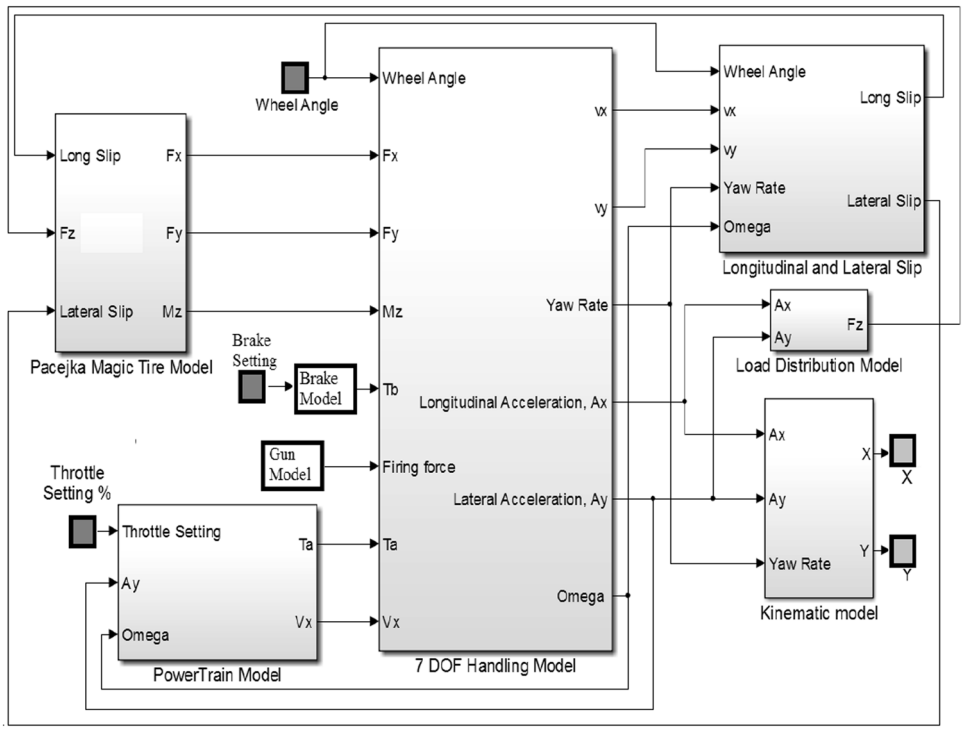

Fig. 2 Configuration of armoured vehicle model

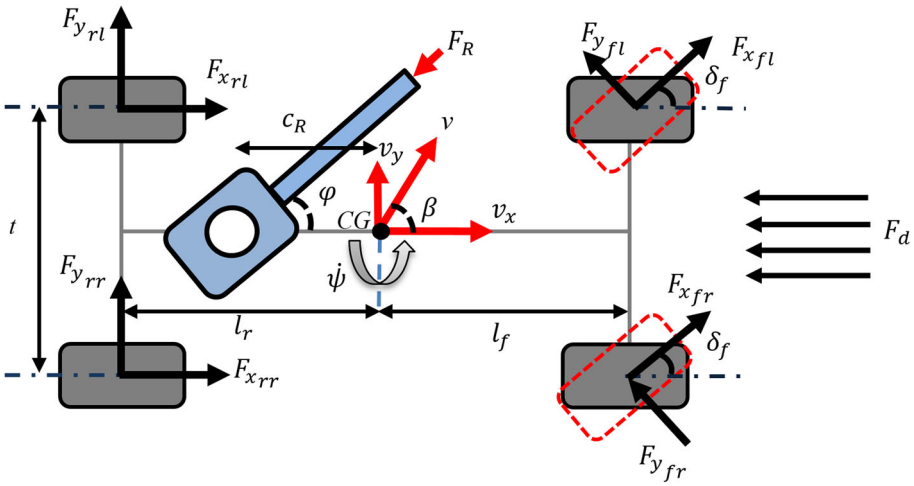

Fig. 3 Lateral model in longitudinal plane [9, 30]

$$\sum M_z = I_{CG}\ddot{\psi}\sum M_{zij} + \left[-F_{yrr} - F_{yrl}\right]l_r + \left[F_{xfl}\sin\delta + F_{yfl}\cos\delta + F_{xfr}\sin\delta + F_{yfr}\cos\delta\right]l_f + \left[F_R\sin\varphi\right]c_R + \left[F_{xfr}\cos\delta - F_{xfl}\cos\delta + F_{yfl}\sin\delta - F_{yfr}\sin\delta - F_{xrl} + F_{xrr}\right]\frac{t}{2} = I_{CG}\ddot{\psi} \quad (3)$$

Here, M_{zij} denotes the self-aligning moments on each wheel where $i = front,\ rear$ and $j = left,\ right$ and I_{CG} is the moment of inertia of the sprung mass about z-axis.

Another 4DOFs are from the rotational motion of each separate wheels about its y-axis, each with rotational speed, ω.

$$\sum M_{y,ij} = I_{y,ij}\dot{\omega}_{ij}$$
$$\tau_{eij} + \tau_{rij} - \tau_{bij} - \tau_{dij} = I_{y,ij}\dot{\omega}_{ij} \quad (4)$$

Here, subscripts ij corresponds to the notation before. τ_r is the reaction torque due to traction force generated from longitudinal force and τ_d is the friction torque for a wheel. τ_b is the braking torque and τ_e is the engine torque supplied to driving wheels. Determining τ_b and τ_e requires one to develop engine and brake system model. Full derivation of these two models can be found in studies by Aparow et al. [32] and [30].

2.1.2 Tire modeling and vertical load distribution model

In this study, an established tire model namely Pacejka model or Magic Equation [33, 34] is used to determine tire contact forces, F_x and F_y, and self-aligning moment, M_z. General function for F_x and F_y, and M_z is shown in Eq. (5) where F_z, α, and σ are the tire vertical force, lateral and longitudinal slips, respectively, and D, C, B, ϕ and S_v represent the properties of the function, respectively.

$$P(F_z, \alpha, \sigma) = D\sin(C\arctan(B\phi)) + S_v \quad (5)$$

Vertical forces acting on each tire are estimated using vertical load distribution model as described in Eq. (6), and outlined by earlier studies [30, 35, 36]. The notations are as described in Figs. 1 and 2.

$$F_{z,fj} = \left[\frac{mg}{2t}(l_r\cos\theta + h\sin\theta)\right] \pm \left[ma_y\left(\frac{h}{t}\right)\left(\frac{l_f}{l}\right)\right] - \left[\frac{ma_x}{2}\left(\frac{h}{l}\right)\right]$$
$$F_{z,rj} = \left[\frac{mg}{2t}(l_f\cos\theta + h\sin\theta)\right] \pm \left[ma_y\left(\frac{h}{t}\right)\left(\frac{l_r}{l}\right)\right] + \left[\frac{ma_x}{2}\left(\frac{h}{l}\right)\right] \quad (6)$$

Meanwhile, lateral slip angle, α and longitudinal slips for the vehicle, σ are determined by Eqs. (7) and (8), respectively. All notations are based on Fig. 1 and 2, $j = left/right$ and for non-steerable wheels, $\delta_{rl} = \delta_{rr} = 0$. For longitudinal slip, ω_{ij} is the rotational speed of each tyre, respectively, and R_{ij} is the tyre radius. Detail model and derivations can be found in [30, 35, 36].

$$\text{Lateral slips, } \alpha_{fj} = \tan^{-1}\left[\frac{v_y + l_f\dot{\psi}}{v_x + \left(\frac{t}{2}\right)\dot{\psi}}\right] - \delta_{fj}$$
$$\alpha_{rj} = \tan^{-1}\left[\frac{v_y - l_r\dot{\psi}}{v_x + \left(\frac{t}{2}\right)\dot{\psi}}\right] - \delta_{rj} \quad (7)$$

$$\text{Longitudinal slips, } \sigma_{ij} = \frac{v_x - (\omega_{ij}\times R_{ij})}{\max(v_x, \omega_{ij}R_{ij})} \quad (8)$$

2.1.3 Vehicle kinematic model

This model is to determine the vehicle position with respect to the global coordinate axes, X–Y [35, 37] by considering the velocity component in local coordinate axes, x–y and instantaneous yaw rotation, ψ. Detail explanation of this model can be found in Amer et al. [38].

$$X = \int_0^{X_0}\left(v_x\cos\psi - v_y\sin\psi\right)dt$$
$$Y = \int_0^{Y_0}\left(v_x\sin\psi + v_y\cos\psi\right)dt \quad (9)$$

2.1.4 Validation of vehicle model

As stated before, model used in this study has been described in detail by Aparow et al. [30]. The vehicle responses have been validated by verifying the simulated results with CARSIM software. Several verification tests were carried out namely Step Steer test, Double Lane Change, and Slalom tests in various speeds to evaluate the model's validity. The vehicle model was proven to be valid and details on the results can be found in the publication.

2.2 Road courses and path development

In any control system development, one should consider the disturbance to the system and come up with a control strategy that will stabilize the system under these disturbances. In path tracking system, disturbances are in terms of variation in road courses and trajectories to be undertaken by the vehicle. In this study, several road courses were used to test the tracking control. Different from previous studies that include path planning module that focused on defining a specific path for the vehicle to follow based on a specified destination and task specific functions [39, 40], this study focuses on development of the path tracking control with pre-defined trajectories without relying on an advanced path planner. Each of the road courses were modeled as a set of points on a global coordinate axes, X and Y which in real life, may represent specific latitude and longitude or vice versa. Series of coordinate points are recorded along the road to represent the road courses. During maneuvering, real-time position of the vehicle on the global axes will be compared alongside these points and the subsequent lateral error and path states will be determined.

Figure 4 shows the six roads used to test the controllers in this study with each of them named according to the nature and shape of the road. They are (from left to right); Straight Road, Multiple Lane Change, Double Lane Change, Hook, S, And Curve. The first three courses on the top of the figure represent shorter roads with sharp turns and small curvature. Meanwhile, the lower three graphs show roads with larger curvature and further range. The Straight road used in this study was taken from the same path used in Amer et al. [9].

3 Path tracking control strategies

In this study, several variants of Stanley controller are developed, optimized and analyzed. Stanley steering controller was the controller on the Stanford University autonomous vehicle that won the DARPA Challenge in 2005 [17, 41]. Two variants of the Stanley controller, namely Stanley with yaw compensation (St-Yaw) and Modified Stanley (Mod-St) are studied by modifying the original Stanley controller.

3.1 Stanley controller (St)

In the original work, the controller consists of a steering command which will determine correctional steering input of wheel angle, δ, in guiding the vehicle to follow the desired path. The original steering command is shown in

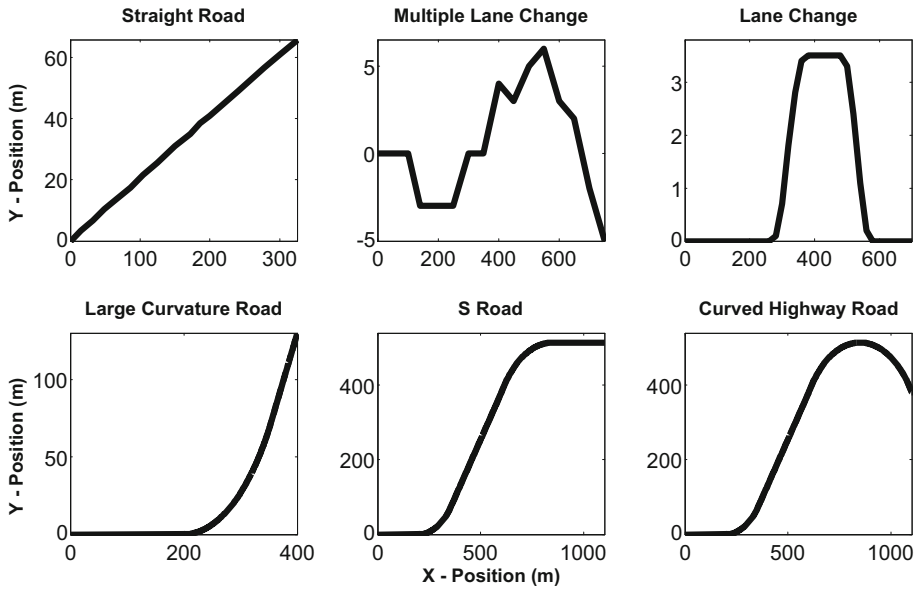

Fig. 4 Road courses for controller testing

Eq. (10). It contains two terms depending on several errors between the vehicle's and path's states as shown in Fig. 5, which are: heading error between the vehicle direction of motion and path direction, ϕ, where $\phi = \psi - \psi_{traj}$; and lateral error, e, measured from the centre of steering wheel axle to the nearest point on path, where v is the vehicle speed, and k is a tuneable gain, associated with the second term.

The full development and stability analysis were presented by Hoffmann et al. [17]. For this controller, Snider [3] has demonstrated the tuning and concluded that $k = 10$ is the best value for the controller. Current vehicle's position is denoted by the X and Y position of the vehicle on the global axes. This will be the reference to get the nearest point on path which is used to determine the instantaneous path yaw rate and direction. These will be compared against vehicle's states to calculate the error for the controller.

$$\delta = \phi + \tan^{-1}\left(\frac{ke(t)}{v(t)}\right) \tag{10}$$

3.2 Stanley controller with yaw compensation (St-yaw)

An extended version of the controller was proposed by Hoffmann et al. [17] with an additional yaw term to compensate the vehicle and path yaw rates which act as dampers, providing reaction forces to sideways motions. The steering command now consists of one additional term, the error between instantaneous path and vehicle's yaw rate, $\left(\dot{\psi} - \dot{\psi}_{traj}\right)$, which associated with a tuneable gain, k_ψ. In this study, another tuneable gain, k_ϕ, is added to the first term. Full steering command for the Stanley Controller with yaw compensation is shown in Eq. (11) with symbols and variables are as defined in Fig. 5 and nomenclature.

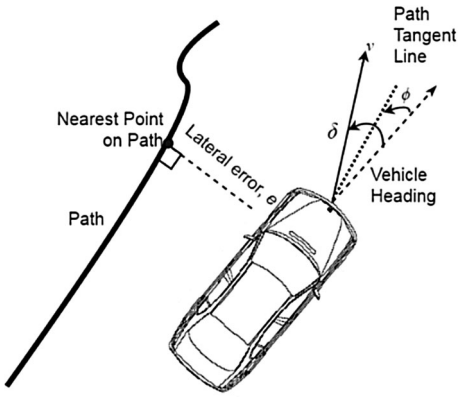

Fig. 5 Parameters for Stanley controller

$$\delta(t) = k_\phi \phi + \arctan\left(\frac{ke(t)}{1 + v(t)}\right) + k_\psi\left(\dot{\psi} - \dot{\psi}_{traj}\right) \tag{11}$$

3.3 Modified Stanley controller (Mod St)

This is the proposed controller for the armoured vehicle. The full controller formula from Hoffmann et al. [17] and Eq. (11) is modified by adding another tuneable gain, k_1, associated with the arctan function in second term to provide more sensitivity to the term in tuning the controller. Therefore, the full steering command for this study now contains four tuneable parameters for its three terms, as presented in Eq. (12). Full configuration of the control structure and state feedbacks required are shown in Fig. 6.

$$\delta(t) = k_\phi \phi + k_1 \arctan\left(\frac{ke(t)}{1 + v(t)}\right) + k_\psi\left(\dot{\psi} - \dot{\psi}_{traj}\right) \tag{12}$$

Each term in Eq. (12) plays a significant role in determining the automatic steering input. The first term with heading error, ϕ, will observe the vehicle's direction with respect to path's direction. The second term with lateral error, e, will observe the vehicle's position with respect to the path. These two terms will stabilize the direction of the vehicle and guide vehicle towards intended path. In large errors, the vehicle will be steered with maximum allowable steering angle until the lateral error diminishes and the vehicle is facing the same direction with path. While the first two terms will guide the vehicle to its intended path, the third term with the yaw rate error stabilizes the yaw response of the vehicle. Negative feedback on yaw rate will dampen the unwanted yaw motions due to rapid steering actions demanded by the first two terms [17].

For stability, the steering command should be saturated at $\pm\ \delta_{\max}$. Preliminary study in Hoffmann et al. [17] have evaluated the stability of the controller within operating region $-\frac{\pi}{2} \le \delta \le \frac{\pi}{2}$. However, in this study, due to the limitation of the steering system, the operating region for the wheel angle was limited to 10 degrees which equivalent to $-\frac{\pi}{18} \le \delta \le \frac{\pi}{18}$.

4 Optimization of controller parameters using particle swarm optimization

In this study, the optimization procedure is aimed to find a set of controller parameters which will produce the best controller action and optimum path tracking performance, as suggested by Amer et al. [9], Khairuddin et al. [21],Jaafar et al. [22]. Position of each particle corresponds to the number of variables to be optimized, e.g., in this case, (St-Yaw), 3 (k_ϕ, k, and k_ψ), and (Mod St), 4 (k_ϕ, k_1, k, and k_ψ). To evaluate the agent's fitness on any given

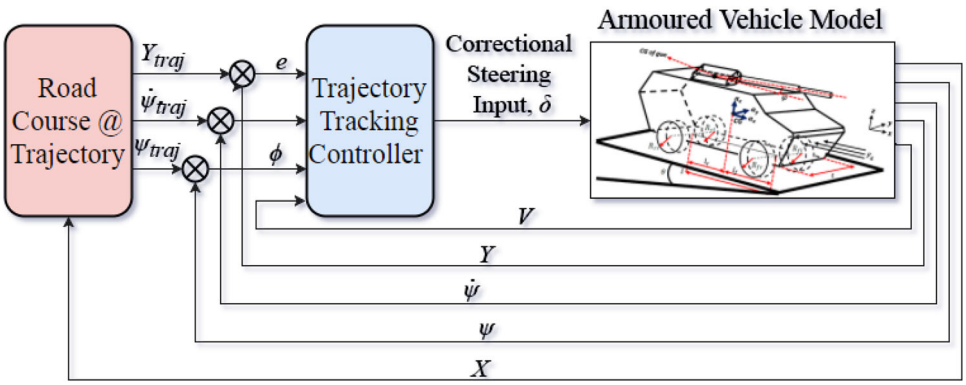

Fig. 6 Structure for Modified Stanley path tracking control

position, fitness function was chosen to be the controller performance. The PSO algorithm will vary each of the controller parameters independently until the optimum performance, i.e., minimum fitness value, is found. To facilitate faster convergence, fitness function was chosen carefully as well as the choice of parameters search space. In this study, average lateral error throughout the maneuvering was chosen since it is the best indicator for the path tracking performance. Thus, fitness function for the control system was chosen to be the root mean square (RMS) value of lateral error, e between vehicle's and path lateral positions shown in Eq. (13). Figure 7 shows the overall procedure to optimize Stanley Controllers using PSO.

$$\text{Fitness Function, } f(k_f, k_1, k, k_y) = \sqrt{\frac{\sum [e(t)]^2}{n}} \tag{13}$$

Selection of PSO parameters itself poses a well-known field of study which has sparked many researches specifically in choosing the right parameter values [42–45]. The main three parameters in the PSO algorithm are Social coefficient, s, Cognitive coefficient, c, and inertial weight, iw. In this paper, the same PSO configuration and parameter values as proposed by previous researchers [21, 22, 26] are used as shown in Table 1. No. of dimensions, N_d was chosen based on the number of variables to be optimized. Also, search space for each parameter was defined to be as small as possible by the lower and upper limits. In this study, the limits for each parameter were found after series of sensitivity analysis (with the same procedure from Amer et al. [9]) which indicate the controller's response under different parameter values. Apart from ensuring faster convergence, these limits are important in order to avoid the controller from entering unstable region which will terminate the overall optimization procedure. Lastly, no. of particles, N_p and iterations, N_i are chosen by try and error method by observing the number of iterations it takes for the solution to converge. In this case, convergence always occurs within ten iterations. Therefore, the number of iterations was set to be 100% more than this to allow convergence even if there is any divergence from solution.

Upon carrying out the optimization process, a set of values for k_ϕ, k_1, k, and k_ψ for the proposed controller are determined for each of the road courses. This translates to 6 sets of parameters corresponding to all 6 road course trajectories. Optimized parameters for Stanley with yaw compensation (St-Yaw) and modified Stanley (Mod St) controllers are shown in Table 2, respectively. With these parameters, simulations are carried out for each road course.

5 Results and discussions

Simulations were carried out within MATLAB/SIMULINK environment using Heun ODE2 solver with fixed 0.001 s step size. The effectiveness of the proposed controller from Eq. (12) was evaluated on the validated armoured vehicle model as explained in Sect. 2 with constant speed of 6 m/s, where the vehicle is assumed to enter the path with the initial 6 m/s speed and kept at the constant speed with zero throttle, brake setting, and firing force. The controller performance was compared against original Stanley controller (St) as proposed by Hoffmann et al. [17], and shown in Eq. (10) and the Stanley controller with yaw compensator (St-Yaw) from Eq. (11) which were simulated using the same simulation parameters. All the controllers' gains were optimized earlier with PSO algorithm using similar procedures as in Sect. 4.

In evaluating the effectiveness of the controller, few responses were chosen, namely the Y-Position of vehicle with respect to path to show the tracking performance; cross-track lateral error, e to quantify tracking performance; the steering angle to denote the controller's effort; and yaw rate responses to observe the maneuvering effect throughout driving. The respective dynamic responses corresponding to each path for each controller are shown in Figs. 8, 9, 10, 11, 12 and 13.

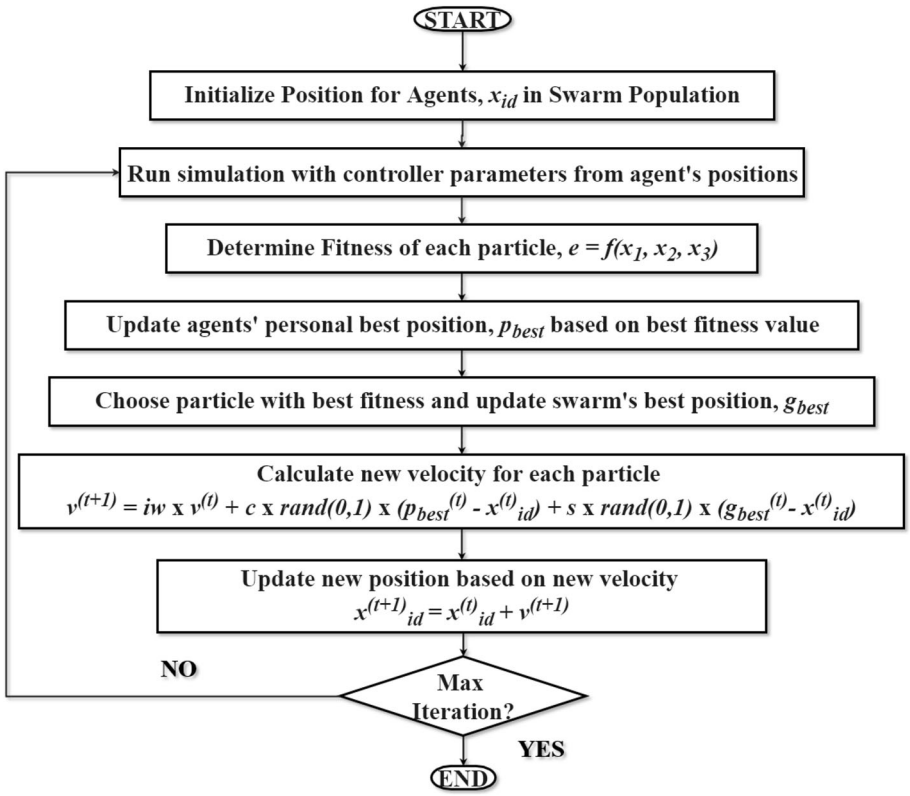

Fig. 7 Procedure for PSO optimization

Table 1 PSO Parameters used for Stanley controller variants

Parameter	St-Yaw	Mod-St
Social coefficient, s	1.42	1.42
Cognitive coefficient, c	1.42	1.42
Inertial weight, iw	0.9	0.9
No. of dimensions, N_d	3 (k_ϕ, k, and k_ψ)	4 (k_1, k_ϕ, k, and k_ψ)
Upper bound limit	[10;10;10]	[10;10;10;10]
Lower bound limit	[− 10; − 10; − 10]	[− 10; − 10; − 10; − 10]
No. of particles, N_p	150	150
No. of iterations, N_i	20	20

Table 2 Optimised parameter values for St-Yaw and Mod-St controllers

	St-Yaw			*Mod-St*			
	k	k_ψ	k_ϕ	k_1	k	k_ψ	k_ϕ
Straight	10	− 0.0242	0.4495	10	10	− 2.964	0.7719
Multiple lane change	10	0.3577	0.6732	10	10	0.3046	1.892
Double lane change	10	0.2987	0.4215	10	9.689	0.0901	0.819
Hook	10	0.2299	0.0221	10	10	0.0423	− 0.058
S	10	− 0.0242	0.4495	10	9.757	0.0642	0.0199
Curve	9.4343	0.2634	− 0.0158	10	10	0.1762	− 0.0051

The results are organized as follows. There are six trajectories each with four results. Each of the six figures corresponds to each path from Fig. 4 and contains four graphs in the figure which corresponds to each result namely tracking performance, lateral error, steering angle, and vehicle's yaw rate, respectively. All graphs are using the same line representations where the original Stanley, St is denoted by the dashed lines, solid grey lines for the St-

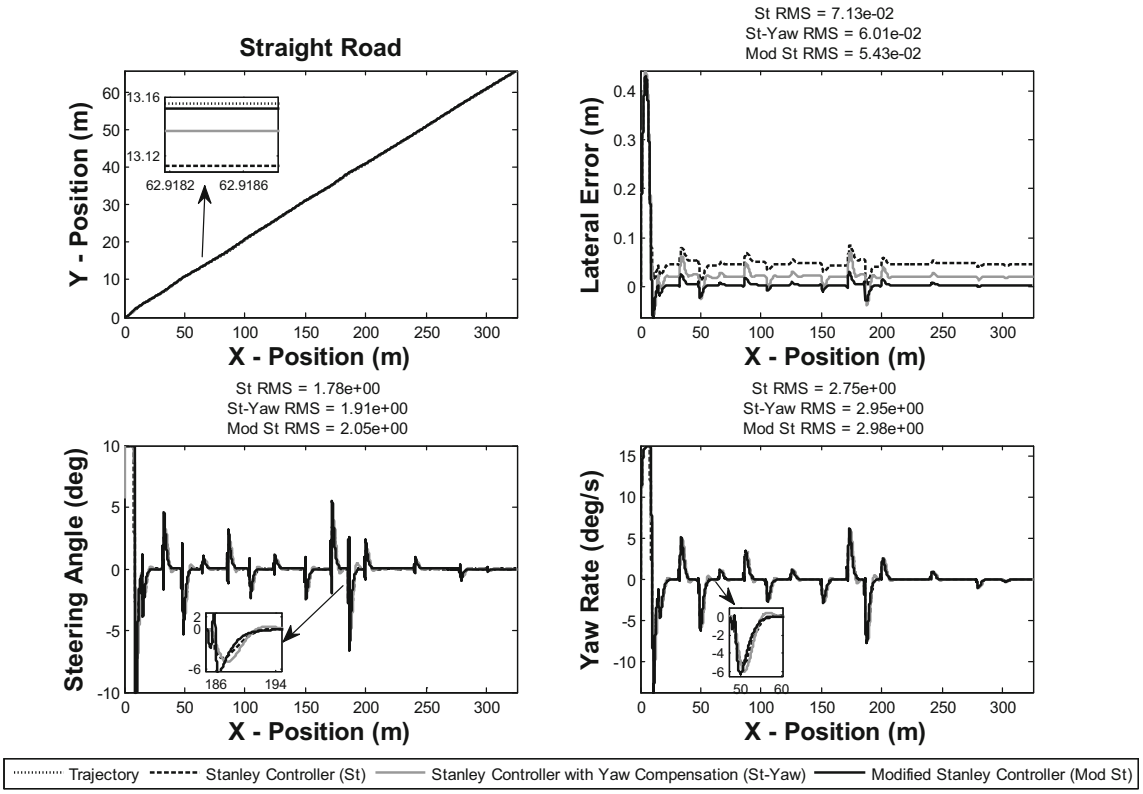

Fig. 8 Performance comparison between Stanley controllers for straight road

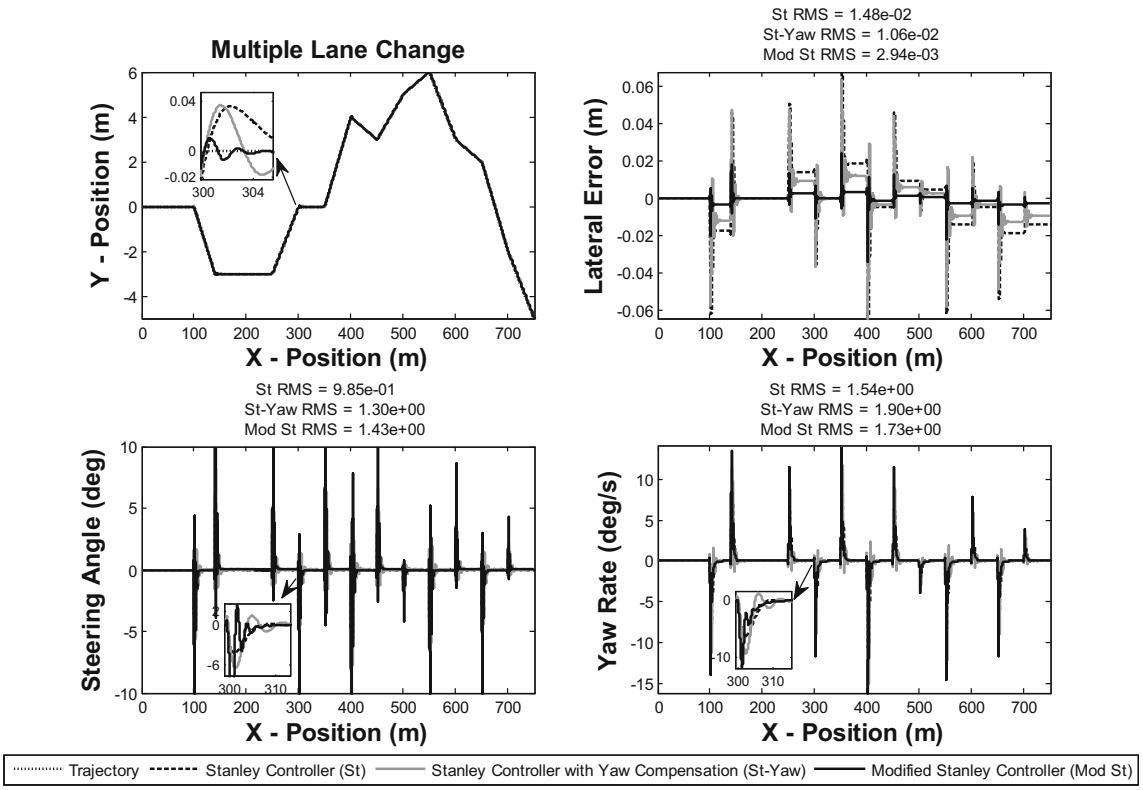

Fig. 9 Performance comparison between Stanley controllers for multiple lane change maneuvering

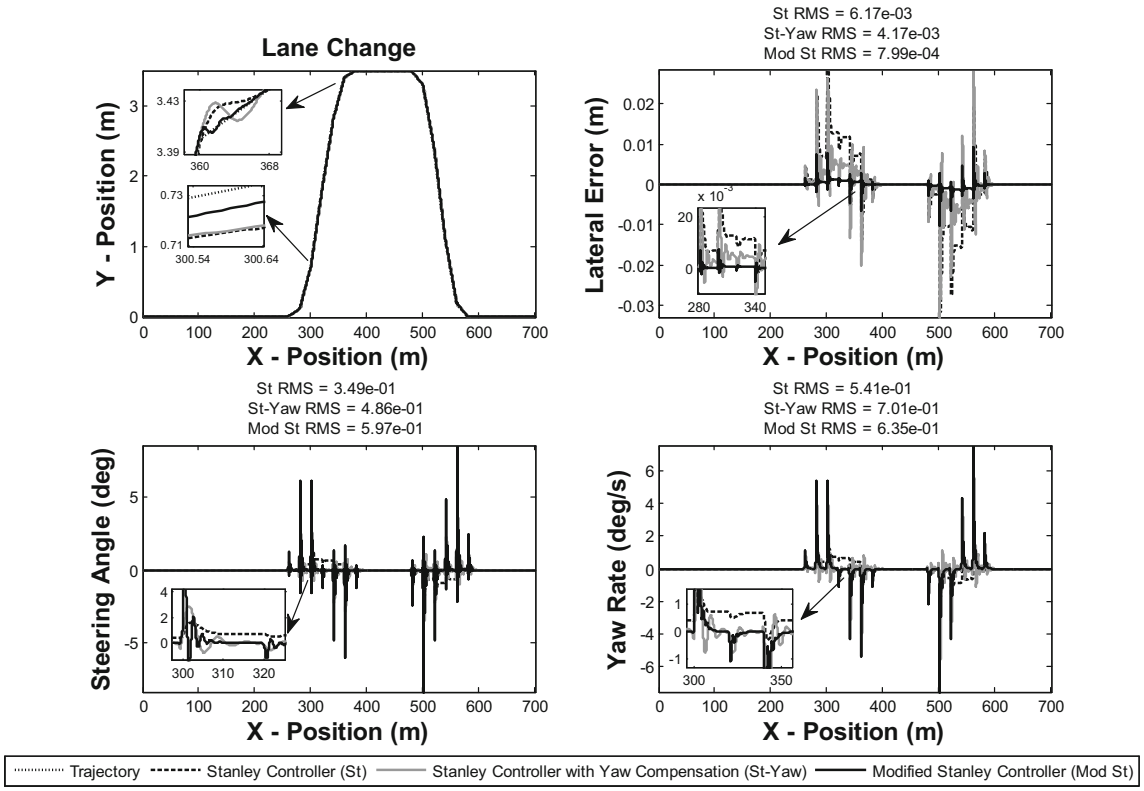

Fig. 10 Performance comparison between Stanley controllers for double lane change manoeuvring

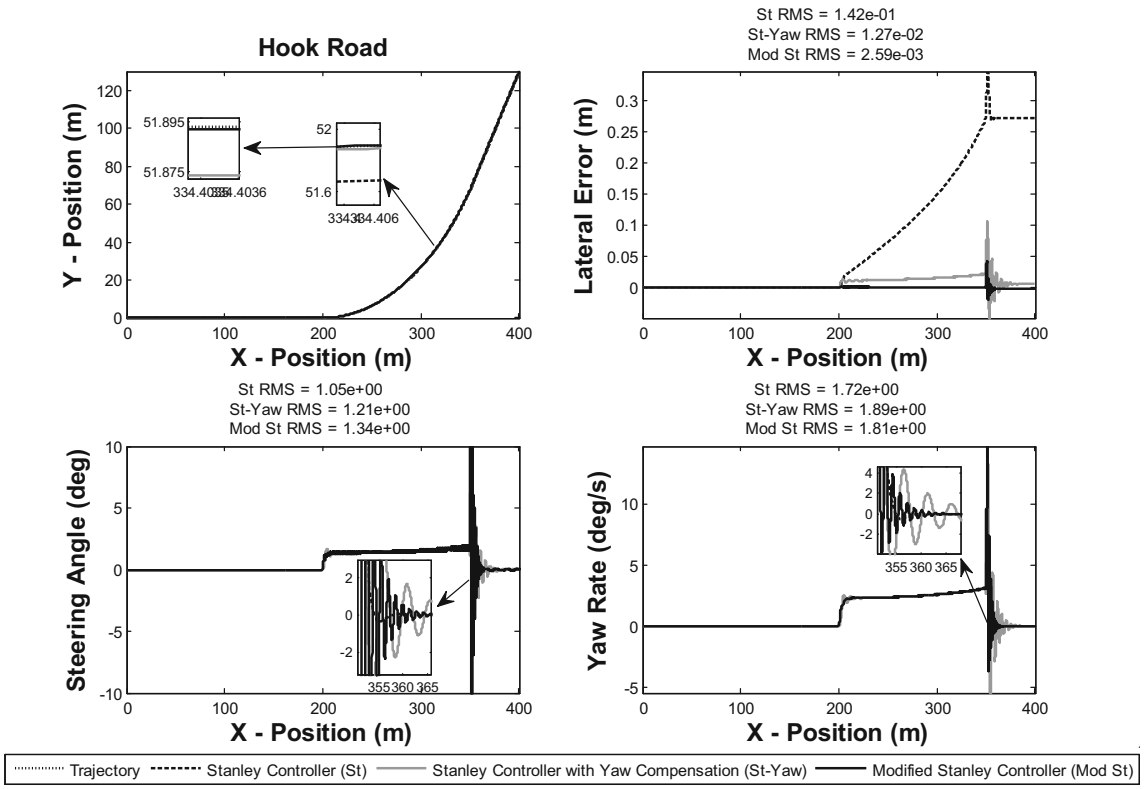

Fig. 11 Performance comparison between Stanley controllers for hook road

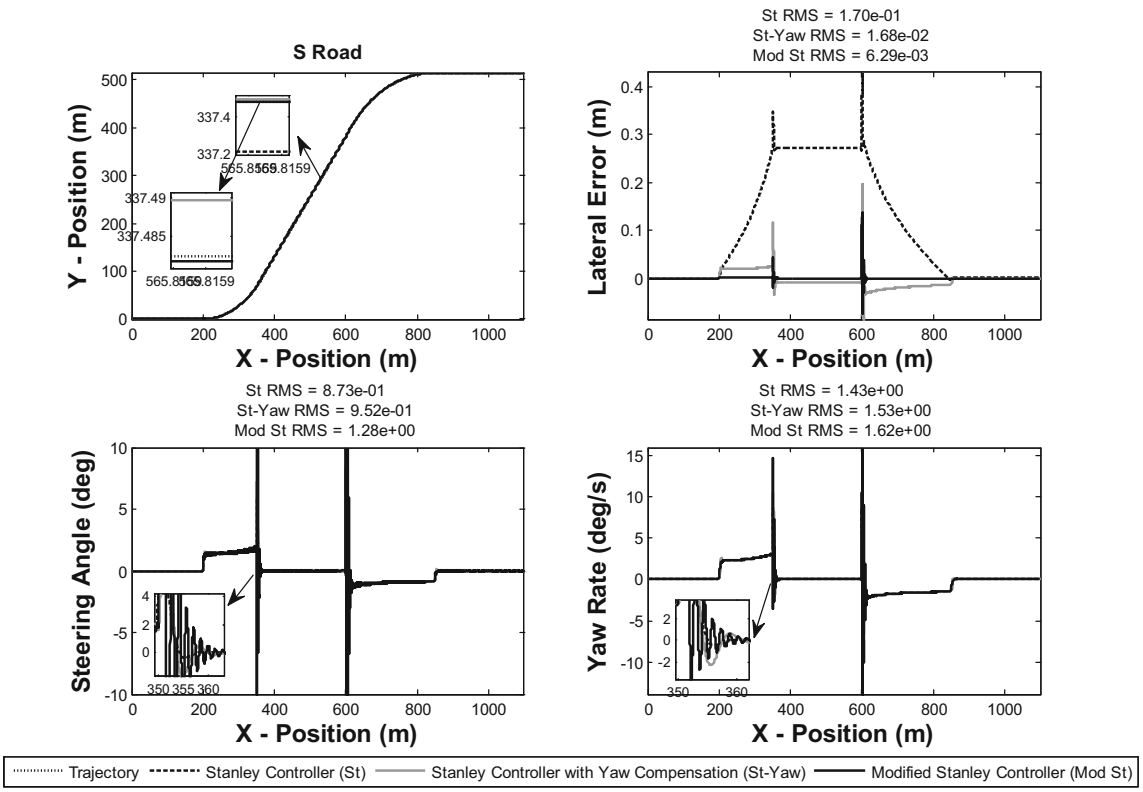

Fig. 12 Performance comparison between Stanley controllers for S road

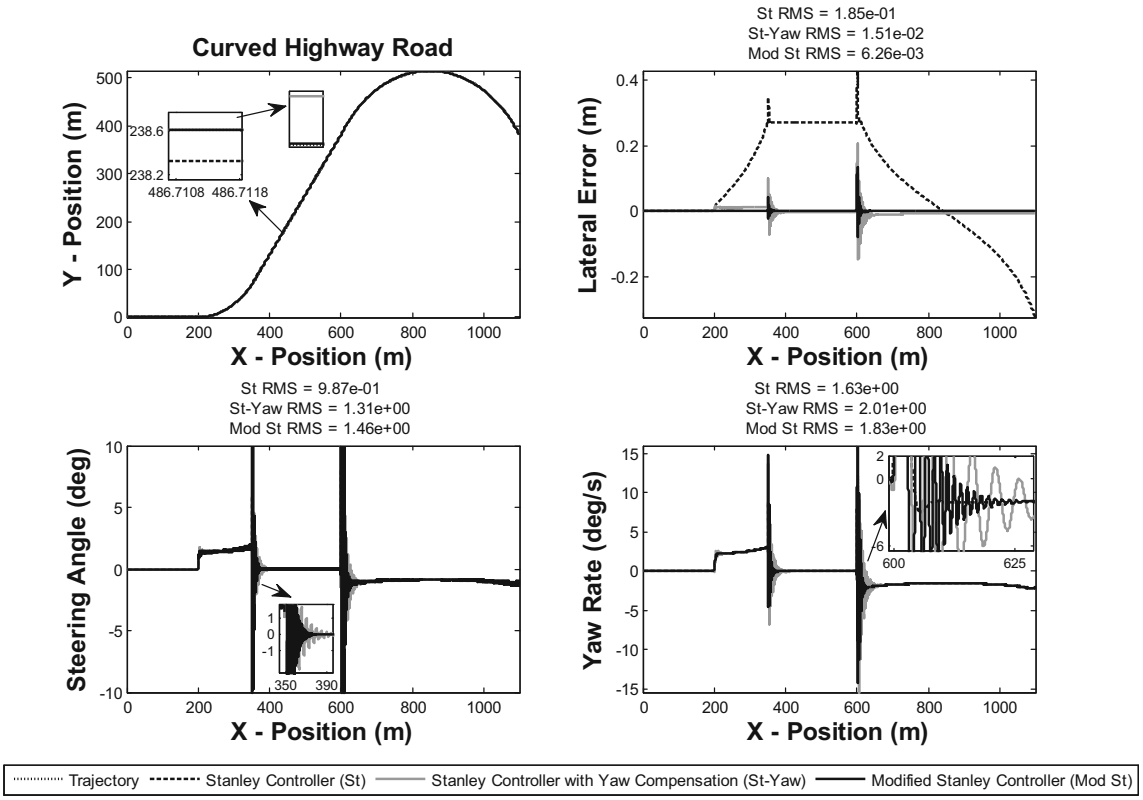

Fig. 13 Performance comparison between Stanley controllers for curved highway road

Yaw controller, and solid black lines for the proposed Mod-St controller. RMS values of the data are used to quantify these results for comparison purposes between the two controllers.

Figure 8 shows the vehicle's performance with all three variants for Stanley Controllers on the straight road. The Y-Position graphs denote the path tracking performance which proved that the three controllers managed to guide the armoured vehicle to follow the desired path. Upon zooming the graph, one can see the better performance by Mod-St controller which managed to steer the vehicle closer to the path. This can be further proven by looking at the second graph showing lateral error results. Modified Stanley controller achieved better lateral error compared to its predecessors most of the time which can be seen from the transient graph and the RMS value with 9.7% improvement compared to the St-Yaw controller.

The excellent performance in lateral error was caused by the increase in controller's effort which can be seen in the steering angle graph. An increase of 7.3% was recorded by the proposed controller compared to its predecessor, St-Yaw. This caused an evident increase in yaw rate values which can be seen in the yaw rate graph with 1% increase of RMS values.

Figure 9 shows the results for multiple lane change maneuver. Comparing the three Stanley controllers for this road yields same improvement in terms of path tracking performance. The proposed controller, modified Stanley managed to guide the vehicle closer to the path at all times. This is proven by the lateral error data which shows less lateral error at all times for the Mod St controller with less oscillations, showing more stable behavior during path tracking. In average, modified Stanley controller achieved about 0.003 *m* lateral error which is 72.3 and 80% less than the St-Yaw and St Controllers, respectively.

It is easier to understand that the increase in lateral error is caused by better steering actions throughout maneuvering. From the steering angle data, the proposed controller was capable to provide faster and larger steering input with RMS of 10% more than its predecessors. In terms of yaw rate, less overall readings were recorded by the vehicle with proposed controller with 1.9 deg/s which is 8.9% less than yaw rate response from vehicle with Stanley controller. This can be attributed to the vehicle's oscillation in following path from tracking graph in the top-left graph.

In Fig. 10, results for double lane change maneuver are shown similarly. From left to right, first graph showed the tracking performance from Y-Position between path and vehicle, followed by lateral error, steering angle in bottom left and lastly, vehicle's yaw rate response. In the first graph, all Stanley controllers yield good tracking performance with the modified Stanley managed to guide the vehicle closer to the path in the close-up result. This is proven by the lateral error data which shows less lateral error at all times for the modified Stanley controller. In average, modified Stanley controller achieved about 0.004 *m* lateral error RMS which is 80.8% less than the St-yaw controller and 87% less than the original Stanley controller. Also, from the close-up result, less oscillations and overshoot were recorded for the proposed controller where the vehicle tracked the path with more stable behavior.

The improvement in lateral error is caused by better steering efforts throughout maneuvering. From the steering angle data, the proposed controller generated faster and larger steering input with RMS of 0.06°, 22.8% more than St-yaw's result. Despite saturated at only 10° wheel angle, the vehicle managed to follow the desired path excellently. In terms of yaw rate, the vehicle with proposed controller recorded 0.064 deg/s yaw rate RMS which is 9.4% less than St-Yaw's controller and 17% more RMS compared to original St controller. This can be attributed to the vehicle's oscillation in following path from tracking graph in the top-left graph. High yaw rate was experienced by the vehicle at cornering due to faster and larger steering angle. Overall, the Mod St controller managed to significantly improve the lateral error (87%) with a small increase in yaw rate response (17%) compared to the original Stanley controller.

Figure 11 shows the results for the Hook road with the same template and arrangement as previous results. Path tracking performance in the first graph showed excellent tracking performance by all controllers with no visible error present. Lateral error data in second graph clarified this result and showed that the modified Stanley controller performed significantly better with smaller lateral error at all times. Overall, the proposed controller recorded 0.0026 m RMS value, 79.6% less than its predecessor, St-Yaw and a 98% less compared to the original St.

The controllers' effort from steering angle data showed the steered angle of vehicle's wheel throughout maneuvering. From the close-up result, it can be seen that the modified Stanley controller managed to provide faster steering actions compared to the unmodified Stanley controller. In terms of overall RMS, the proposed controller recorded 1.34° steering angle which is 10.7 and 27% more than the St-Yaw and original St controllers, respectively. This is the contributing factor that the proposed controller shown better performance in path tracking.

The last graph on bottom-right showed the vehicle's yaw rate data which displayed oscillating yaw rate experienced by the vehicle with modified Stanley. St-Yaw controller, on the other hand showed that larger yaw rate experienced by the vehicle. However, the RMS was determined to be nearly the same with 1.72, 1.89 and 1.81 deg/s for St, St-Yaw and Mod St, respectively. This

translates into 4.2% decrease and 5.2% increase in yaw rate response by the Mod St controller compared to St-Yaw and original St, respectively. Therefore, the proposed controller managed to improve the lateral error and path tracking performance by 98% compared to only 5% increase in yaw rate responses.

In Fig. 12, simulation data for S road is presented. Similarly, tracking performance in the first graph showed excellent performance by all controllers with the modified Stanley controller performed better in steering the vehicle towards desired path. This can be further clarified by observing the lateral error data in the next graph. The modified controller logged less lateral error at all times throughout the maneuvering with overall RMS value of 0.0063 m, 62.6% less than the 0.017 m recorded by the St-Yaw controller and 96% less than the original St controller.

The excellent path tracking performance is mainly attributed to the faster and higher steering action provided by the modified Stanley controller. In terms of RMS value, the modified controller provided 1.28° of steered wheel angle compared to 0.952° from the St-Yaw controller and 0.873° from the original Stanley controller. This is 35% and 47% increase of RMS wheel angle values, respectively. However, the rapid steering actions caused an increase in the overall yaw rate response experienced by the vehicle. 1.62 deg/s of yaw rate was recorded by the vehicle with modified Stanley controller which was 5.9% and 13.3% more than that recorded by St-Yaw and St controllers, respectively. Therefore, it can be said that the proposed controller managed to record significant improvement in lateral error of 96% to the expense of an increase in yaw rate response of only 13%.

Figure 13 shows the results for the final simulation test with the curved highway road. The results exhibited similar pattern to other roads with good performance from all controllers in terms of path tracking. In terms of lateral error, the proposed controller recorded the RMS cross-track error of 0.063 m, which is 58.5 and 97% less compared to St-yaw and St controllers, respectively. Similarly, larger wheel angles were recorded throughout the maneuvering with noticeable faster controller action. 1.46° of wheel angle RMS were recorded by the modified Stanley controller which are 11.5% higher compared to the unmodified Stanley controller. In terms of vehicle yaw rate, the proposed controller managed to guide the vehicle with 9% less yaw rate RMS value compared to St-yaw's with 2.01 deg/s.

Looking at the overall performance by all three controllers for all trajectories, one can conclude that the three approaches yield good results in following desired path for all roads. However, the original Stanley controller recorded significantly high cross track errors, especially in larger longer road courses where the lateral error was seen to achieve as high as 0.4 m. The modified controllers (St-Yaw and Mod St), on the other hand, managed to keep the lateral errors well within 0.1 m. This is due to the fact that both controllers were used with properly tuned parameters. Looking at the lateral errors data, significant distinction can be seen between both controllers. The modified Stanley controller revealed lower cross track errors for all trajectories with improvements ranging from 9.7 to 80.8% compared to the St-Yaw controller.

Evidently, these improvements were caused by better effort from the controller. This can be seen from the steering angle graphs which presented the correctional wheel angle input from controllers. 7.3–34.5% increase in controller's effort was recorded for all trajectories. This increase, however, was smaller than the recorded improvement in lateral error earlier. The proposed modified controller also produced better maneuvering with lower yaw rate responses compared to its counterpart. All trajectories were navigated by the proposed controller with lower yaw rate responses with improvements range 4.2–9.4% except for straight and S roads with slightly higher responses of 1 and 5.9%, respectively. This was caused by the faster steering actions provided by the proposed controller.

From presented results, the proposed controller namely modified Stanley (Mod St) controller was proven to perform better compared to the basic Stanley controller with noticeable improvement in lateral error along desired trajectory. One might be curious on how the proposed controller would fare against other geometric controllers available. Snider [3] has demonstrated extensive works on six different trajectory tracking controllers namely Pure Pursuit, Stanley, kinematic LQR, and LQR with road preview, which were tested on three different road courses. The study proven that among all the controllers, Stanley displayed the best tracking performance despite its simple configuration. The study uses a standard passenger's vehicle available in CARSIM software, while this paper focuses on armoured vehicle implementation. It should be noted that the armoured vehicle considered in this study is a 4-wheel lightweight armoured vehicle with slightly different parameters. However, the basic dynamics are still the same and therefore, same observations can be expected where Stanley can outperform the other controllers. Furthermore, the proposed Mod St controller was further enhanced from the basic Stanley with improved sensitivity and optimization with PSO. Therefore, the proposed controller is predicted to perform better than other basic controllers available.

Another benefit for the modified Stanley controller can be realized by observing the convergence behavior during optimization process. With more tuneable parameters compared to its predecessor, the whole controller command

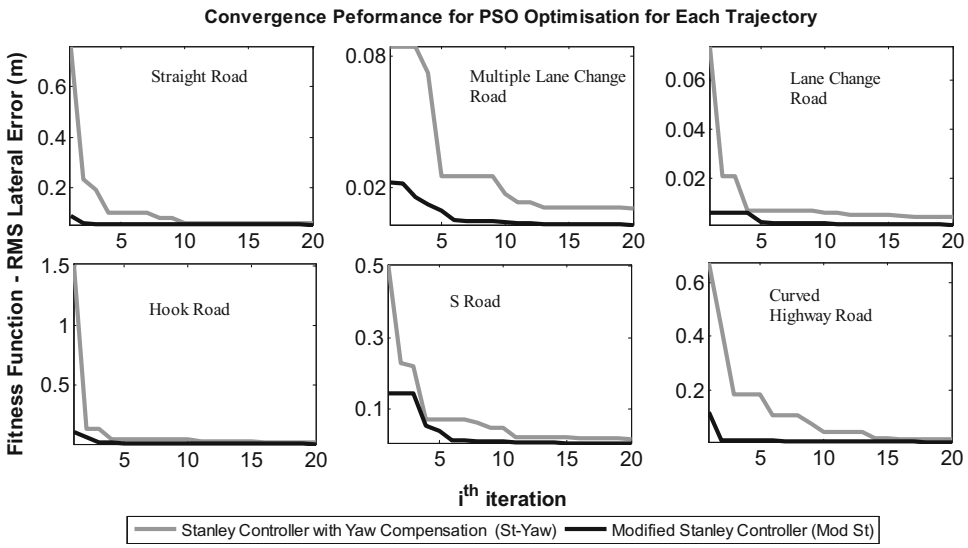

Fig. 14 Optimisation convergence performance for each path

was made more sensitive to tuning which result in faster convergence to optimum solution. Optimization convergence performance between St-Yaw and Mod-St controller for each path is shown in Fig. 14. It is apparent from these results that the optimization procedure managed to obtain better solution with each iteration. Also, it is noticeable that the modified controller needed less iterations to find the optimum solutions compared to its predecessor.

Looking properly at the results in Fig. 14, the modified controller was optimized very well even on the first iteration with 72–93% better solutions compared to the optimized Stanley controller. Also, the overall convergence was achieved within 10 iterations with better results of up to 80% better fitness value of the solutions. Optimization for the straight path shown in the first graph recorded the least improvement in final solution with 9% improvement only. However, it can be seen that the optimum solution was achieved within only 2 iterations. These results are indicating good optimization performance after modifying the Stanley controller.

6 Conclusions

In this paper, a path tracking controller for an armoured vehicle is proposed by modifying an existing geometric controller named Stanley Controller. Three variants of Stanley controller were compared namely St, St-Yaw, and Mod St Controllers. Both St-Yaw and Mod St were optimized using Particle Swarm Optimization. The three controllers were implemented on a validated 7DOF armoured vehicle model that includes a nonlinear tire model, longitudinal and lateral slips estimator, drivetrain model, load distribution model, and kinematic model. Then, the controllers were evaluated on six different trajectories considering large curvature roads as well as roads with sharp maneuverings.

In general, both modified controllers performed well in guiding the vehicle to follow the intended path. However, the proposed controller, Mod St was proven to improve the vehicle's responses significantly. Three dynamic performances were compared between the controllers, namely the lateral cross-track error, correctional steering input, and yaw rate responses. Overall, the proposed controller shown significant improvement of up to 80, 35, and 9.4% for each responses, respectively, compared with its predecessor, the St-Yaw controller. The controller competency compared to St-Yaw controller was also assessed in terms of its effectiveness in optimization processes. With the same optimization procedure, convergence for the proposed controller was achieved within 10 iterations for all trajectories which were better than St-yaw. Also, the converged solutions were better for the proposed controller as well, with solution fitness values of up to 93% less than that of St-yaw controller.

In terms of recommendation for future works, it is obvious that the performance of this controller on any given path depends solely on how well the parameter-tuning process was. While the proposed controller may offer a better optimization performance, the controller still needs to be tuned properly to achieve desirable results. This is a known issue where the same conclusion can be made for all geometric based controllers [3, 17, 46]. Therefore, development of a new adaptive method to determine these parameters can be a good research prospect.

Acknowledgements The authors would like to thank the Malaysian Ministry of Education for their financial supports and technical advises for this research through research grant RAGS (RAGS/1/2014/TK01/UPNM/1).

References

1. Sathiyanarayanan M, Azharuddin S, Kumar S (2014) Four different modes to control unmanned ground vehicle for military purpose. Int J Eng Dev Res 2(3):3156–3166
2. Zakaria MA, Zamzuri H, Mazlan SA, Zainal SMHF (2012) Vehicle path tracking using future prediction steering control. Procedia Eng 41:473–479. https://doi.org/10.1016/j.proeng.2012.07.200
3. Snider JM (2009) Automatic steering methods for autonomous automobile path tracking. Robotics Institute, Carnegie Mellon University, Pittsburgh
4. Rossetter EJ (2003) A potential field framework for active vehicle lanekeeping assistance. PhD Dissertation, Stanford University, Stanford
5. Fierro R, Lewis FL (1995) Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In: Decision and Control, Proceedings of the 34th IEEE Conference on, 13–15 Dec 1995, pp 3805–3810 vol. 3804. https://doi.org/10.1109/cdc.1995.479190
6. Aithal H, Janardhanan S (2013) Trajectory tracking of two wheeled mobile robot using higher order sliding mode control. In: International Conference on Control Computing Communication & Materials (ICCCCM), Allahabad, India, Aug 3–4 2013. IEEE, pp 1–4. https://doi.org/10.1109/icccm.2013.6648908
7. Imine H, Madani T (2013) Sliding-mode control for automated lane guidance of heavy vehicle. Int J Robust Nonlinear Control 23(1):67–76
8. Park M, Lee S, Han W (2015) Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm. ETRI J 37(3):617–625. https://doi.org/10.4218/etrij.15.0114.0123
9. Amer NH, Zamzuri H, Hudha K, Aparow VR, Kadir ZA, Abidin AFZ (2016) Modelling and trajectory following of an armoured vehicle. In: 2016 SICE International Symposium on Control Systems (ISCS), Nagoya, Japan, March 7–16, 2016. IEEE, pp 1–6
10. Martins FN, Celeste WC, Carelli R, Sarcinelli-Filho M, Bastos-Filho TF (2008) An adaptive dynamic controller for autonomous mobile robot trajectory tracking. Control Eng Pract 16(11):1354–1363
11. Merabti H, Belarbi K, Bouchemal B (2016) Nonlinear predictive control of a mobile robot: a solution using metaheuristcs. J Chin Inst Eng 39(3):282–290. https://doi.org/10.1080/02533839.2015.1091276
12. Coulter RC (1990) Implementation of the pure pursuit path tracking algorithm. Robotics Institute, Carnegie Mellon University, Pittsburgh
13. Amidi O (1990) Integrated mobile robot control. Robotics Institute, Carnegie Mellon University, Pittsburgh
14. Kim D-H, Han C-S, Lee JY (2012) Sensor-based motion planning for path tracking and obstacle avoidance of robotic vehicles with nonholonomic constraints. Proc Inst Mech Eng C. https://doi.org/10.1177/0954406212446900
15. Shan Y, Yang W, Chen C, Zhou J, Zheng L, Li B (2015) CF-Pursuit: a pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. Int J Adv Rob Syst 12(134):1–13
16. Zakaria MA, Zamzuri H, Mazlan SA (2016) Dynamic curvature steering control for autonomous vehicle: performance analysis. IOP Conf Ser 114(1):012149
17. Hoffmann GM, Tomlin CJ, Montemerlo D, Thrun S (2007) Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing. In: American Control Conference, 2007. ACC '07, 9–13 July 2007, pp 2296–2301. https://doi.org/10.1109/acc.2007.4282788
18. Töro O, Bécsi T, Aradi S (2016) Design of lane keeping algorithm of autonomous vehicle. Period Polytech Transp Eng 44(1):60–68. https://doi.org/10.3311/PPtr.8177
19. Ab Wahab MN, Nefti-Meziani S, Atyabi A (2015) A comprehensive review of swarm optimization algorithms. PLoS ONE 10(5):e0122827. https://doi.org/10.1371/journal.pone.0122827
20. Zhang K (2008) A minimum zone method for evaluating straightness errors using pso algorithm. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues, pp 308–314
21. Khairuddin IM, Dahalan A, Sarayati A, Abidin AFZ, Lai YY, Nordin NA, Sulaiman SF, Jaafar HI, Mohamad SH, Amer NH (2014) Modeling and simulation of swarm intelligence algorithms for parameters tuning of PID controller in industrial couple tank system. Adv Mater Res 903:321–326
22. Jaafar HI, Mohamed Z, Abidin AFZ, Ghani Z (2012) PSO-tuned PID controller for a nonlinear gantry crane system. In: IEEE International Conference on Control System, Computing and Engineering (ICCSCE), IEEE, pp 515–519
23. Vinodh Kumar E, Raaja GS, Jerome J (2016) Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter. Appl Soft Comput 41:77–90. https://doi.org/10.1016/j.asoc.2015.12.023
24. Eberhart RC, Kennedy J A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, 1995. New York, NY, pp 39–43
25. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on, IEEE, pp 69–73
26. Veeramachaneni K, Peram T, Mohan C, Osadciw L (2003) Optimization using particle swarms with near neighbor interactions. In: Cantú-Paz E, Foster J, Deb K et al (eds) Genetic and evolutionary computation, vol 2723. Lecture Notes in Computer Science. Springer, Berlin Heidelberg, pp 110–121. https://doi.org/10.1007/3-540-45105-6_10
27. Girbés V, Armesto L, Tornero J, Solanes JE (2011) Smooth kinematic controller vs pure-pursuit for non-holonomic vehicles. In: Conference Towards Autonomous Robotic Systems, 2011. Springer, pp 277–288
28. Yang J-M, Kim J-H (1999) Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. IEEE Trans Robot Autom 15(3):578–587
29. Xin M, Minor MA (2013) Variable structure backstepping control via a hierarchical manifold set for graceful ground vehicle path following. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, pp 2826–2832
30. Aparow VR, Hudha K, Ahmad MMHM, Jamaluddin H (2016) Development and verification of 9-DOF armored vehicle model in the lateral and longitudinal directions. J Teknol 78(6):117–137
31. Kadir ZA, Mazlan SA, Zamzuri H, Hudha K, Amer NH (2015) Adaptive fuzzy-PI control for active front steering system of armoured vehicles: outer loop control design for firing on the move system. Stroj Vestn J Mech E 61(3):187–195
32. Aparow VR, Ahmad F, Hudha K, Jamaluddin H (2013) Modelling and PID control of antilock braking system with wheel slip reduction to improve braking performance. Int J Veh Saf 6(3):265–296
33. Bakker E, Nyborg L, Pacejka HB (1987) Tyre modelling for use in vehicle dynamics studies. SAE, Warrendale
34. Pacejka H (2006) Tire and vehicle dynamics. Elsevier, Oxford
35. Ping EP, Hudha K, Jamaluddin H (2010) Hardware-in-the-loop simulation of automatic steering control for lanekeeping

manoeuvre: outer-loop and inner-loop control design. Int J Veh Saf 5(1):35–59
36. Short M, Pont M, Huang Q (2004) Simulation of vehicle longitudinal dynamics. University of Leicester, Embedded Systems Laboratory, Leicester
37. Wang J, Steiber J, Surampudi B (2008) Autonomous ground vehicle control system for high-speed and safe operation. In: American Control Conference, 2008, Seattle, WA, USA, June 11–13, IEEE, pp 218–223. https://doi.org/10.1109/acc.2008.4586494
38. Amer NH, Zamzuri H, Hudha K, Kadir ZA (2017) Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. J Intell Robot Syst 86(2):225–254. https://doi.org/10.1007/s10846-016-0442-0
39. Raffo GV, Gomes GK, Normey-Rico JE, Kelber CR, Becker LB (2009) A predictive controller for autonomous vehicle path tracking. IEEE Trans Intell Transp Syst 10(1):92–102. https://doi.org/10.1109/TITS.2008.2011697
40. Zakaria MA, Zamzuri H, Mamat R, Mazlan SA (2014) Optimized potential radius reference generator algorithm for autonomous vehicle controller development. Appl Mech Mater 663:198–202
41. Thrun S, Montemerlo M, Dahlkamp H, Stavens D, Aron A, Diebel J, Fong P, Gale J, Halpenny M, Hoffmann G, Lau K, Oakley C, Palatucci M, Pratt V, Stang P, Strohband S, Dupont C, Jendrossek L-E, Koelen C, Markey C, Rummel C, van Niekerk J, Jensen E, Alessandrini P, Bradski G, Davies B, Ettinger S, Kaehler A, Nefian A, Mahoney P (2006) Stanley: the robot that won the DARPA grand challenge. J Field Rob 23(9):661–692. https://doi.org/10.1002/rob.20147
42. Van Den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176(8):937–971
43. Shi Y, Eberhart RC (1998) Parameter selection in particle swarm optimization. In: Porto VW, Saravanan N, Waagen D, Eiben AE (eds) Evolutionary programming VII: 7th International Conference, EP98 San Diego, California, USA, March 25–27, 1998 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 591–600. https://doi.org/10.1007/bfb0040810
44. Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: Evolutionary Computation, 1997, IEEE International Conference on, 13–16 Apr 1997, pp 303–308. https://doi.org/10.1109/icec.1997.592326
45. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73
46. Paden B, Cap M, Yong SZ, Yershov D, Frazzoli E (2016) A Survey of motion planning and control techniques for self-driving urban vehicles. arXiv preprint arXiv:160407446. doi:https://arxiv.org/abs/1604.07446