Original Paper

MEASUREMENT AND CONTROL

Measurement and Control
2023, Vol. 56(9-10) 1510–1522
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/00202940231165257
journals.sagepub.com/home/mac
SAGE

Fuzzy predictive Stanley lateral controller with adaptive prediction horizon

Ahmed Abdelmoniem[1], Abdullah Ali[1], Youssef Taher[1], Mohamed Abdelaziz[2] and Shady A Maged[1]

Abstract

The challenge of trajectory tracking of autonomous vehicles (AVs) is a critical aspect that must be effectively addressed. Recent studies are concerned with maintaining the yaw stability to guarantee the customers' comfort throughout the journey. Most of the geometrical controllers solve this task by dividing it into consecutive point stabilization problems, limiting the controllers' ability to handle sudden trajectory changes. One research presented a predictive Stanley lateral controller that uses a discrete prediction model to mimic human behavior by anticipating the vehicle's future states. That controller is limited in its use, as the parameters must be manually tuned for every change in the maneuver or vehicle velocity. This article presents a novel approach for trajectory tracking in autonomous vehicles, by introducing a fuzzy supervisory controller that automatically adapts to changes in the vehicle's velocity and maneuver by estimating the prediction horizon's length and providing different weights for each controller. The proposed method overcomes the limitations of traditional controllers that require manual tuning of parameters, making it ready for real-world experiments. This is the main contribution of the research in this paper. The suggested technique demonstrated an advantage over the Basic Stanley controller and the manually tuned predictive Stanley controller in terms of the total lateral error and the model predictive control (MPC) in terms of computational time. The performance is determined by performing various simulations on V-Rep and hardware-in-the-loop (HIL) experiments on an E-CAR golf bus. A broad selection of velocities is used to validate the behavior of the vehicle while working on different maneuvers (double lane change, hook road, S road, and curved road).

Keywords

Autonomous vehicles, predictive Stanley controller, path tracking, model predictive control

Date received: 27 November 2022; accepted: 5 March 2023

Introduction

Autonomous systems have demonstrated remarkable applications in replacing inefficient or redundant human efforts. Self-driving cars considered as one of the areas which draw those fascinated with AI applications. Autonomous vehicles act as a crucial pillar in setting up the future picture of the country's infrastructure. It has a significant contribution to relieving road congestion and reducing car accidents caused by human mistakes. A lot of research relevant to this area has appeared on the scene to realize the potential and solve the challenges to reach this goal.

The future of urban mobility is supposed to be autonomous vehicles. By the end of 2025, self-driving cars are expected to be technologically mature enough for widespread use.[1] The most significant benefits of integrating autonomous vehicles into the transportation system are: (1) AVs are not prone to human errors, which will ensure passengers' safety during the trip, (2) Allows more significant opportunity to utilize wasted driving time in increasing production, (3) Providing simple, safe, and private transportation for those with disabilities that prohibit them from driving, and (4) Increase road capacity and reduce congestion, resulting in a saving of wasted energy "fuel consumption."[2]

In order for any self-driving car to be able to plan, navigate, and perform tasks in its environment, it has

[1]Mechatronics Engineering Department, Ain Shams University, Cairo, Egypt
[2]Automotive Engineering Department, Ain Shams University, Cairo, Egypt

Corresponding author:
Shady A Maged, Mechatronics Engineering Department, Ain Shams University, 1 Elsarayat St., Abbasya, Cairo 11517, Egypt.
Email: Shady.maged@eng.asu.edu.eg

to be able to localize itself in this environment as the vehicle position is the direct feedback of any lateral controller. Usually, a fusion algorithm is used as in Osman et al.[3] alongside different sensors, such as LIDARS, Global Positioning Systems (GPS), Inertial Measurement Units (IMUs), Cameras, and wheel-encoders.

This article introduces a Lateral steering control system for passenger cars, as it's one of the preeminent modules in autonomous vehicles. In our implementation, the trajectory is provided, and the car is controlled to pursuit the path by calculating the optimal steering angle. A Supervisory Controller (Fuzzy Logic) is also introduced to supplement what has been done in Abdelmoniem et al.[32] Abdelmoniem et al. propose the Predictive Stanley controller (PS) that mimics the driver driving behavior. It is relies on a discrete kinematic model that estimates the vehicle's coming states, allowing the use of the basic controller for upcoming expected states. Furthermore, to maintain the performance of PS, its parameters must be manually tuned for every change in the maneuver or vehicle speed. Consequently, a supervisory controller is added to automate the parameters tuning process and make the controller capable of being used in practical experiments.

The key contribution of this article is the use of a fuzzy supervisory controller to improve the parameterization of the predictive Stanley controller. The proposed method addresses the limitations of previous research by incorporating an online tuning feature for the controller's weights, allowing it to adapt to real-time conditions and eliminating the need for predetermined weights for different paths or vehicle speeds.

The article is organized as follows: In Section I, the design of the supervisory controller and how it was integrated with the predictive Stanley controller to make it suitable for real-time experiments is described. Section II provides an overview of the tests conducted to validate the proposed method, the evaluation criteria used, and the results obtained. Finally, in Section III, the conclusions and future directions are summarized.

Related work

The trajectory tracking module in AVs considered as a crucial challenge of motion controller design. This can be attributed to the vehicle's mobility constraints, the expected sharp maneuvers and the elevated vehicle velocities.

Among the several approaches examined was the PID controller.[4] Its simple structure allows for its ease of implementation on non-computationally extensive devices. However, the PID controller needs to have its coefficients K_P, K_I, and K_D tuned to their optimal values and, even then, it is not guaranteed that the controller gives its best performance. This, unfortunately, restricts the controller's capability to adapt with the environment changes where it cannot have an appropriate correctness at a wide-ranging of maneuvers or high vehicle's speeds. The limitations of the PID method led to the introduction of the adaptive methods that can adapt to different types of environments. For the tracking of predefined paths, Zhao et al.[5] used the lateral error (E), that depends on both longitudinal and side velocity, and the yaw angle (θ) in their proposed adaptive PID controller. Data driven & Neural Networks (NN) fanatics had their share carrying out a development were they added an NN to the previously developed adaptive PID control and that to improve the selection of the controller's parameters.[6] However, due to environmental noise its hard to obtain the labeling signal of supervised learning which results in limiting the application.

Other types of navigation controllers include the geometric controllers types. These make use of only the car's kinematics while tracking a reference path. Accordingly, such types of controllers ignore the driving and resistance forces applied on the car and assume that there are no slip between the tires and the road at all times. The most popular types of geometrical controllers used in lateral control in autonomous vehicles are (1) Pure Pursuit controller and (2) Stanley controller.

The pure pursuit controller Elbanhawi et al.,[7] Barton,[8] and Campbell,[9] being one of the most commonly used geometric controllers, maintains the vehicle on the road by guessing the necessary steering direction.[10] Sadly according to Coulter,[11] when implemented, the Pure Pursuit controller is only restricted to outside applications.[12] A supervisory controller (fuzzy controller) was introduced by Ollero et al.[13] The controller uses trajectory features, speed, and the error of the tracking to optimize the look-ahead distance and some other pure pursuit factors. Shan et al.[14] introduced CF-pursuit as a brand new pursuit method method where the error of the fitting is reduced using a clothoid C curve instead of the circles that are utilized in conventional pure-pursuit. In addition, they suggested a fuzzy system to establish a proper look-ahead distance online using the path's curvature.

The second category of geometric controllers is the Stanley controller,[15] responsible for Stanford's to DARPA challenge win in 2006. Studies Amer et al.[16] and Paden et al.[17] describe a good performance for the Stanley controller by taking together the side error and the yaw error into consideration. An earlier study by Amer et al.[18] introduced an adaptive Stanley controller that uses a fuzzy supervisory controller to automatically adjust the controller's parameters. However, this method only adjusts the parameters of a single controller and does not incorporate any additional controllers for future states. Early database information collected using Particle Swarm Optimization (PSO) is used as the basis for supervisory controller relying on the car speed and the yaw error. Although the controller showed great performing at high velocities, it remains untested on extremely severe trajectories.

Additional categories of geometric controllers exist like Follow-the-carrot[19] as well as other geometrical controllers Kanayama et al.[20] and Kuwata et al.[21]

Various methods of Optimal Control techniques are used among the autonomous vehicles. The most common methods used are (1) Model Predictive Control and (2) H-inf controller.

Model Predictive Control – MPC for short – is a repetitive method of optimization for the expected car states in a defined horizon window, although manipulating the control variables. The first input of the optimized horizon is then set to be achieved by the vehicle, and the whole procedure is repeated.[22] Additional developments on MPC were made like NMPC to cope with non-linearities in the systems and showed worthier performing. However, NMPC needs expensive computational time which restricts the products to perform in a practically. Mehrez et al.,[23] developed NMPC using a recently developed tool-kit to control nonholonomic robots. NMPC Path tracking control was later developed to pursuit a pre-defined reference path by Ritschel et al.,[24] on a VW Golf VII series vehicle. Where a path factor was added to the states to be optimized along with a virtual control input. MPC main drawback is that it needs weights tuning for each speed ranges and paths.

Additional controllers were reviewed such as the H-inf controller[25] which is used as an optimal control theory that achieves stabilization with a perfectly adequate performance by reducing the gain parameter, that is the result of minimizing the closed loop system. The non-linear system is linearized by figuring the system's jacobian matrices. To calculate the gain, the solution of Ricatti equation at each sampling step is required.[26]

Sliding Mode Controller is a robust control scheme Dong and Nguang[27] and Rigatos et al.[28] which is also widely used in motion control. Yet, its main drawback is the chattering phenomenon that occurs as the trajectory reaches the surfaces and remains there, it introduces some oscillations around the surface. However, many implementations were developed to solve this issue as in Nayl et al.[29] and Montaseri and Yazdanpanah.[30]

To sum up everything that has been stated so far, furthermost of the aforementioned approaches tried to implement a supervisory controller to style the basic controller's parameters to be adaptive as in Ollero et al.[13] and Amer et al.[18] or to reformulate the error determination technique as in Shan et al.[14] Zakaria et al.[31] proposed a trajectory tracking technique by the use of future estimate control, where the side error is determined based on only one upcoming point with no regards to the current car state. Nevertheless, the major trouble is that all of those techniques calculates the error based on one single point which is the closest point to the car. And that limits the controller's capability to deal with unexpected variations of the path yaw angle at extreme velocities. This publication proposes a novel approach that emulates the driving behavior of a human driver by reducing errors along the trajectory, rather than just at a single reference point, as previously done by Zakaria et al.[31] using a kinematic model. This new method anticipates the future states of the car, allowing the control algorithm to be applied not only to the current state but also to the expected future states. The study applies the Predictive Stanley controller and compares it with the basic Stanley controller developed in Hoffmann.[15]

The main contribution of this article is the introduction of a supervisory fuzzy controller to tackle the parameterization of the predictive Stanley controller. Our paper addresses limitations of the research presented in Abdelmoniem et al.[32] by incorporating an online tuning feature for the controller's weights. This allows the controller to adapt to real-time conditions and eliminates the need for predetermined weights for different paths or vehicle speeds. As a result, our approach improves upon the real-time performance of the controller, this implementation showed considerably better results over most of the controllers mentioned. Even with respect to the MPC, it needed less computational cost where the MPC took more time solving the same problem.

Methodology

The Basic Stanley (BS) controller uses the vehicle's current position and the desired path to calculate its heading error and lateral error. Abdelmoniem et al.[32] introduced the predictive Stanley controller, which utilizes a prediction model to anticipate the future states of the vehicle, thereby enhancing its overall performance. However, the efficiency of this controller is dependent on the vehicle speed and maneuver and as a result, its parameters must be manually adjusted for each change in speed range or maneuver, making its real-time application a challenge.

The proposed supervisory fuzzy controller receives the current state's lateral error and its rate of change as inputs. These inputs are used to calculate the optimal time step (dt) and prediction horizon (N) for the predictive model, including the first basic Stanley (BS) controller's weight K_0. Consequently, the controller can handle the sudden trajectory changes at different ranges of the vehicle's speed without tuning the previously mentioned parameters manually.

Basic Stanley

The Basic Stanley (BS) controller is divided into three regions, as illustrated in equation (1). These consist of the (low-high saturated – nominal) regions. Where ψ is defined to be the difference between the vehicle's heading angle (ψ_V) and trajectory's heading angle (ψ_{traj}) at the reference point and δ is the steering angle. The point of the shortest distance to the vehicle position, $e(t)$ is the lateral error, and W is the controller gain.

$$\delta(t) = \begin{cases} \psi + \arctan\left(\frac{We(t)}{V(t)}\right) & |\psi(t) + \arctan\left(\frac{We(t)}{V(t)}\right)| < \delta(max) \\ \delta(max) & \psi(t) + \arctan\left(\frac{We(t)}{V(t)}\right) \geqslant \delta(max) \\ -\delta(max) & \psi(t) + \arctan\left(\frac{We(t)}{V(t)}\right) \leqslant -\delta(max) \end{cases} \tag{1}$$

Predictive Stanley controller

The predictive Stanley controller (PS) consists of several consecutive Basic Stanley (BS) controllers and a prediction model. The prediction model presented by Abdelmoniem et al. is used to estimate the vehicle's future states at a fixed time step. Each controller receives its input state $[X_i, Y_i, \Theta_i]$ from the prediction model. Furthermore, every Basic Stanley (BS) controller calculates a control action based on the corresponding state received. The predictive Stanley (PS) controller's final control action is the weighted sum of all the control actions. Figure 1 shows the block diagram of the predictive Stanley controller.

Equations (2) and (3) illustrate the controller's mathematical formulation after adding the outputs of the Basic Stanley (BS) controllers, where K_i is the weight assigned to each controller in the future state i and $\delta(t)$ is the final output of the predictive Stanley controller.

$$\delta(t) = \sum_{i=0}^{N} K_i \left[\psi(t) + \arctan\left(\frac{We(t)}{V(t)}\right)_i \right] \tag{2}$$

$$\sum_{i=0}^{N} K_i = 1 \tag{3}$$

Limitations of predictive Stanley controller. The predictive Stanley controller has two main parameters: the time step (dt) by which the model predicts future states and the control action weight K_0 for the Basic Stanley controller$_0$. The values of these parameters significantly affect the controller's performance, and they need to be manually tuned for each change in the maneuvers or vehicle's speed. Consequently, this limits its use in practical applications where a vehicle is expected to perform different maneuvers with a significant variation in speed ranges.

Fuzzy predictive Stanley controller

The proposed controller uses fuzzy logic to estimate the optimal values of the prediction model's time step (dt) and the first controller's weight (K_0). This allows the Predictive Stanley to be used in real-time experiments without manually tuning the parameters for each maneuver or vehicle speed. Figure 2 shows the block diagram of the fuzzy predictive Stanley controller where e_l and $d(e_l)$ are the lateral error and its rate of change, respectively. In addition, the trajectory of the vehicle is represented as a series of waypoints, which include the x, y, and theta coordinates. These waypoints act as set points for the controller. The current controller, which receives the real-time position of the vehicle as input, selects a reference point on the path and calculates the lateral error and heading based on this point.

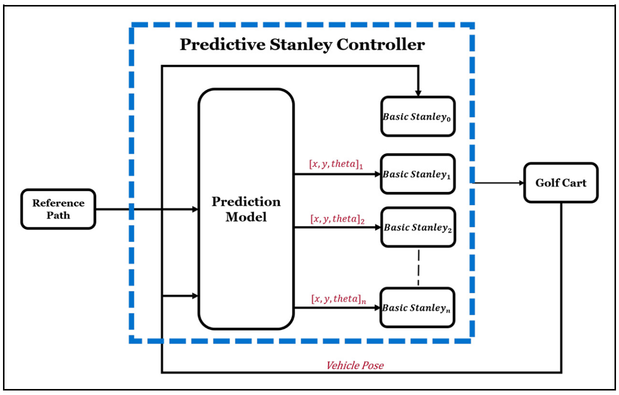


Figure 1. Predictive Stanley controller block diagram.

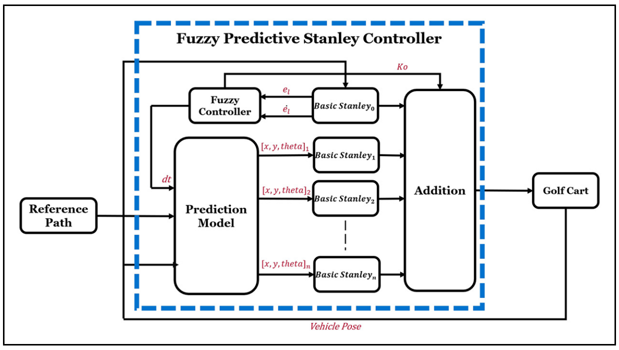


Figure 2. Fuzzy predictive Stanley controller block diagram.

This section details the input membership functions, fuzzy rules, and output membership functions. Those components are essential to design the fuzzy supervisory controller. That improves the Predictive Stanley (PS) controller's overall performance in terms of the lateral error and gives it the capability to be used in real-time environments. where the linguistic variables: High (H), Low (Low), Medium-High (MH), Medium Low (ML), Negative-Big (NB), Negative-Medium (NM), Negative-Small (NS), zero (ZO), Positive-Small (PS), Positive-Medium (PM), and Positive-Big (PB) are used to represent the controller inputs and the outputs.

Input membership functions. The lateral error and its rate of change de(t) received numerically and then normalized to range from −1 to 1. These values are mapped to the following linguistic variables (NB, BM, NS, ZO, PS, PM, and PB). Figures 3 and 4 show the membership functions.

Output membership functions. The first output is K_0 which is the weight of the controller that depends on the

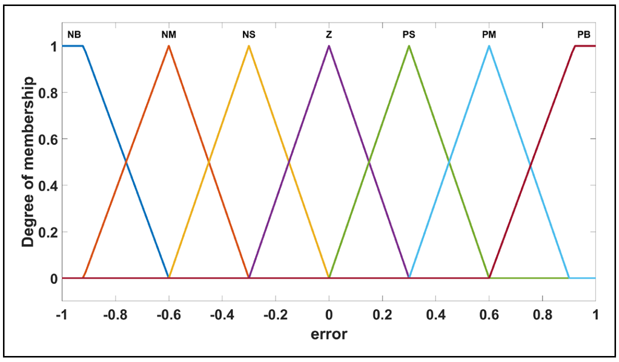


Figure 3. Error membership function.

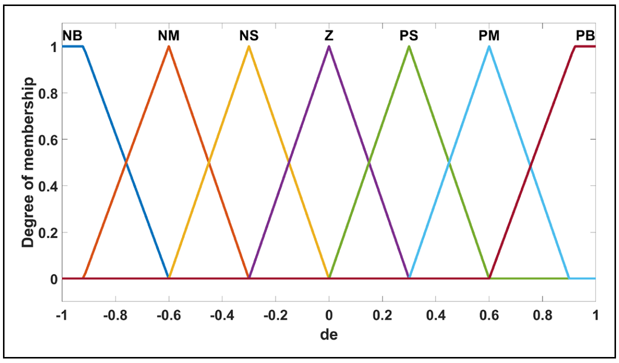


Figure 4. Rate of error membership function.

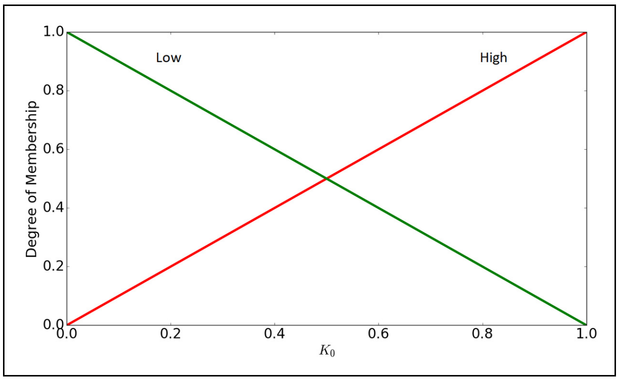
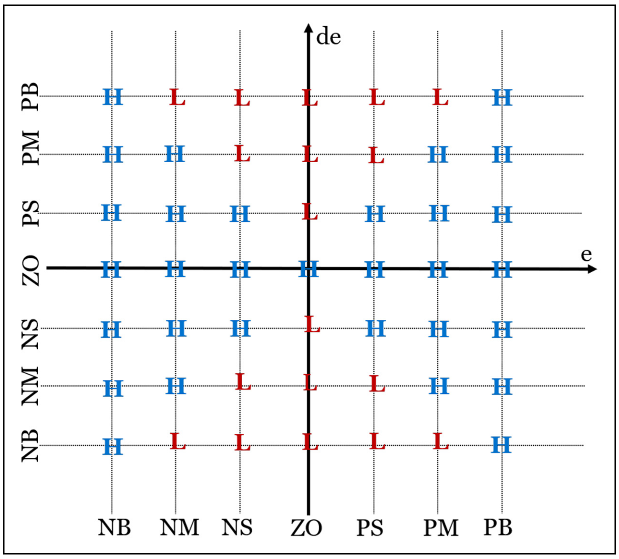


Figure 5. K_0 membership function.

vehicle's current state, as shown in Figure 2 and illustrated in equation (2). K_0 ranges between 0 and 1, and the sum of all weights ($K_0 \rightarrow K_N$) must be equal to one, as shown in equation (3).

The optimal value of K_0 is calculated based on the output membership function shown in Figure 5. The fuzzy rules are designed as in Figure 6, which gives lower values for K_0 if the lateral error is smaller than its rate of change. In this case, the controller's final output will be biased toward the controllers in the future states to deal with the future error.

The second output is the time step dt passed to the prediction model[32] as shown in Figure 7 and equations (4)–(6). It also ranges between 0 and 0.6.



	NB	NM	NS	ZO	PS	PM	PB
PB	H	L	L	L	L	L	H
PM	H	H	L	L	L	H	H
PS	H	H	H	L	H	H	H
ZO	H	H	H	H	H	H	H
NS	H	H	H	L	H	H	H
NM	H	H	L	L	L	H	H
NB	H	L	L	L	L	L	H

Figure 6. Fuzzy rules of K_0.

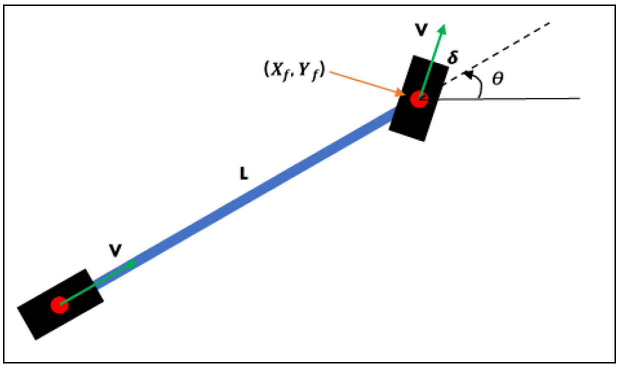


Figure 7. Kinematic bicycle model.

$$\theta_{(N)} = \theta_{(N-1)} + \frac{V \tan \delta}{L} dt \tag{4}$$

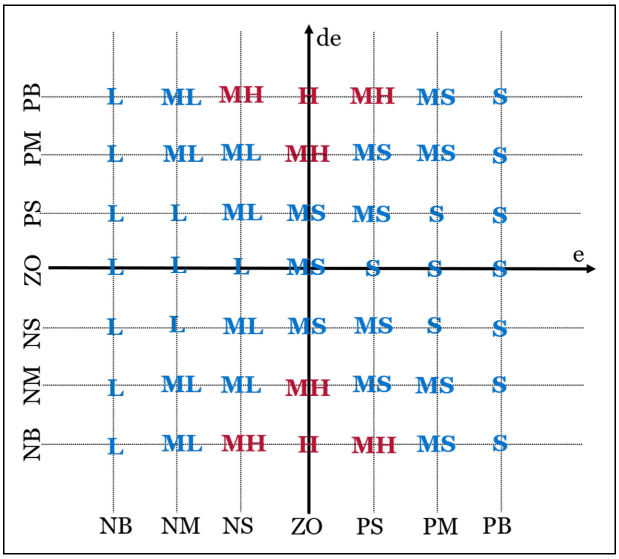
$$X_{f(N)} = X_{f(N-1)} + V \cos(\theta + \delta) dt \tag{5}$$

$$Y_{f(N)} = Y_{f(N-1)} + V \sin(\theta + \delta) dt \tag{6}$$

The optimal value of dt is calculated based on the output membership function shown in the figure. The fuzzy rules are designed as in Figure 8, which gives higher values for dt if the lateral error is smaller than its rate of change. In this case, the model will predict the future states further than before, giving the capability to the overall controller to deal with the future error.

To summarize, the areas in red in Figures 6 and 8 indicate that the final control action will be heavily influenced by the future state controllers, while the blue areas indicate that the current state controller will have a greater impact.

Fuzzy rules. Table 1 shows the fuzzy rules used to infer the control action weight (K0).



de \ e	NB	NM	NS	ZO	PS	PM	PB
PB	L	ML	MH	H	MH	MS	S
PM	L	ML	ML	MH	MS	MS	S
PS	L	L	ML	MS	MS	S	S
ZO	L	L	L	MS	S	S	S
NS	L	L	ML	MS	MS	S	S
NM	L	ML	ML	MH	MS	MS	S
NB	L	ML	MH	H	MH	MS	S

Figure 8. Fuzzy rules of *dt*.

Table 1. Fuzzy rules of K_0.

		de(*t*)						
		NB	NM	NS	ZO	PS	PM	PB
e(*t*)	NH	H	H	H	H	H	H	H
	NM	L	H	H	H	H	H	L
	NL	L	L	H	H	H	L	L
	ZO	L	L	L	H	L	L	L
	PL	L	L	H	H	H	L	L
	PM	L	H	H	H	H	H	L
	PH	H	H	H	H	H	H	H

Table 2. Fuzzy rules of *dt*.

		de(*t*)						
		NB	NM	NS	ZO	PS	PM	PB
e(*t*)	NH	L	L	L	L	L	L	L
	NM	ML	ML	L	L	L	ML	ML
	NL	MH	ML	ML	L	ML	ML	MH
	ZO	H	MH	ML	ML	ML	MH	H
	PL	MH	ML	ML	L	ML	ML	MH
	PM	ML	ML	L	L	L	ML	ML
	PH	L	L	L	L	L	L	L

Table 2 shows the rules used to infer the time step output *dt* for the prediction model.

Adaptive prediction horizon

Equation (7) illustrates how the prediction horizon is calculated. Where N, V, and dt are the prediction horizon, the actual vehicle's longitudinal velocity, and the prediction model's time step.

It's clear that the prediction horizon (N) increases at a high vehicle's speed or when the current state's controller's weight is relatively high giving the overall controller the ability to deal with sudden errors

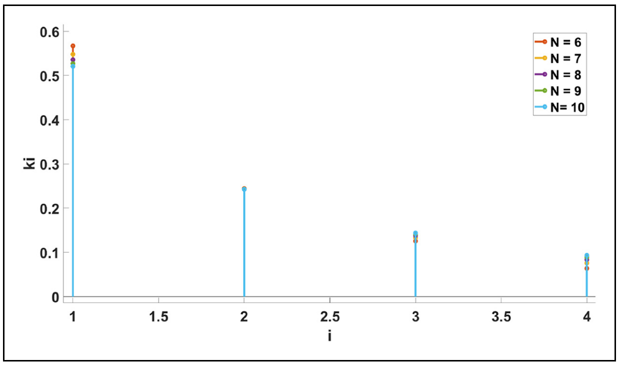


Figure 9. Controllers' weights (K_i) versus order of each controller (i).

$$N = \frac{K_0^2 * V}{dt} \tag{7}$$

Equation (8) illustrates that we measure each controller's weight in the future state (K_i), where K_f, i, and η refer to the future states' controllers' overall contribution, each controller's order, and a normalization factor, respectively.

$$K_i = \eta \frac{K_f * (N - i)}{1 + N * i} \tag{8}$$

As shown in Figure 9, each controller's weight K_i decreases as the predicted state i becomes further from the current state. The value for K_i remains mostly unchanged with only very slight variations as the total number of predicted states N increases.

Experimental work and results

The proposed methodology was tested and validated on various maneuvers, including double lane change, hook road, and curved road, with varying velocities through multiple simulations and real-world experiments. This section showcases the results of the trials and the evaluation criteria used. The results are also compared with those obtained from the Predictive Stanley controller (PS) and the Model Predictive Controller (MPC). The evaluation standards used in this section are also presented.

Evaluation metrics

The evaluation metrics presented by Abdelmoniem et al.[32] are used when comparing the performing of the suggested controller to both the predictive and basic Stanley controllers. Testing was performed using several maneuvers at different velocities to calculate the following errors as shown in equations (9)–(12) the RMS of the lateral error e_{RMS}, the RMS of the heading

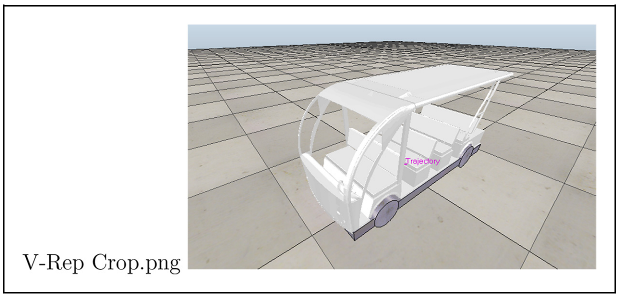


Figure 10. Simulation environment.

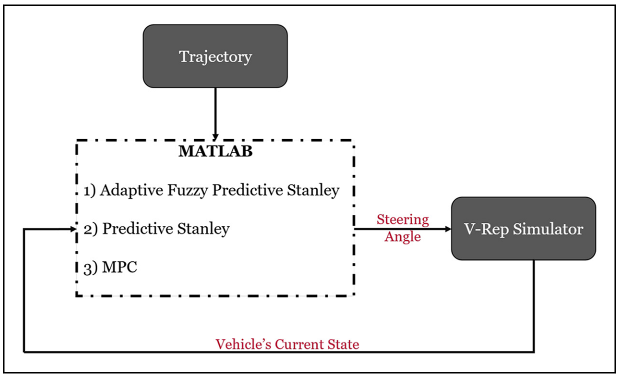


Figure 11. Communication architecture.

error ψ_{RMS}, the RMS of the yaw rate r_{RMS}, and the RMS of the change in the control action ΔU_{RMS}.

$$e(t)_{RMS} = \sqrt{\sum \frac{e(t)^2}{n}} \tag{9}$$

$$\psi(t)_{RMS} = \sqrt{\sum \frac{\psi(t)^2}{n}} \tag{10}$$

$$r(t)_{RMS} = \sqrt{\sum \frac{r(t)^2}{n}} \tag{11}$$

$$\Delta U(t)_{RMS} = \sqrt{\sum \frac{\Delta U_{RMS}(t)^2}{n}} \tag{12}$$

Simulation setup

The proposed controller was developed using MATLAB and the Ode3 (Bogacki-Shampine) solver, with a fixed time step of 0.01 s. Vehicle modeling was performed in V-REP, as depicted in Figure 10, with the V-REP car model serving as the system plant. The current state of the car is communicated to the proposed controller through ROS. Subsequently, the controller sends the control output from MATLAB to the V-REP simulator, as shown in Figure 11.

Figure 12. Golf cart.

Real-time experiments setup

The level-3 autonomous golf cart, shown in Figure 12, was used in the proposed controller's real-time validation and testing. Similar tests were performed using the previously mentioned controllers for comparisons.

The golf cart was fitted with a VLP-16 LiDAR, a Vn-100 IMU, and wheel encoders. The ROS package "loam-velodyne"[33] is used to estimate the vehicle's position base on the LIDAR odometry. The IMU and the wheel encoders also provided an estimate of the vehicle's odometry and current velocity. Furthermore, a sensor fusion algorithm was performed using Extended Kalman Filter (EKF).[34]

The fused odometry pose is then fed to the Adaptive Monte Carlo Localization (AMCL) ROS package,[35] alongside a static map and the LIDAR points as shown in Figure 13.

The vehicle's estimated pose is published to the proposed controller's node through ROS to publish the low-level controller's desired steering angle.

Experimenting with MPC

Considering the following non-linear system,

$$\dot{X}(t) = f(x(t), u(t)), \tag{13}$$

where $X \in \mathbb{R}^n$ is the state vector $= [x, \ y, \ \theta]^T$ and $U \in \mathbb{R}^m$ is the control vector $= [\delta]$, and the system's kinematic model which was discretized using the Euler Discretization technique into a sample form to use in the prediction of the next states.

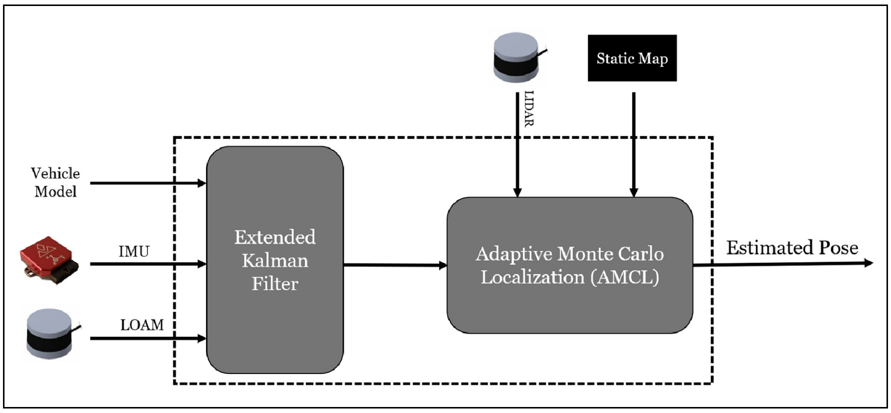


Figure 13. Real-time system architecture.

Table 3. FPS-PS (10 m/s).

Maneuvers	e_{RMS} (FPS-PS)	ψ_{RMS} (FPS-PS)	r_{RMS} (FPS-PS)	ΔU_{RMS} (FPS-PS)
CR	(0.0477–0.0713)	(0.0464–0.0612)	(0.1356–0.1494)	(0.6541–0.6231)
HR	(0.1271–0.1327)	(0.1011–0.1164)	(0.2041–0.2055)	(0.6021–0.5608)
DLC	(0.0323–0.0430)	(0.0389–0.0446)	(0.1023–0.1198)	(0.5125–0.4127)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} vcos(\theta + \beta) \\ vsin(\theta + \beta) \\ \frac{vcos\beta sin\delta}{L} \end{bmatrix} \tag{14}$$

The cost function to be minimized over the prediction horizon is defined as:

$$J(x,u) = \| x_u(k) - x_{ref}(k) \|_Q^2 + \| u(k) - u_{ref}(k) \|_R^2 \tag{15}$$

$$\min \ J_N(x,u) = \sum_{k=0}^{N-1} (x_u(k), u(k)) \tag{16}$$

subject to the following constraints:

$$x_u(k+1) = f(x_u(k), u(k)) \tag{17}$$

$$x_u(0) = x_0 \tag{18}$$

$$u(k) \in U, \forall k \in [0, N-1] \tag{19}$$

$$x_u(k) \in X, \forall k \in [0, N] \tag{20}$$

where x_{ref} is the predetermined reference path, x_u is the current vehicle state, $Q = \text{diag}\ (q_x, q_y, q_\theta)$, $R = \text{diag}\ (r_v, r_\delta)$ (where Q and R are positive definite weighting matrices), and N is the prediction horizon. Non-linear Model Predictive control online optimization problem is stated by transforming the Optimal Control Problem 16 with constraints (17 → 20) to a non-linear programming problem 21 and 22 and solving using NLP optimization technique.

$$\min_w \Phi(w) \tag{21}$$

$$s.t. \ g(w) \leq 0, g_2(w) = 0, \tag{22}$$

The reference path is assigned as a pre-defined vector of points

$$[x_{ref}, y_{ref}, \theta_{ref}]^T$$

The point where the prediction horizon starts is set by finding the one with the shortest distance to the vehicle.

Results

Simulation results. This section demonstrates the obtained results from the simulation and plots the actual path followed by the vehicle when carrying out several maneuvers at different velocities with the Fuzzy Predictive Stanley (FPS), Predictive Stanley (PS), Basic Stanley (BS), and MPC controllers.

(1) Fuzzy Predictive Stanley Versus Predictive Stanley (FPS vs PS)

Table 3 demonstrates the superiority of the FPS controller over the PS controller when performing different maneuvers at 10 m/s.

Table 4. FPS-PS (15 m/s).

Maneuvers	e_{RMS} (FPS-PS)	ψ_{RMS} (FPS-PS)	r_{RMS} (FPS-PS)	ΔU_{RMS} (FPS-PS)
CR	(0.0783–0.1240)	(0.0524–0.0895)	(0.1421–0.1620)	(0.6614–0.4682)
HR	(0.3472–0.6854)	(0.1692–0.2330)	(0.2815–0.4561)	(0.7251–0.8283)
DLC	(0.0361–0.0470)	(0.0423–0.0458)	(0.1021–0.1350)	(0.6715–0.5704)

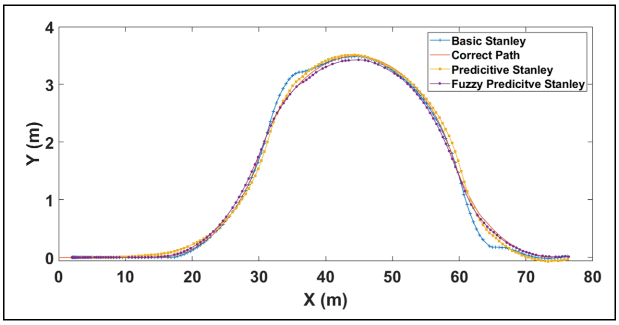


Figure 14. Simulation curved road 10 m/s.

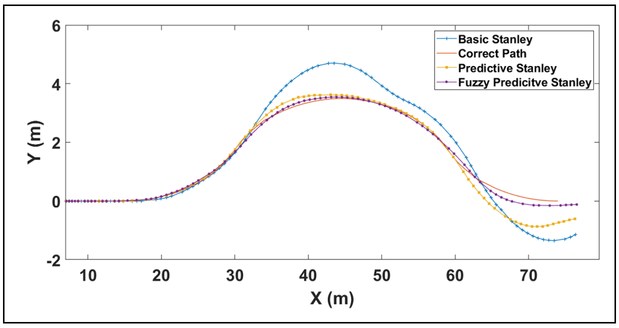


Figure 15. Simulation curved road 15 m/s.

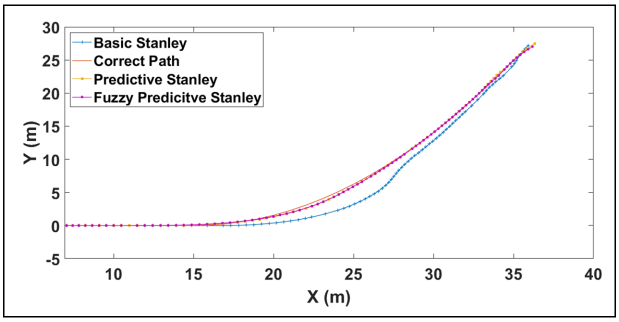


Figure 16. Simulation hooked road 10 m/s.

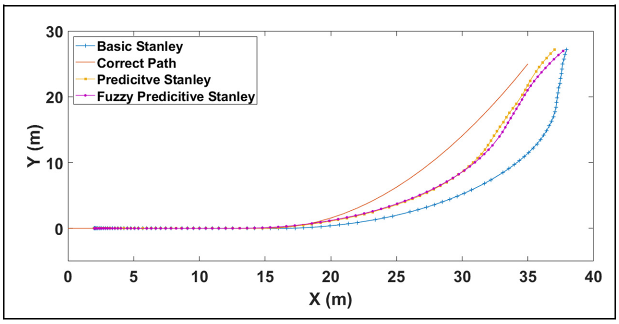


Figure 17. Simulation hooked road 15 m/s.

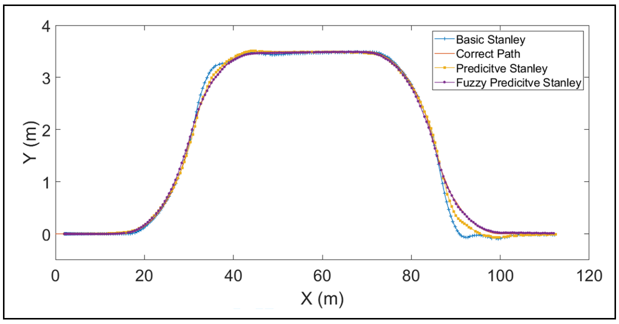


Figure 18. Simulation double lane change 10 m/s.

Table 4 demonstrates the superiority of the FPS controller over the PS controller when performing different maneuvers at 15 m/s.

Figure 14 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a curved road at 10 m/s.

Figure 15 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a curved road at 15 m/s.

Figure 16 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a hook road at 10 m/s.

Figure 17 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a hook road at 15 m/s.

Figure 18 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a double lane change at 10 m/s.

Figure 19 demonstrates the superiority of the fuzzy predictive Stanley controller over the predictive and the basic Stanley controllers for a double lane change at 15 m/s.

When analyzing the outcomes, it is important to acknowledge that the predictive Stanley controller was manually calibrated for each maneuver type and at various vehicle speeds. In contrast, the adaptive fuzzy predictive Stanley controller required only a single tuning process and was subsequently appropriate for all types of maneuvers and vehicle speeds. Furthermore, the adaptive fuzzy Stanley controller outperformed the predictive Stanley controller significantly in all maneuvers and at all vehicle speeds, despite the manual tuning efforts made for the predictive controller.

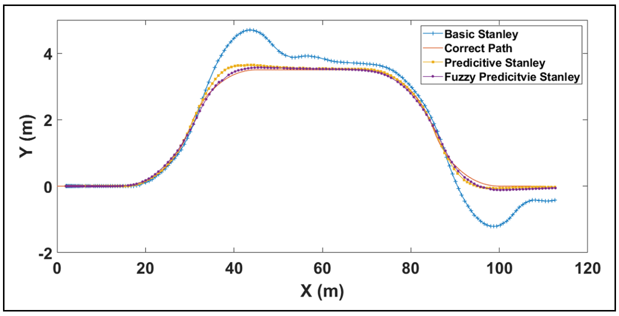


Figure 19. Simulation double lane change 15 m/s.

Table 5. FPS-MPC (10 m/s).

Maneuvers	e_{RMS} (FPS-MPC)
CR	(0.0477–0.0421)
DLC	(0.0323–0.0301)

Table 6. FPS-MPC (15 m/s).

Maneuvers	e_{RMS} (FPS-MPC)
CR	(0.0783–0.0692)
DLC	(0.0361–0.0312)

(2) Fuzzy Predictive Stanley Versus Model Predictive Control (FPS vs MPC)

Previous simulations illustrated the clear advantage that the fuzzy predictive Stanley controller has over the predictive Stanley controller. For that reason, the fuzzy predictive Stanley controller was compared with a model predictive controller. Both controllers' accuracies and processing times were taken into consideration.

Table 5 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller in terms of the lateral error at 10 m/s.

Table 6 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller in terms of the lateral error at 15 m/s.

Figure 20 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a curved road at 10 m/s.

Figure 21 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a curved road at 15 m/s.

Figure 22 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a double lane change at 10 m/s.

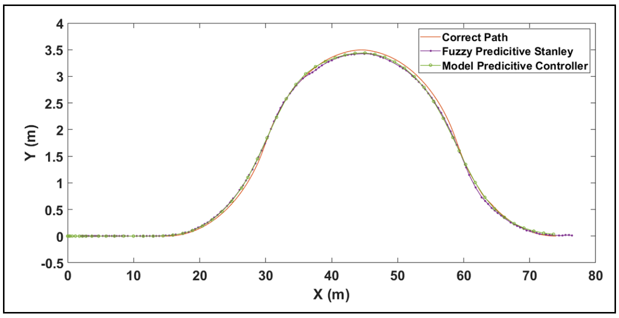


Figure 20. Simulation MPC versus FPS curved road 10 m/s.

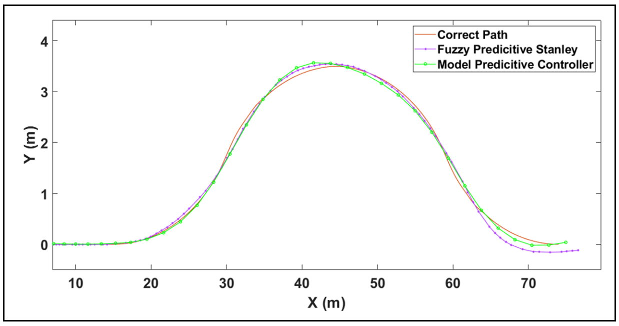


Figure 21. Simulation MPC versus FPS curved road 15 m/s.

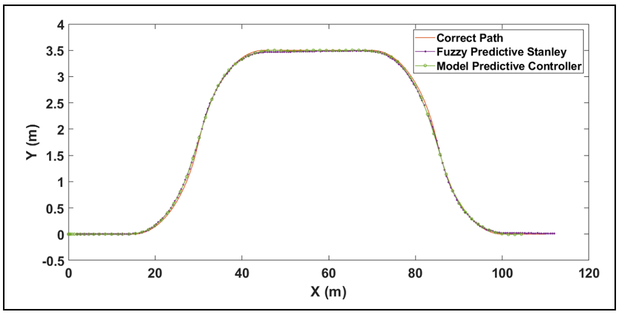


Figure 22. Simulation MPC versus FPS double lane change 10 m/s.

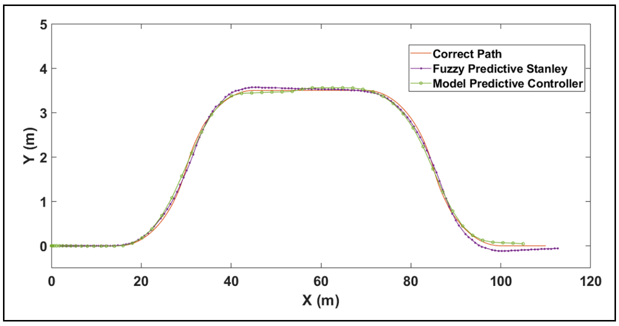


Figure 23. Simulation MPC versus FPS double lane change 15 m/s.

Figure 23 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a double lane change at 15 m/s.

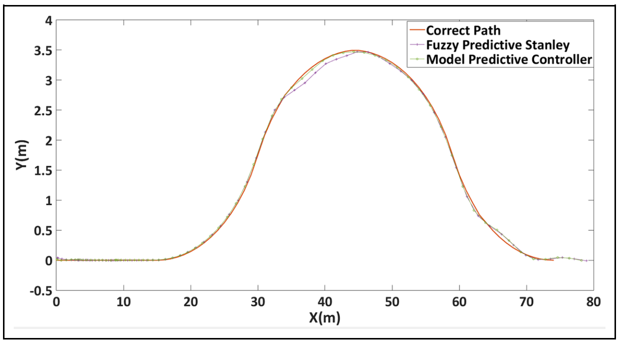


Figure 24. MPC versus FPS curved road 10 m/s.

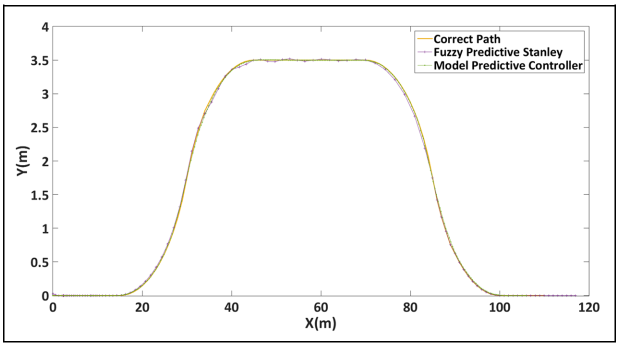


Figure 25. MPC versus FPS double lane change 10 m/s.

Despite their similarity in performance, the fuzzy predictive Stanley controller showed significantly lower processing times than the model predictive controller. This makes it more suitable for real-time hardware implementations where computational power may be limited.

Real time experiments. As demonstrated by the simulation results, the FPS controller and MPC displayed comparable performance. However, the fuzzy predictive Stanley controller displayed significantly lower processing times compared to the MPC. In this section, our objective is to validate these results through real-time experiments, which are similar to the simulations, but implemented on a golf cart. Due to the speed limit of our vehicle, these experiments were only conducted at 10 m/s.

We only did this experiments on 10 m/s, due to the speed limit of our vehicle.

Figure 24 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a curved road at 10 m/s.

Figure 25 shows the approximate similarity in performance of the fuzzy predictive Stanley controller and the model predictive controller for a double lane change at 10 m/s.

Table 7 illustrates the comparable performance of the fuzzy predictive Stanley controller and the model predictive controller in terms of lateral error at 10 m/s. Furthermore, it highlights the advantage of the FPS in terms of computational time, making it a more suitable choice for real-time applications.

Table 7. FPS-MPC (10 m/s).

Maneuvers	e_{RMS} (FPS-MPC)	Average Frequency (FPS-MPC)
CR	(0.0523–0.0487)	(15–4 Hz)
DLC	(0.0389–0.0375)	(15–4 Hz)

Conclusion

A study on the Adaptive Fuzzy Predictive Stanley Lateral Controller for a golf cart (14 seaters) is presented in this article. The adaptive fuzzy approach mainly adds a supervisory controller to the Predictive Stanley (PS) controller to perform online tuning to its parameters K_0 and dt. An adaptive prediction horizon is maintained using a decaying function that determines the weight K_i of the control action based on the predicted horizon (N) and the order of the controller i.

Different maneuvers were tested using V-REP simulations and real-life experiments on a level-3 autonomous golf cart at different vehicle speeds. The results obtained using the FPS controller had lower values of e_{RMS}, ψ_{RMS}, and r_{RMS}, indicating an overall advantage over the PS controller. The FPS controller also proved successful when operating in real-time. This was not previously possible for the predictive Stanley controller due to its need for manual tuning at different speed ranges for each maneuver type.

The fuzzy predictive Stanley controller was then evaluated against a path-following MPC. Although both controllers produced comparable levels of error, the proposed FPS controller had significantly lower processing times compared to the MPC. This advantage in computational efficiency makes the FPS controller a more practical solution for real-time applications where computational resources may be limited.

Acknowledgement

We are thankful to our colleagues Ahmed Osama, Hady Hafez, and Fatma Maher who provided expertise that greatly assisted the research.

Authors' contributions

All the authors "Abdelmoniem, Ali, Afifi, Abdelaziz, and Maged" equally shares the contribution of this research.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Shady A Maged https://orcid.org/0000-0001-8641-9985

References

1. Fafoutellis P and Mantouka E. Major limitations and concerns regarding the integration of autonomous vehicles in urban transportation systems. In: Nathanail E and Karakikes I (eds) *Data analytics: paving the way to sustainable urban mobility*. Vol. 879. Cham: Springer, 2019, pp.739–747.
2. Giuffrè T, Canale A, Severino A, et al. Automated vehicles: a review of road safety implications as a driver of change. In: *Proceedings of the 7th CARSP conference*, Toronto, ON, Canada, 18–21 June 2017.
3. Osman M, Hussein A and Al-Kaff A. Intelligent vehicles localization approaches between estimation and information: a review. In: *IEEE international conference of vehicular electronics and safety (ICVES)*, Cairo, Egypt, 4–6 September 2019.
4. Farag W. Track maneuvering using PID control for self-driving cars. *Recent Adv Electr Electron Eng* 2020; 13: 91–100.
5. Zhao P, Chen J, Song Y, et al. Design of a control system for an autonomous vehicle based on adaptive-PID. *Int J Adv Robot Syst* 2012; 9: 44.
6. Li Z, Li Y, Yang C, et al. Motion control of an autonomous vehicle based on wheeled inverted pendulum using neural-adaptive implicit control. In: *2010 IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 18–22 October 2010.
7. Elbanhawi M, Simic M and Jazar R. The role of path continuity in lateral vehicle control. *Procedia Comput Sci* 2015; 60: 1289–1298.
8. Barton MJ. *Controller development and implementation for path planning and following in an autonomous urban vehicle*. Bachelor's Thesis, The University of Sydney, Australia, 2001.
9. Campbell SF. *Steering control of an autonomous ground vehicle with application to the DARPA urban challenge*. Master's Thesis, Massachusetts Institute of Technology, USA, 2007.
10. Wallace R, Stentz A, Thorpe CE, et al. First results in robot road-following. In: *IJCAI*, 1985, pp.1089–1095.
11. Coulter RC. *Implementation of the pure pursuit path tracking algorithm*. Report, Carnegie-Mellon University Pittsburgh, Robotics Institute, 1992.
12. Li H, Luo J, Yan S, et al. *Research on parking control of bus based on improved pure pursuit algorithms*. In: *International symposium on distributed computing and applications for business engineering and science (DCABES)*, Wuhan, China, 8–10 November 2019.
13. Ollero A, García-Cerezo A and Martínez JL. Fuzzy supervisory path tracking of mobile reports. *Control Eng Pract* 1994; 2: 313–319.
14. Shan Y, Yang W, Chen C, et al. CF-pursuit: a pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *Int J Adv Robot Syst* 2015; 12: 134.
15. Hoffmann GM. Autonomous automobile trajectory tracking for off-road driving. In: *American control conference*, New York, NY, USA, 9–13 July 2007.
16. Amer NH, Hudha K, Zamzuri H, et al. Hardware-in-the-loop simulation of trajectory following control for a light armoured vehicle optimised with particle swarm optimisation. *Int J Heavy Veh Syst* 2019; 26: 663.
17. Paden B, Cap M, Yong SZ, et al. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans Intell Vehicles* 2016; 1: 33–55.
18. Amer NH, Hudha K, Zamzuri H, et al. Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle. *Robot Auton Syst* 2018; 105: 94–111.
19. Rankin AL, Crane CD, Armstrong DG, et al. Autonomous path-planning navigation system for site characterization. In: *Navigation and control technologies for unmanned system*, Orlando, FL, USA, 1996.
20. Kanayama Y, Kimura Y, Miyazaki F, et al. A stable tracking control method for an autonomous mobile robot. In: *IEEE international conference on robotics and automation*, Cincinnati, OH, USA, 13–18 May 1990.
21. Kuwata Y, Teo J, Karaman S, et al. Motion planning in complex environments using closed-loop prediction. In: *AIAA guidance, navigation, and control conference and exhibit*, Honolulu, HI, USA, 18–21 August 2008.
22. Qin SJ and Badgwell TA. A survey of industrial model predictive control technology. *Control Eng Pract* 2003; 11: 733–764.
23. Mehrez MW, Mann GKI and Gosine RG. Formation stabilization of nonholonomic robots using nonlinear model predictive control. In: *2014 IEEE 27th Canadian conference on electrical and computer engineering (CCECE)*, Toronto, ON, Canada, 4–7 May 2014.
24. Ritschel R, Schrödel F, Hädrich J, et al. Nonlinear model predictive path-following control for highly automated driving. *IFAC-PapersOnLine* 2019; 52: 350–355.
25. Osuna-Ibarra L, Caballero-Barragán H, Loukianov AG, et al. Tracking control using optimal discrete-time H_∞ for mechanical systems: applied to robotics. *Robot Auton Syst* 2019; 119: 201–208.
26. Rigatos G, Siano P, Wira P, et al. Nonlinear optimal control for autonomous navigation of a truck and trailer system. In: *2017 18th international conference on advanced robotics (ICAR)*, Hong Kong, China, 10–12 July 2017.
27. Dong L and Nguang SK. *Consensus tracking of multi-agent systems with switching topologies*. 1st ed. London: Academic Press, 2020.
28. Rigatos GG, Tzafestas CS and Tzafestas SG. Mobile robot motion control in partially unknown environments using a sliding-mode fuzzy-logic controller. *Robot Auton Syst* 2000; 33: 1–11.
29. Nayl T, Nikolakopoulos G, Gustafsson T, et al. Design and experimental evaluation of a novel sliding mode controller for an articulated vehicle. *Robot Auton Syst* 2018; 103: 213–221.

30. Montaseri G and Yazdanpanah MJ. A model predictive control approach to predict sliding surface. *IFAC Proc Volumes* 2008; 41: 9894–9898.
31. Zakaria MA, Zamzuri H, Mamat R, et al. A path tracking algorithm using future prediction control with spike detection for an autonomous vehicle robot. *Int J Adv Robot Syst* 2013; 10: 309.
32. Abdelmoniem A, Osama A, Abdelaziz M, et al. A path-tracking algorithm using predictive Stanley lateral controller. *Int J Adv Robot Syst* 2020; 17: 1729881420974852.
33. Zhang J and Singh S. LOAM: lidar odometry and mapping in real-time. *Robot Sci Syst* 2014; 2(9): 1–9.
34. Ahmad H and Namerikawa T. Extended Kalman filter-based mobile robot localization with intermittent measurements. *Syst Sci Control Eng* 2013; 1(1): 113–126.
35. Li T, Sun S and Duan J. Monte Carlo localization for mobile robot using adaptive particle merging and splitting technique. In: *IEEE international conference on information and automation*, Harbin, China, 20–23 June 2010.