Hindawi
Complexity
Volume 2023, Article ID 8720849, 9 pages
https://doi.org/10.1155/2023/8720849

Research Article

A New Self-Tuning Nonlinear Model Predictive Controller for Autonomous Vehicles

Yasin Abdolahi[1], **Sajad Yousefi**[2] **and Jafar Tavoosi**[1]

[1] *Department of Electrical Engineering, Ilam University, Ilam, Iran*
[2] *Department of Electrical Engineering, Technical and Vocational University (TVU), Tehran, Iran*

Correspondence should be addressed to Jafar Tavoosi; j.tavoosi@ilam.ac.ir

Received 22 August 2022; Revised 30 November 2022; Accepted 3 January 2023; Published 16 January 2023

Academic Editor: Francesco Lo Iudice

Autonomous driving has recently been in considerable progress, and many algorithms have been suggested to control the motions of driverless cars. The model predictive controller (MPC) is one of the efficient approaches by which the speed and direction of the near future of an automobile could be predicted and controlled. Even though the MPC is of enormous benefit, the performance (minimum tracking error) of such a controller strictly depends on the appropriate tuning of its parameters. This paper applies the particle swarm optimization (PSO) algorithm to find the global minimum tracking error by tuning the controller's parameters and ultimately calculating the front steering angle and directed motor force to the wheels of an autonomous vehicle (AV). This article consists of acquiring vehicle dynamics, extended model predictive control, and optimization paradigm. The proposed approach is compared with previous research in the literature and simulation results show higher performance, and also it is less computationally expensive. The simulation results show that the proposed method with only three adjustable parameters has an overshoot of about 8% and its RMSE is 0.72.

1. Introduction

Driverless cars have become popular in recent years because they promise to transform cities, provide universal access to mobility, and increase transport efficiency [1]. Another reason which attracts researchers and automobile industries is autonomous cars are expected to be by far the safest and more comfortable compared to human driving. Human driving is accident-prone as our tiredness and even wrong or delayed decisions undeniably affect our driving efficiency, which causes loss of property and deaths [2]. In recent decades, numerous studies of research pertinent to the dynamic and control of four-wheel automobiles have been conducted. In [3], the statistical performance of kinematic and dynamic bicycle models is analyzed. Such models are the simplification of four-wheel cars in which two rear wheels and two front wheels are lumped into one (imaginary) rear and front wheel, respectively. A more complex vehicle dynamic is developed in [4] which consists of five connected subsystems: one vehicle body and four wheels, rigidly coupled to the vehicle body. But the more sophisticated a model, the more difficult would be the identification of its parameters and the more challenging approaches to design efficient controllers. Moreover, the nonlinearities of the vehicle's dynamic represent big challenges against the controller design. While, in many cases, linearization is a good tradeoff between simplicity and accuracy, the nonlinear models are generally expected to be more trustworthy in various scenarios as cover all dynamical moods of a system [5].

In the case of controllers, different control techniques have been utilized for the steering angle and speed control of autonomous vehicles. Proportional-integral-derivative (PID) controller is one of the well-known classical methods. In [6], PID gains are adjusted by a radial basis neural network for AVs to control the AV's speed. A PID controller is developed for controlling AV's lateral position in [7]. However, PID controllers cannot handle the system constraints, and they cannot guarantee effective performance in different maneuvers. In [8], a car-following structure is developed for AVs based on fuzzy logic. The introduced control mechanism is developed for the AVs on the highways to provide a safe traffic stream. In [9], a fuzzy logic controller is introduced for a mobile robot to follow various trajectories. However, the developed fuzzy logic controller is complicated in that it requires special microprocessing for practical implementation. In [10, 11], neural networks are utilized for path planning to AVs. The availability of the data set for training and testing the neural networks represents a big challenge against the implementation of neural networks. If the data set is not enough and not accurate, the performance of the neural network might not be efficient. On the other side, the MPC is evaluated as a sufficient control technique for different applications including simple and complicated models [12, 13]. In [14], a linear stochastic MPC is proposed for the safety guarantee in some manure and highway simulations showing the effectiveness of the proposed method regarding performance and safety. In [15], there is a comparison between linear and nonlinear MPCs for cruise control in electric vehicles. The performance of both linear and nonlinear MPCs depends on the suitable tuning of their parameters. In the other words, the tracking of regular and irregular reference signals with fewer overshoot, short rise and settling time, and small steady-state error is the main target of the controller which is only met by accurate tuning. These performance criteria are proposed in many studies such as in [16, 17]. However, they limited the trajectory tracking to only step responses. To avoid this, in this paper we propose the minimum error tracking which is efficient for any reference signals (it is not limited to step signals). To tune MPC's parameters, many approaches like [18] present the application of the genetic algorithm to find the optimal weighting of MPC. In reference [19], MPC is supervised by fuzzy logic. The fuzzy logic supervisor fulfills the online tuning of the predictive control parameters. Moreover, neural networks are mostly used for model identification in MPCs. In [20], a novel tuning approach for MPC based on the artificial neural network along with the PSO algorithm is formulated. But this approach is not promisingly able to find the global optimum. Similarly, in [4], a modified neural network is proposed to tune MPC's parameters for robotic manipulators. It is worth noting that throughout the tuning process, computational expenses are a considerable matter. In [21], a new low computational burden design is proposed to adjust the gains of the hybridization between the discrete-time Laguerre function and MPC based on the social ski driver algorithm. This algorithm is developed as a recent artificial intelligence technique. Furthermore, in the concept of robustness of uncertainties, a robust model predictive strategy based on an arithmetic optimization algorithm in [22] tends to handle the uncertainties for automatic voltage regulators [23]. A new robust control strategy is introduced in the paper to design the MPC gains. In [24, 25], another optimal nonlinear MPC is designed based on a new modified multitracker optimization algorithm. The merits of this improved intelligence technique are confirmed to track different linear and nonlinear trajectories and to prevent it from becoming trapped in a local optimum in the process of tuning its parameters. In addition, in [25], a nonfragile PID controller is also proposed to deal with uncertainties while determining the optimal gains of a PID controller.

In this paper, we use the kinematic bicycle model and nonlinear MPC to control the lateral position and speed of AVs. The controller parameters are tuned based on particle swarm optimization. The proposed procedure is more computationally efficient than what have been proposed in the literature such as [17, 18]. Moreover, the simulation result shows the performance of tracking desired trajectories is much better than what was proposed in the mentioned studies.

2. Vehicle Dynamic and Model

Figure 1 depicts the main forces on a car on the road. The vehicle's longitudinal motion, governed by Newton's second law of motion, can be described as follows [26]:

$$\begin{aligned} \sum F &= m\frac{\mathrm{d}v(t)}{\mathrm{d}t}, \\ F_{\text{air}} &= \rho A c_D v^2(t), \\ F_{\text{friction}} &= mgc_{rr}\cos(\theta), \\ F_{\text{roll}} &= mg\sin(\theta), \end{aligned} \tag{1}$$

$$m\frac{\mathrm{d}v(t)}{\mathrm{d}t} = F_{\text{wheel}} - F_{\text{air}} - F_{\text{friction}} - F_{\text{roll}}. \tag{2}$$

where m is the mass of the vehicle, including powertrain inertial effects, $v(t)$ is longitudinal velocity. Here, F_w is chosen to be the sum of tractive or braking force at the wheels, F_g is the road gradient resistance, and F_r is rolling resistance in which c_{rr} is the coefficient of friction and θ is the road slope which is a function of position. In the term representing aerodynamic drag, ρ_a is the air density, A is the vehicle front area, and C_D is the aerodynamic drag coefficient.

The nonlinear continuous time equations that describe the kinematic bicycle model [3] are summarized in (6)–(9). See Figure 2 where x and y are the coordinates of the center of mass in an inertial frame (X, Y), ψ is the inertial heading, and l_f and l_r represent the distance from the center of the mass of the vehicle to the front and rear axles, respectively. β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the car, and the control inputs are the front steering angles δ.

$$\beta(t) = \arctan\left(\frac{l_r}{l_r + l_f}\tan(\delta(t))\right), \tag{3}$$

$$\dot{\psi}(t) = \frac{v(t)}{l_r}\sin(\beta(t)), \tag{4}$$

$$\dot{x}(t) = v(t)\cos(\psi(t) + \beta(t)), \tag{5}$$

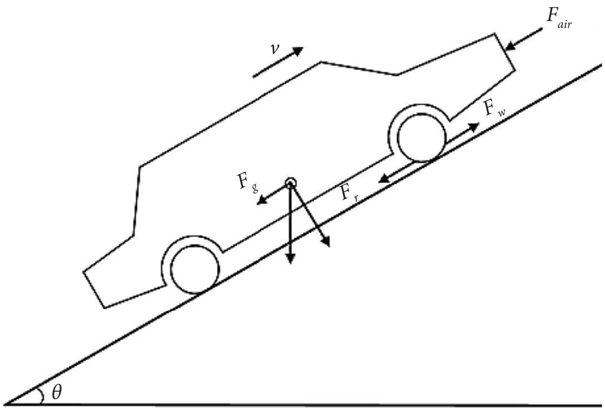

Figure 1: Longitudinal forces acting on the car.

$$\dot{y}(t) = v(t)\sin(\psi(t) + \beta(t)). \tag{6}$$

3. Nonlinear Model Predictive Control

A nonlinear system could be described in the following model:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t)), \tag{7}$$

$$y(t) = h(x(t), d(t)). \tag{8}$$

Here, $x(t)$ is the state space, $f(.)$ is the nonlinear dynamic, $u(t)$ is the input, $d(t)$ denotes the external disturbance on the system, $y(t)$ is the output, and $h(.)$ is the measurement model. MPC uses a model of the system to make predictions about the system's future behavior, and by solving an online optimization algorithm to find the optimal control action, drives the predicted output to the reference. It can also handle input and output constraints. MPC has preview capability; it can incorporate future reference information into the control problem to improve controller performance. MPC predicts the future control signal within finite steps named the control horizon "M" and utilizes the prediction of the system output within finite steps named the prediction horizon "P" to predict the proper control moves. The best control moves are selected according to the decreasing of a quadratic objective function. This objective function is formulated in the following term:

$$J = \sum_{k=1}^{P} \left\| y^{\mathrm{des}}(t+k) - y(t+k|k) \right\|_{Q} + \gamma \sum_{k=0}^{M-1} \| \Delta u(t+k|k) \|_{R}. \tag{9}$$

Subject to the system dynamic (7) and (8) and the following inequalities:

$$\Delta u_{\min} < \Delta u(k) < \Delta u_{\max}. \tag{10}$$

$$u_{\min} < u(k) < u_{\max}. \tag{11}$$

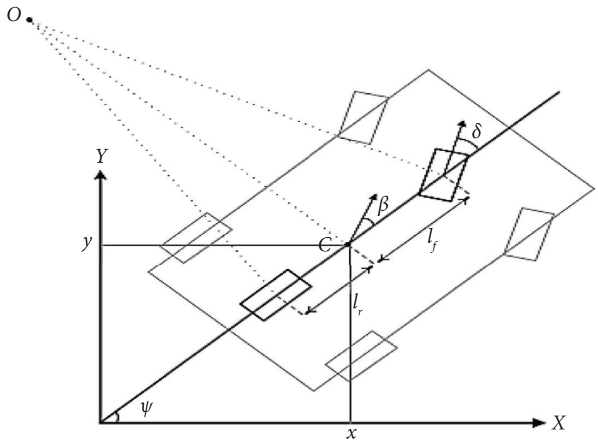

Figure 2: Kinematic bicycle model.

Here, $y^{\mathrm{des}}(t)$ is the desired reference, $\Delta u(k) = u(k) - u(k-1)$, $\|.\|_Q, \|.\|_R$ are the Euclidian norm, and γ is the input-output ratio weighting factors. In (9), the first term penalizes the output reference tracking, and the second one penalizes the slope of changes in the input signal.

The constraints (10) and (11) are considered for both longitudinal speed and lateral position. These make the algorithm much more liable to be implemented in reality. As the signal control of longitudinal velocity is supposed to be the motor force which obliviously cannot be overvalued or fluctuated in high frequencies because of mechanical constraints in the vehicle's motor. For controlling the lateral position, in addition to mechanical constraints, the steering angle should be intentionally limited for the sake of safety driving, especially in high speeds. Nevertheless, if the algorithm suggests an over or under-rated signal control, it is considered as a lower or upper limited band, respectively.

The extended dynamic model predictive controller (EDMC) is based on a nonlinear model and a simultaneous linear model. During computing control signals, EDMC tries to reduce the difference between nonlinear and simultaneous linear models, doing so based on step responses. In a linear system, the step response of a system is computed by the convolution series.

$$y(t) = \sum_{i=1}^{\infty} g(i) \Delta u(t-i). \tag{12}$$

For the P step of output prediction and M steps of horizon control, this could be shown in the following matrices. Here, Y_m is the simultaneous linear model output, $G_{P\times M}$ is the Toeplitz matrix of step responses which are driven by feeding a step signal to the nonlinear dynamic. For computing the P elements of $Y_{\mathrm{past}}(t+1)$, it does not need to compute all elements of step response and it can be computed by evaluating $\Delta u(i) = 0$, $i = t, t+1, \ldots.$ in the nonlinear dynamic.

$$\begin{bmatrix} y_m(t+1|t) \\ y_m(t+2|t) \\ y_m(t+3|t) \\ \vdots \\ y_m(t+P|t) \end{bmatrix}_{P\times 1} = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & g_{P+M-2} \\ g_P & g_{P-1} & g_{P-2} & \cdots & g_{P+M-1} \end{bmatrix}_{P\times M} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \\ \vdots \\ \Delta u(t+M-1) \end{bmatrix}_{M\times 1} + \begin{bmatrix} g_2\Delta u(t-1) + g_3\Delta u(t-2) + \ldots \\ g_3\Delta u(t-1) + g_4\Delta u(t-2) + \ldots \\ g_4\Delta u(t-1) + g_4\Delta u(t-2) + \ldots \\ \vdots \\ g_{P+1}\Delta u(t-1) + g_{P+2}\Delta u(t-2) + \ldots \end{bmatrix}_{P\times 1}, \quad (13)$$

$$Y_m(t+1) = G\Delta U(t) + Y_{\text{past}}(t+1). \quad (14)$$

By assuming $Q = I_{P\times P}$ and $R = I_{M\times M}$, the matrix form of the objective function described in (9) can be rewritten in the following form:

$$J_1 = \left(Y^{\text{des}} - Y\right)^T \left(Y^{\text{des}} - Y\right) + \gamma \Delta U^T \Delta U, \quad (15)$$

where $Y = Y_m + D$; by defining $E = Y^{\text{des}} - Y_m - D$, the signal control is

$$\Delta U(t) = \left(G^T G + \gamma I\right)^{-1} G^T E. \quad (16)$$

Moreover, the disturbance term is composed of nonlinear disturbance and measurement disturbance, shown by D_{nl} and D_{ext}, respectively.

$$D = D_{nl} + D_{\text{ext}}. \quad (17)$$

For the external disturbance, the best approach is to assume this disturbance does not change in the near future and equal to the difference between the measurement output and the nonlinear mode's output.

$$D_{\text{ext}} = \left(y(t) - y_{nl}(t)\right) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{P\times 1}, \quad (18)$$

$$D_{nl} = \left(Y_{nl}(t+1) - Y_m(t+1)\right)_{P\times 1}. \quad (19)$$

Since $Y_{nl}(t+1)$ and $Y_m(t+1)$ are functions of $\Delta U(t)$, in the main loop of the algorithm, there is a fixed point iterative approach to solve the following equations:

$$D_{nl} - \left(Y_{nl}(t+1) - Y_m(t+1)\right) = 0, \quad (20)$$

$$D_{nl}^{l+1} = D_{nl}^{l} + \alpha\left(Y_{nl}(t+1) - Y_m(t+1)\right). \quad (21)$$

4. Optimization Problem

Particle swarm optimization (PSO) is a bio-inspired algorithm that optimizes an objective function by iteratively trying to improve a candidate solution with regard to a given measure of quality in the solution space. It is different from other optimization algorithms in such a way that only the objective function is needed and it is not dependent on the gradient or any differential form of the objective. This method was originally proposed in [27] to find the maximum or minimum of a function defined on a multidimensional vector space. In this paper for the NLMPC, the mean square error (MSE) objective function is suggested to find the optimal parameters: prediction horizon (P), control horizon (M), and weighting factor (γ).

$$J_2 = \frac{1}{(T - t_0)} \sum_{t=t0}^{T} \left(y^{\text{des}}(t) - y(t)\right), \quad (22)$$

where t_0 and T is the start and end time of desired pattern, respectively. What was developed in references [17, 18] is not only limited to step reference signal but also is rather computationally expensive than what is proposed in this paper. However, the objective function J_2 (22) is quite efficient for any desired pattern. In addition, the objective function J_1 (15) has only three variables to the optimization problem.

Following would be the algorithm of tuning EDMC using PSO. Assume there are N particles by which the position of each particle is denoted by $X^i(j) = [P^i(j), M^i(j), \gamma^i(j)]$. Beside the position, there is a velocity for each particle, denoted by $V^i(j)$. In the iteration of j, the position and velocity of particles in the next iteration are updated by the following rules:

$$\begin{aligned} V^i(j+1) &= wV^i(j) + c_1\left(X^i_{p_{\text{best}}} - X^i(j)\right) + c_2\left(X_{g_{\text{best}}} - X^i(j)\right), \\ X^i(j+1) &= X^i(j) + V^i(j+1). \end{aligned} \quad (23)$$

The constant w is the inertia weight. It determines how much should the particle keep on with its previous velocity. The constants c_1 and c_2 are called the cognitive and the social coefficients, respectively. They control how much weight should be given between refining the search result of the particle itself and recognizing the search result of the swarm. $X^i_{p_{\text{best}}}$ is the position that gives the minimum J_2 which is ever explored by particle i, and $X_{g_{\text{best}}}$ is that explored by all the particles in the swarm.

The proposed algorithm of EDMC along with PSO can be expressed in Algorithm 1 and Figure 3.

5. Simulations and Results

In this section, the performance of the proposed optimization paradigm, in comparison with the approaches in [17, 18] is shown. Moreover, the robustness of the EDMC in the case of disturbance rejection and the uncertainties in the model's parameters is illustrated. To begin with, in the PSO-MPC approach in [17] and GA-MPC in [18], the

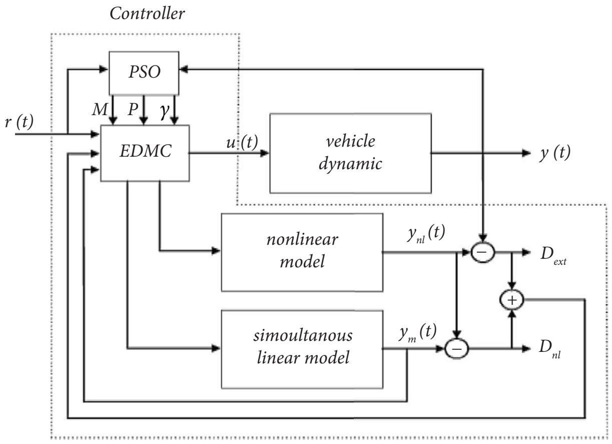

Figure 3: Scheme of EDMC tuned by PSO.

```
Given the patterns of $y^{\text{des}}$ for both lateral position and longitudinal velocity
PSO loop for finding optimal $P$, $M$ and $\gamma$
For $t = t_0, t_1, \ldots, T$
  Compute the step responses by evaluating $\Delta u(i) = 0$, $i = t_0, t_0 + 1, \ldots.$ in the nonlinear dynamics, using equations (5)–(8).
  Evaluating Toeplitz and $Y_{\text{past}}$ matrices, using (13) and (14).
  Evaluating $D_{\text{ext}}$, using (18)
  Loop for computing $\Delta u$ and $D_{nl}$
    Initialing $D_{nl} = 0$
    Computing $\Delta u$, using (16)
    Update $D_{nl}$, using (21)
  End
  Computing input signals: $u(t) = u(t-1) + \Delta u(t)$
End
```

Algorithm 1: The pseudocode of the main steps of the proposed paradigm.

weighting factors of the control horizon and prediction horizon were supposed to be different in each step. As a result, the PSO algorism has to search in the higher dimensional space, which makes the algorithm computationally expensive, particularly while the long horizon of control and prediction are needed. However, proposed objective function (15), only has to have three parameters to tune.

The nominal parameters of the vehicle dynamic are shown in Table 1 [28]. And, the suggested EDMC parameters (M, P, γ) are tuned by the PSO algorithm for each desired output pattern.

The performance of the auto-tuned MPC for the control of speed and lateral position is shown in Figures 4 and 5, respectively. The response of the PSO-MPC algorithm, alongside the results of methods presented in [17, 18] ($M = 2$ and $P = 20$ for speed control and $M = 2$ and $P = 15$ for lateral position control), is shown. While the objective function in the PSO-MPC is defined to reduce overshoot, steady-state error, and rise and settling time, the objective function in this paper (MSE) represents the response performance more efficiently.

The results of the comparison are summarized in Tables 2 and 3 for the vehicle speed and lateral position, respectively.

Table 1: The vehicle parameters.

m	2020 kg
ρ	1.184 kg/m^3
A	2.1 m^2
C_D	0.33
C_{rr}	0.0015
l_f	1.4 m
l_r	1.65 m
g	9.81 m/s^2

Tables 2 and 3 show that the proposed method has a much better performance compared to the other two methods. Note that the sampling time for the speed controller is $T_s = 0.1$ sec and the PSO algorithm would tune the parameters at = 2, $P = 32$, and $\gamma = 2.1e - 4$. However, $T_s = 0.01$ sec is chosen for the lateral position controller at the constant speed of 30 m/sec and the parameters are tuned at = 2, $P = 21$, and $\gamma = 23$. The results show that it can provide a good tracking performance and is more computationally effective rather than conventional MPCs. It is worth noting that there is no limitation to simultaneously implementing both speed and lateral position controllers. Another merit of the proposed approach is the robustness in the case of uncertainties in the model parameters and external

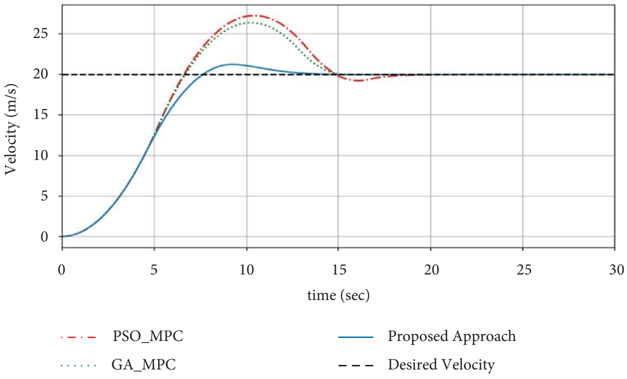

Figure 4: Speed control.

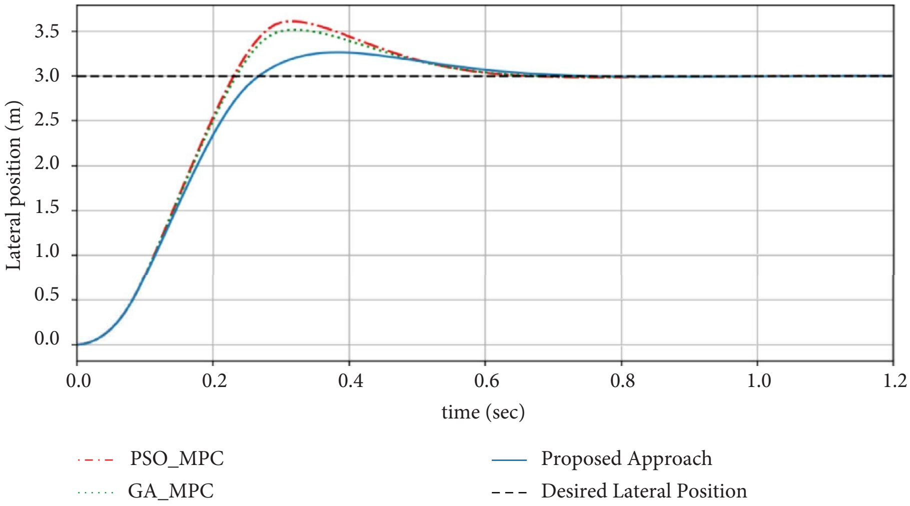

Figure 5: Lateral position control.

Table 2: The response characteristics of vehicle’s speed.

	Proposed approach	GA-MPC	PSO-MPC
Number of tuned parameters	3	22	22
RMSE	6.77	7.10	7.24
Overshoot	6.16%	31.84	36.28%
Steady-state error	0.00	0.00	0.00
Rise time	4.70 sec	4.2 sec	4.20 sec
Settling time	11.20 sec	14.50 sec	17.20 sec

Table 3: The response characteristics of vehicle’s lateral position.

	Proposed approach	GA_MPC	PSO-MPC
Number of tuned parameters	3	17	17
RMSE	0.72	0.73	0.73
Overshoot	8.01%	17.13	20.27%
Steady-state error	0.00	0.00	0.00
Rise time	0.16 sec	0.15 sec	0.14 sec
Settling time	0.62 sec	0.58 sec	0.58 sec

disturbances. Figures 6–9 show how the controller deals with uncertainty in the parameters in C_D, C_{rr}, l_r, l_f, comparing to the nominal value (Table 1).

$$\begin{aligned} l_{r_{\text{devitated}}} &= 0.5 l_r, \\ l_{f_{\text{devitated}}} &= 1.5 l_r, \\ C_{rr_{\text{devitated}}} &= 1.5 C_{rr}, \\ C_{D_{\text{devitated}}} &= 0.5 C_D. \end{aligned} \tag{24}$$

It is crystal clear that the proposed paradigm can control the lateral position and speed of an AV at a high range of performance. In comparison with the previous studies [17, 18], the proposed algorithm not only results in better performance but also takes less time to tune the controller’s parameters. The details are shown in Tables 2 and 3. Moreover, the robustness of the controller shows it is not sensitive to accurate identification of the nonlinear model’s parameters. Figure 10 illustrates the performance of the tracking nonlinear reference signal in the presence of fluctuating slope of the road.

The result shows the robustness of the proposed approach in no stationary environment.

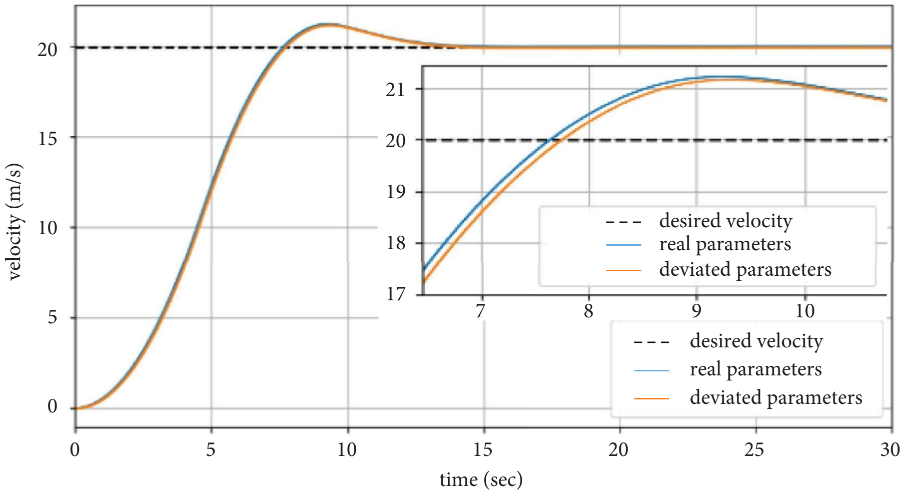

Figure 6: Tracking desired velocity in the presence of uncertainties in the model parameters.

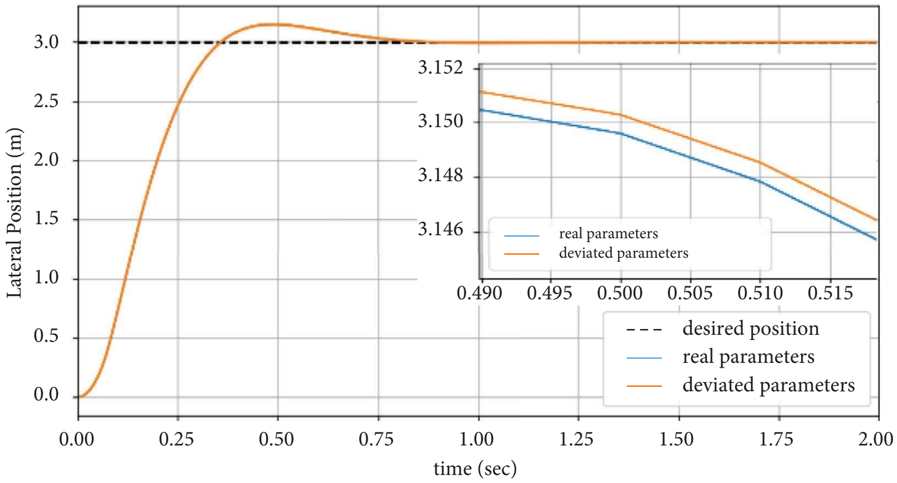

Figure 7: Tracking desired lateral position in the presence of uncertainties in the model parameters.

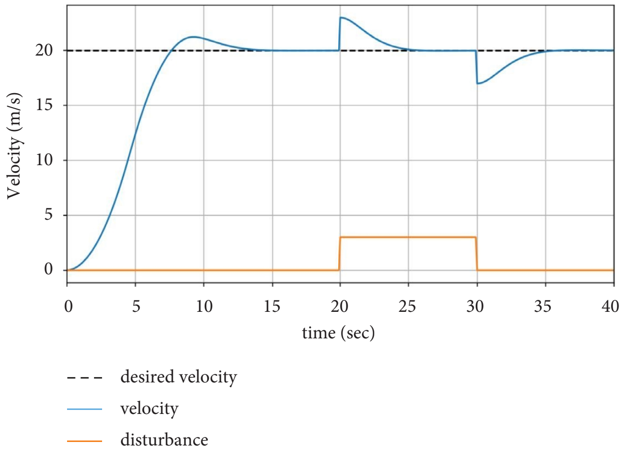

Figure 8: Tracking desired velocity in the presence of external disturbance.

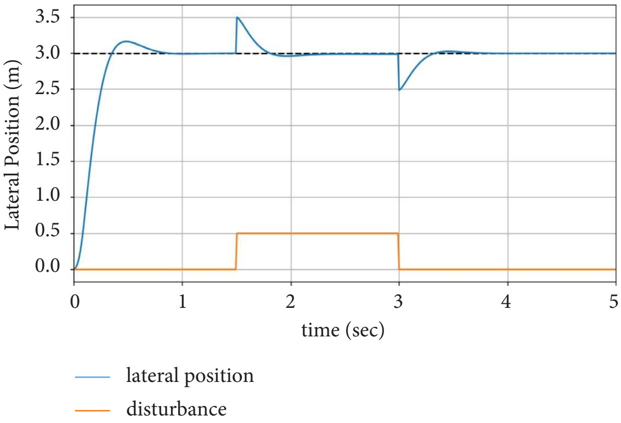

Figure 9: Tracking desired lateral position in the presence of external disturbance.

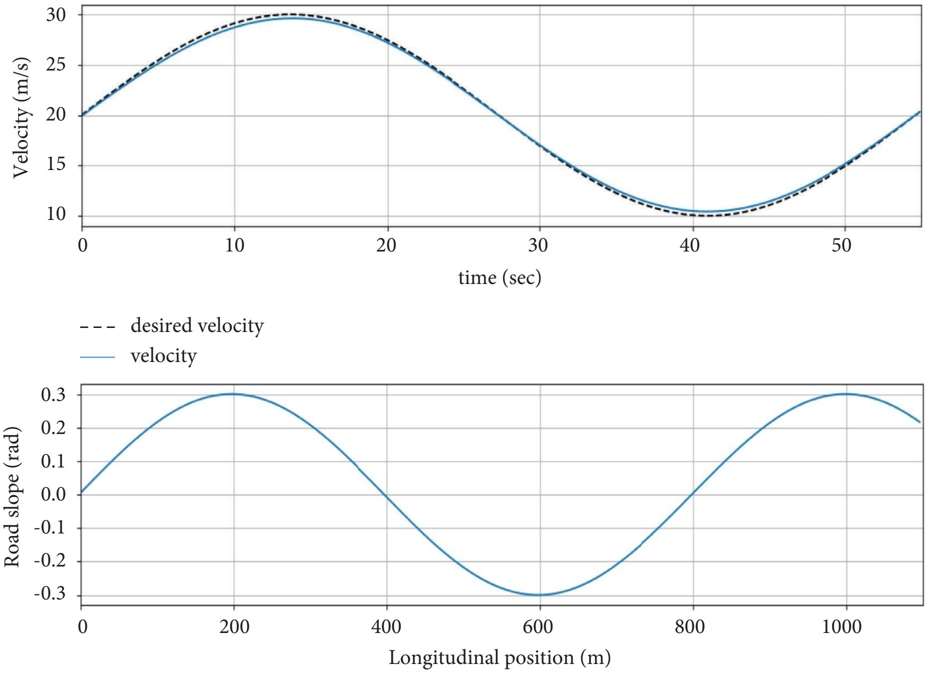

Figure 10: Tracking nonconstant desired velocity on a nonconstant slope road.

6. Conclusions

For the formation control of the lateral position and speed of an AV, this paper first modeled the dynamic of a four-wheel vehicle by applying the kinematic bicycle model. Afterward, the study formulated the EDMC algorithm. Finally, the PSO algorithm is devoted to finding the optimal solution by tuning the controller's parameters. The output results emphasize that the proposed method is superior to the other methods and it is more effective to track regular and irregular trajectories with a desired mean square error, short settling time, and less overshoot. The simulation results show the proposed approach desirably meets the performance criteria. Moreover, the suggested technique is robust versus the system parameters uncertainties. For future work, the reference signals could be extracted from recently developed control strategies such as reinforcement learning. Furthermore, regarding collision avoidance and safe driving, the motion of other vehicles should be taken into account [24].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Srinivasan, I. Sa, A. Zyner, V. Reijgwart, M. I. Valls, and R. Siegwart, "End-to-end velocity estimation for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6869–6875, 2020.

[2] L. Fridman, D. E. Brown, M. Glazer et al., "Mit autonomous vehicle technology study: large-scale deep learning based analysis of driver behavior and interaction with automation," 2017, https://arxiv.org/abs/1711.06976.

[3] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Seoul, Republic of Korea, July 2015.

[4] K. Macek, K. Thoma, R. Glatzel, and R. Siegwart, "Dynamics modeling and parameter identification for autonomous vehicle navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, November 2007.

[5] M. Elsisi, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Effective nonlinear model predictive control scheme tuned by improved NN for robotic manipulators," *IEEE Access*, vol. 9, Article ID 64278, 2021.

[6] L. Nie, J. Guan, C. Lu, H. Zheng, and Z. Yin, "Longitudinal speed control of autonomous vehicle based on a self-adaptive PID of radial basis function neural network," *IET Intelligent Transport Systems*, vol. 12, no. 6, pp. 485–494, 2018.

[7] X. Han, X. Zhang, Y. Du, and G. Cheng, "Design of autonomous vehicle controller based on BP-PID," in *Proceedings of the IOP Conference Series: Earth and Environmental Science*, Hubei, China, December 2018.

[8] L. González, E. Martí, I. Calvo, A. Ruiz, and J. Pérez, "Towards risk estimation in automated vehicles using fuzzy logic," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, WAISE, Västerås, Sweden, September 2018.

[9] X. Xiang, C. Yu, and Q. Zhang, "Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties," *Computers & Operations Research*, vol. 84, pp. 165–177, 2017.

[10] S. Inkyung, C. Bongjun, and N. Peter, "On the training of a neural network for online path planning with offline path

planning algorithms," *International Journal of Information Management*, vol. 57, 2020.

[11] J. Asgari, F. Borrelli, and H. Eric Tseng, "Discussion on hybrid parameter-varying model predictive control for autonomous vehicle steering," *European Journal of Control*, vol. 14, no. 5, pp. 432–434, 2008.

[12] X. Du, K. K. K. Htet, and K. K. Tan, "Development of a genetic-algorithm-based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 6970–6977, 2016.

[13] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, pp. 3363–3370, 2019.

[14] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 1, pp. 1–14, 2022.

[15] Y. Jia, R. Jibrin, and D. Görges, "Energy-optimal adaptive cruise control for electric vehicles based on linear and nonlinear model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14173–14187, 2020.

[16] G. Wang, c Tzueng, and K. Chang, "Neural-network-based self-tuning PI controller for precise motion control of PMAC motors," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 2, pp. 408–415, 2001.

[17] R. Suzuki, F. Kawai, H. Ito, C. Nakazawa, Y. Fukuyama, and E. Aiyoshi, "Automatic tuning of model predictive control using particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, Honolulu, HI, USA, April 2007.

[18] E. Joelianto and F. Hernawan, "Multiplexed model predictive control weighting selection using genetic algorithm," in *Proceedings of the 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering*, pp. 23–25, Bandung, Indonesia, November 2009.

[19] J. Mamboundou and N. Langlois, "Indirect adaptive model predictive control supervised by fuzzy logic," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, June 2011.

[20] Y. Shin, R. Smith, and S. Hwang, "Development of model predictive control system using an artificial neural network: a case study with a distillation column," *Journal of Cleaner Production*, vol. 277, Article ID 124124, 2020.

[21] H. Moumouh, N. Langlois, and M. Haddad, "A novel tuning approach for MPC parameters based on artificial neural network," in *Proceedings of the IEEE 15th International Conference on Control and Automation*, pp. 16–19, (ICCA), Edinburgh, UK, July 2019.

[22] M. Elsisi and M. A. Ebrahim, "Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics," *International Journal of Intelligent Systems*, vol. 36, no. 11, pp. 6968–6987, 2021.

[23] M. Elsisi, M. Q. Tran, H. M. Hasanien, R. A. Turky, F. Albalawi, and S. S. M. Ghoneim, "Robust model predictive control paradigm for automatic voltage regulators against uncertainty based on optimization algorithms," *Mathematics*, vol. 9, no. 22, p. 2885, 2021.

[24] M. Elsisi, "Optimal design of nonlinear model predictive controller based on new modified multitracker optimization algorithm," *International Journal of Intelligent Systems*, vol. 35, no. 11, pp. 1857–1878, 2020.

[25] M. Elsisi, "Optimal design of non-fragile PID controller," *Asian Journal of Control*, vol. 23, no. 2, pp. 729–738, 2021.

[26] J. Han, A. Vahidi, and A. Sciarretta, "Fundamentals of energy efficient driving for combustion engine and electric vehicles: an optimal control perspective," *Automatica*, vol. 103, pp. 558–572, 2019.

[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95 - International Conference on Neural Networks*, Perth, Australia, December 1995.

[28] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path Planning and tracking for vehicle collision avoidance based on model predictive control with multi-constraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2017.