Path Tracking Method Based on Model Predictive Control and Genetic Algorithm for Autonomous Vehicle

Meng Wang, 1 Juexuan Chen, 1 Hanwen Yang, 2 Xin Wu, 1 and Longjiao Ye 1

1 Wuhan Goyu Intelligent Technology Co., Ltd, Wuhan 430200, China
2 Shanghai JiaoTong University, Shanghai 200240, China

Correspondence should be addressed to Juexuan Chen; juexuanchen@hotmail.com

Received 8 June 2022; Revised 22 August 2022; Accepted 29 August 2022; Published 20 September 2022

Model predictive control (MPC) is often used for controlling the autonomous vehicle tracking the target path. But to apply MPC schemes, the nonlinear model of vehicle kinematics needs to be approximately converted to a linear format, and the path tracking problem has to be converted into certain formats in order to implement the solver of convex quadratic programming. To solve these issues, a control strategy combining MPC and genetic algorithm (GA) is put forward. The nonlinear predictive model is adopted to predict the future movement of a controlled vehicle. The objective function is established according to the future movement and target path. Instead of using a convex quadratic programming solver, GA is applied to solve the optimization problem. The proposed MPC-GA method can handle the arbitrary nonlinear problem and make the objective function more comprehensible and flexible. This method is applied in solving the path tracking problem of an autonomous vehicle. Both simulations and on-field tests are conducted. The results validate the efficiency of the proposed MPC-GA path tracking method in comparison with traditional methods. With the MPC-GA controller, the automatic driving on the park road is basically realized. The control strategy can be considered as an alternative method to solve the path tracking problem for an autonomous vehicle.

1. Introduction

With the rapid development of artificial intelligence, autonomous vehicle arouses more and more scholars’ interest [1]. Generally speaking, the autonomous vehicle can be divided into three modules: perception, decision and planning, and trajectory control. The trajectory control contains lateral control (path tracking) and longitudinal control. The aim of path tracking is to track the target trajectory in a fast and exact way under the limit of operation conditions and restrictions [2, 3]. As one of the most important parts, path tracking control has been a significant research topic.

Traditional control methods, such as PID [4] and fuzzy control [5], are widely used in path tracking control or steering control. They have the characteristic of simple structure and quick response. Because PID control and fuzzy control generate the steering actions exclusively based on the tracking errors without requiring the vehicle information, it is unnecessary to set up an explicit mathematical model of the controlled vehicle. Single PID controller or single fuzzy controller is poor of precision, so some scholars propose neural network PID control method [6, 7] and fuzzy-PID control method [8, 9]. The type of composite control system improves the tracking precision to some extent. However, traditional control methods fail to consider the actuator saturation and physical limit [10] and only take into account the current deviation between the actual state and ideal state, instead of future deviation. What is more, traditional methods generally require a labor-intensive tuning task to achieve acceptable path tracking performance.

Pure Pursuit (PP) algorithm is another simple approach for path tracking. Due to the simplicity of its math model, the PP algorithm is chosen as the main algorithm for path controlling of Ackermann steering vehicles [11, 12]. PP, as a geometric control algorithm, determines the look-ahead point according to the look-ahead distance, then obtains the vehicle steering angle control input according to the current
heading angle and the distance between the current position and the look-ahead point position, so that the vehicle follows an arc trajectory passing through the look-ahead point [13]. PP control method is also applied for the path tracking problem of four-wheel independent steering robot [14]. The path tracking control is realized considering the correlation between the look-ahead distance and velocity, as well as the lateral error between the robot and the reference path.

In recent years, some new control theories and applications have been extensively developed, such as model predictive control [15], model-free control (MFC) [16], and static output feedback control (SOF) [17]. More and more scholars put forward the MPC into autonomous vehicle [18, 19]. MPC controller design is based on an identified model from experimental data in open loop. The model is represented in state-space form, and the future values of the state trajectory and output variables are used as optimization objective with respect to the future values of the control signal [20]. The MPC objective is a finite time quadratic cost function accounting for output error and control amplitude. The goal of MPC is to compute a future control sequence in a defined horizon that can drive the vehicle close to the reference path. This is accomplished by minimizing a multi-stage cost function with respect to the future control actions, considering a set of constraints both in the control actions and the plant outputs [21]. Different methods, such as PP, Stanley, Linear Quadratic Regulator (LQR), and MPC with Ackermann steering model are investigated [22], and these methods are tested on different shape paths in simulation experiments. It is demonstrated that the performance of the MPC controller is better than those using Pure Pursuit and Stanley controllers.

In fact, vehicle motion is a kind of nonlinear time-varying system. In most academic studies [23, 24], the nonlinear time-varying model is approximately transformed into a linear time-invariant model. The purpose of this approximate treatment is to facilitate calculations. However, those ignored nonlinear factor and time variation factor will induce unacceptable errors. At the same time, the solving process of MPC with restrictions is often changed into a constrained quadratic programming problem. Solving this quadratic programming problem remains a difficult task, and only a few results have been presented in the literature [25, 26]. Taking into account the above facts, this study proposes an approach for introducing GA into the MPC process. On one hand, the nonlinear time-varying model of vehicle motion is used directly to predict future values of the state trajectory and output variables, instead of approximate treatment. With accurately predicting future values, the objective function value is obtained by comparing the predicted values and reference values. On the other hand, GA is introduced to search for the optimal solution. GA is an adaptive global optimization probabilistic search algorithm that is inspired by referring to natural selection and evolution mechanisms in the biological world. Using a genetic algorithm, the global optimal control signal can be obtained, instead of solving the quadratic programming problem. The scheme to implement GA is much simpler than traditional MPC. To implement the proposed MPC-GA, random solutions are generated according to the encoding principle of GA first. With the given solutions, a nonlinear time-varying model is used to forecast future travel paths. Then, the forecast paths are evaluated according to their deviation from the reference path. After the evolution processing including selection, mutation, and crossover operations, the optimal solution will be obtained. Lastly, the optimal solution is converted to an executable control quantity.

The remainder of this study is organized as follows: Section 2 describes the basic mathematical models about vehicle motion and traditional MPC. Section 3 introduces the proposed MPC-GA method. The specific application scheme of MPC-GA is described in Section 4. At last, Section 5 displays the simulation and experiment results. This study ends with the conclusions of this work in Section 6.

2. Basic Mathematical Model

The mathematical model of vehicle motion involves a kinematics aspect and a dynamic aspect. Generally, the kinematic model is suitable for control law design at a low speed [27]. In this study, we focus on autonomous technology in defined areas, such as the industrial park, tourism areas, and campus roads. So, the allowed vehicle speed is usually limited within 40 km/h, and only the kinematic model is taken into account. In fact, the vehicle kinematics model is a nonlinear system. The mathematical model has to be transformed into a linear format. Then, linear MPC can be chosen for implementation.

2.1. Vehicle Kinematics Model. A simplified bicycle kinematics model is used to describe vehicle motion at low speed. For the sake of simplified analysis, it is assumed that the machine moves at a low constant speed over a flat surface with no wheel slippage and the wheels are considered rigid wheels [28]. Usually, the automotive rear axle center is regarded as the reference point, and the control task is to make the center point follow the target path. According to the kinematics analysis, the vehicle’s motion equation is given as [29, 30]:

$$\begin{cases}
\dot{x} = v \cos \theta, \\
\dot{y} = v \sin \theta, \\
\dot{\theta} = v \tan \left( \frac{\delta}{L} \right),
\end{cases}$$

where $(x, y)$ is displacement in the global reference frame coordinates as illustrated in Figure 1, $\theta$ is the heading angle of the vehicle, $v$ is the longitudinal velocity at the middle of the rear axis, $L$ is the wheel base, and $\delta$ is the steering angle of the front wheel.

The vehicle kinematics model can be expressed as a nonlinear equation shown in (2). The state variable is $X(t) = [x(t) \ y(t) \ \theta(t)]^T$, and the control amount is $U(t) = [v(t) \ \delta(t)]^T$. In practical application, as the vehicle speed changes slightly
in a short period of time, the speed control variable $v$ is considered approximately constant.

$$\dot{X}(t) = f(X(t), U(t)). \quad (2)$$

2.2. Linear Predictive Model. Because the discrete state-space model is generally adopted for the controlled system in MPC, the nonlinear continuous state model of the vehicle should be linearized and discretized [31]. Ignoring higher-order derivatives, Taylor expansion is performed at an arbitrary reference point $(X_r(t), U_r(t))$ with a nonlinear state equation:

$$X(t) = f(X_r(t), U_r(t)) + \frac{\partial f}{\partial x}\bigg|_{X=X_r, U=U_r} (X(t), X_r(t)) + \frac{\partial f}{\partial U}\bigg|_{X=X_r, U=U_r} (U(t), U_r(t)). \quad (3)$$

where $\frac{\partial f}{\partial x}\bigg|_{X=X_r, U=U_r}$ and $\frac{\partial f}{\partial U}\bigg|_{X=X_r, U=U_r}$ represent the Jacobian matrix of the path tracking system with respect to the state quantity $X(t)$ and the control quantity $U(t)$ respectively, $X_r(t)$ represents the reference state with $X_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$, and $U_r(t)$ represents the reference control amount with $U_r = [v_r(t), \delta_r(t)]^T$.

Combining (2) and (3), the linear model (4), namely the error equation of the vehicle, can be obtained. The linear model (4) describes the dynamic characteristics of error signals. $\Delta X$ represents the error signal with $\Delta X = X(t) - X_r(t)$, and $\Delta U(t)$ denotes the control input error with $\Delta U(t) = U(t) - U_r(t)$. At the same time, matrices $A(t)$ and $B(t)$ are the Jacobians of system in relation to $X(t)$ and $U(t)$ respectively:

$$\Delta \dot{X}(t) = A(t) \Delta X(t) + B(t) \Delta U(t). \quad (4)$$

Using Euler’s method, a time-variant discrete linear model is obtained as (5), and equation (6) is the corresponding output function.

$$\Delta X(k + 1) = \Delta A(k) \Delta X(k) + \Delta B \Delta U(k), \quad (5)$$

$$\Delta Y(k + 1) = \Delta C(k) \Delta X(k + 1). \quad (6)$$

Further, linear MPC constructs the state quantity $X_\delta(k) = [\Delta X(k) \Delta U(k-1)]^T$ and control increment $U_\delta(k) = [\Delta U(k) \Delta U(k-1)]^T$, so as to obtain the linear predictive model and output model as (7) and (8).

$$X_\delta(k + 1) = A_\delta(k) X_\delta(k) + B_\delta(k) U_\delta(k), \quad (7)$$

$$Y_\delta(k + 1) = C_\delta(k) X_\delta(k + 1). \quad (8)$$

The prediction horizon and control horizon of the controller based on MPC are set to $N_p$ and $N_c$, then the outputs over the prediction horizon $N_p$ can be derived according to (9):

$$\begin{bmatrix}
Y_\delta(k + 1| k) \\
Y_\delta(k + 2| k) \\
\vdots \\
Y_\delta(k + N_p| k) \\
Y_\delta(k + N_p + 1| k) \\
\vdots \\
Y_\delta(k + N_p + N_c| k)
\end{bmatrix} =
\begin{bmatrix}
C_\delta A_\delta \\
C_\delta A_\delta B_\delta \\
\vdots \\
C_\delta A_\delta^{N_p-1} B_\delta \\
C_\delta A_\delta^{N_p-2} B_\delta \\
\vdots \\
C_\delta A_\delta^{N_p+N_c-2} B_\delta \\
C_\delta A_\delta^{N_p+N_c-1} B_\delta \\
\end{bmatrix}
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
U_\delta(k| k) \\
\vdots \\
U_\delta(k + N_c - 1| k)
\end{bmatrix}. \quad (9)$$

2.3. Objective Function and Optimization Problem. The optimization problem can be understood as follows: to find the control inputs in a bounded group of possible values that drive the state variables from an initial position to the origin. Therefore, an objective function is designed, which regards the control quantity $U_\delta$ as the variable. To ensure tracking the target signal quickly and smoothly, the objective function
is always designed as (10). The first term of the formula represents weighting error between the predicted value and the target signal. It reflects the path tracking performance. The second term considers the volatility of control quantities, which is to ensure the control quantities change smoothly. Coefficients \( q_i \) and \( r_i \) represent the weight respectively. At the same time, some constraint conditions are given as (11):

\[
\min J = \sum_{i=1}^{N_p} \| Y_g(K + i|k) \|_Q + \sum_{j=0}^{N_c-1} \| \Delta U_g(K + j|k) \|_K^2, \text{s.t.} \Delta U_{\min} \leq \Delta U \leq \Delta U_{\max}. 
\]

The above optimization problem is a typical convex quadratic programming problem with linear constraints. Some previous works [32, 33] have provided the methods to solve this kind of problem, and we will not repeat them here.

When applying MPC schemes, the nonlinear model of vehicle kinematics needs to be converted to a linear format, and the path tracking problem has to be converted into certain formats in order to implement the solver of convex quadratic programming. However, those ignored nonlinear factor and time variation factor will induce unacceptable errors. Also, solving this quadratic programming problem remains a difficult task.

3. Control Method Based on MPC-GA

Considering that GA is an adaptive heuristic search technique inspired by the evolutionary ideas of natural selection and genetics [34, 35], this section constructs a new control method that combines MPC and GA. Compared with traditional MPC, it is no need to convert the nonlinear model to a linear format or implement the solver of convex quadratic programming.

MPC-GA uses a population of points at a time in contrast to the single-point approach by traditional MPC methods. The convergence is monotonic and a global solution is obtained. The solving process of the MPC-GA method can be generally divided into three steps. First of all, a group of random solutions are initialized complying with GA encoding rules. The solution group is considered as the initial population and each solution is considered as an individual. Next, MPC provides the prediction model, and the predictive model is utilized to predict the state variables and output variables at each time step. The predictive variables will be compared with the corresponding reference variables, which is to construct the objective function. The objective function will be used for calculating the fitness value that is used in the selection process of GA. Lastly, selection, mutation, and crossover operations on the genes of individuals are conducted based on the principles of GA, and iterative operation is implemented in order to search optimal individual or solution.

3.1. Control Structure of MPC-GA. The control system diagram is shown in Figure 2. Vehicle sensors provide all measured variables that contain the current vehicle position. According to the current position displayed on the global map, the local reference input is generated. The reference input \( Y_r \) is in fact a path segment that starts from the current position and ends at a certain distance ahead. The sensors also provide the measured variables that constitute initial state \( X_0 \). With the MPC-GA controller, a control quantity will be calculated and output to the controlled object.

The working principle of MPC-GA controller: in each control cycle, initial solutions \( \{ U_k \} \) are generated randomly. Given the initial candidate solutions and the current state of controlled object, the corresponding predicted outputs \( \{ Y_p \} \) are obtained according to the nonlinear predictive model. Through the iterative optimization of GA, the final solution \( U(t) \) that minimizes the deviation between the reference signal \( Y_r \) and predicted output \( \{ Y_p \} \) is searched. Lastly, the optimal solution is applied to the controlled object, and the control task of the current control cycle is finished. With the passage of time, the output of controlled object gradually approximates the desired signal. Figure 3 shows the difference between MPC-GA and traditional MPC.

According to Figure 3, MPC-GA is based on the principle and implementation steps of traditional MPC. The main improvements are: (1) MPC-GA directly uses the nonlinear time-varying model of vehicle as the predictive model to predict the states and outputs, instead of using an approximate predictive model. Therefore, the predicted states and outputs are more accurate. (2) MPC-GA is able to deal with the free-form objective function, including the quadratic objective function. (3) Given the objective function, MPC-GA utilizes the genetic algorithm to solve the optimization goal, instead of using other quadratic programming solvers. Based on MPC and classical GA schemes, the specific steps are described in detail in the following:

3.2. Initial Solution. First of all, it is need to determine the solution space of the problem based on the description of the optimization problem [36]. The design of the initial solution is an important aspect, since the initial solution relates to the
convergence speed to find an optimal solution. Meanwhile, the initial solution must be feasible, so as to satisfy the constraints. Aiming at these requirements, each candidate solution \( U \) is defined as a sequence of \( N_C \) discrete values as:

\[
U = \{u_1, u_2, \ldots, u_i, \ldots, u_{N_C}\}, \quad i = 1, 2, \ldots, N_c.
\]  

For each control variable \( u_i \), the physical limit is a basic constraint, which means the control quantity of each cycle should not exceed the maximum output of the actuator (maximum wheel angle, for instance). GA provides a random way to generate the initial solutions. Here, binary coding is adopted to initialize the variable \( u_i \) taking into account that \( |u_i| \leq u_{\text{max}} \). Supposing that each variable \( u_i \) is encoded as an 8 bits binary string \((A_1A_2A_3A_4A_5A_6A_7A_8)\), \( A_2 = 0 \) or 1), the range of \( u_i \) can be limited in \([-u_{\text{max}}, u_{\text{max}}] \) through a simple transformation relation. For example, equation (14) provides one form of the transformation relation.

\[
u_i = u_{\text{max}} \frac{(A - 127.5)}{127.5}, \quad A = (A_1A_2A_3A_4A_5A_6A_7A_8)_2.
\]  

All bits of solution \( U \) are assigned a random value 0 or 1, which is to generate a chromosome. By repeating this operation, the initial population is produced. With decoding operation, they are converted to real values of candidate solutions. Figure 4 gives the generation process of the initial population.

3.3. Nonlinear Prediction and Solution Evaluation. With the given initial solutions and state-space equation of the system, state variables and output variables can be predicted at every future cycle. Traditionally, a linear predictive model is used to make this prediction. By applying GA, both linear model and nonlinear model are feasible, since GA is a forward way to heuristically search for the optimal solution.
Supposing that \( U(k), U(k+1), \ldots, U(k+Nc-1) \) are the control variables and \( g \) is the arbitrary nonlinear function, the future states are predicted as:

\[
X(k+i+1|k) = g(X(k+i|k), U(k+i|k)), i = 1, 2, \ldots, Nc-1.
\]

(15)

Correspondingly, the predicted outputs of future cycles are:

\[
Y_p(k+i+1|k) = C(k+i+1|k)X(k+i+1|k), i = 1, 2, \ldots, Nc-1.
\]

(16)

In traditional MPC, a trade-off between control quantity and tracking performance is considered. The objective function has to meet certain formats in order to implement the solver of convex quadratic programming. By introducing GA, all the control variables and outputs are obtained. So, the objective function can be free-format equations. We can define the objective function of MPC-GA as:

\[
obj = f(Y_p, Y_r, U).
\]

(17)

GA provides an efficient means to solve the optimization problem, even if discontinuities in the objective function or constraints are present [37]. Obviously, one format of the objective function can be the same as an objective function that traditional MPC adopts, namely:

\[
obj = \sum_{i=1}^{Nc} \left( Y_p(k+i|k) - Y_r(k+i|k) \right)^T q_i \left( Y_p(k+i|k) - Y_r(k+i|k) \right) + \sum_{j=0}^{Nc-1} \left( U(k+j+1|k) - U(k+j|k) \right)^T r_i \left( U(k+j+1|k) - U(k+j|k) \right).
\]

(18)

Similarly, the first term of the objective function represents the error between the predicted value and the target signal. The second term reflects the volatility of control quantities.

### 3.4. Iterative Optimization

After the candidate solutions are evaluated, the next step is to implement iterative optimization. Firstly, the best members are selected according to their fitness. Subsequently, the next generation is generated through crossover operation. Then, a mutation operation is conducted in order to avoid trapping in local minimums [38].

The selection process has the main purpose to select the best candidates among the population. Fitness calculation is defined depending on the objective function for each solution candidate in the current population. In detail, fitness calculation is defined as the reciprocal of objective function because minimization of the objective function corresponds to maximum coincidence. The fitness of \( i \)-th individual is defined as (19). The selection is usually driven by fitness-ranking evaluation by means of a specific mathematical method. There exist many different approaches in the literature, and in this study “Roulette Wheel Selection” method [39] is adopted. According to the method, the probability of being selected is defined as (20), where \( N \) represents the population size of the genetic algorithm.

\[
Fitness(i) = \frac{1}{obj(i)}.
\]

(19)

\[
P(i) = \frac{\text{Fitness}(i)}{\sum_{j=1}^{N} \text{Fitness}(j)}.
\]

(20)
After a limited number of iterations, the optimal solution population adapts to the environment more than the genetic operators in natural genetics, a population is generated after the crossover task is completed. Other individuals exchange corresponding segments with each intersection point are selected randomly, and the two in-

Intersection of a sequence of candidates of the optimization problem is defined as the desired steering angle. Consequently, each individual of GA is expressed as (22):

\[
\Delta \delta = \begin{bmatrix} \Delta \delta_1, \Delta \delta_2, \ldots, \Delta \delta_i, \ldots, \Delta \delta_{Nc} \end{bmatrix}, i = 1, 2, \ldots, Nc - 1.
\] (22)

The increment \( \Delta \delta \) reflects the variation of steering angle, and the variation has to satisfy steering angle limitation for safe driving. That is to say, \( \Delta \delta \) should be assigned with a small value if the vehicle speed is high, while a larger value can be assigned to \( \Delta \delta \) if the vehicle speed is low. By limiting the steering angle according to vehicle speed, the controller not only provides safe driving but also guarantees high model accuracy with the dynamic vehicle model [41]. On the other hand, the physical limitations of the steering system are also considered. If the control quantity \( \delta \) exceeds the limitation (max steering angle), the decoder will output the max steering angle.

Comparing traditional encoding method that encodes the solution \( \delta_i \) into binary string directly, this increment encoding method removes the possibility that the independent variables can freely vary from \(-u_{max}\) to \(u_{max}\). So, the computational complexity of the optimization problem is obviously reduced. On the other hand, this encoding method enables the variations of control variables to satisfy constraint conditions easily.

4.2 Nonlinear Prediction. Each individual is converted to real values of control variables. They are used to predict the future trajectory of a vehicle. Different from the traditional linear model prediction, MPC-GA is able to deal with both linear model and nonlinear model, since GA is a forward way to heuristically search for the optimal solution. With the given current states and control variables, the vehicle trajectory of every future cycle can be predicted based on the vehicle kinematics model. Considering the nonlinear kinematics model of vehicle after discretization:
\[
\begin{align*}
\dot{x}(k+1) &= x(k) + T v \cos \theta(k), \\
\dot{y}(k+1) &= y(k) + T v \sin \theta(k), \\
\dot{\theta}(k+1) &= \theta(k) + T v \tan \left( \frac{\delta(k)}{L} \right),
\end{align*}
\]

where \( T \) is the control cycle. Because the change of speed is small during the predicted period, we regard speed \( v \) as a constant that is equal to the current speed. Meanwhile, the less control variables facilitate to reduce the time consumption of the algorithm. Supposing that \( U(k|k), U(k+1|k), \ldots, U(k+Nc-1|k) \) represent the control variables, the future states can be obtained using the predictive model as shown in (25). Correspondingly, the predicted vehicle trajectory can be expressed as (26):

Figure 5: The progress of MPC-GA based path tracking control method.

Figure 6: Straightway road simulation with different speed.
Figure 7: Simulation on curve road that conforms to sinusoidal functions.

Figure 8: Complex situation 1 (reference path: \( y = x \sin (x/30)/10 \)).
Figure 9: Complex situation II (reference path: circle).

Figure 10: Executed paths of the three methods.
\begin{align*}
X(k + i|k) &= g(X(k + i - 1|k)), i = 1, 2, \cdots, N_c, \\
U(k + i - 1|k),
\end{align*}
\begin{equation}
\tag{25}
\end{equation}

\begin{align*}
Y(k + i|k) &= \begin{bmatrix}
x(k + i|k) \\
y(k + i|k) \\
\theta(k + i|k)
\end{bmatrix} \\
&= \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}X(k + i|k), i = 1, 2, \cdots, N_c.
\end{align*}
\begin{equation}
\tag{26}
\end{equation}

To summarize, the predicted vehicle trajectory can be obtained with the following steps:

4.3. Objective Function. In order to balance the performance between position deviation and heading deviation, the objective function of MPC-GA is defined as the form of a quadratic equation as follows:

\begin{equation}
\text{obj} = a \sum_{i=1}^{N_c} \left[ (x(k + i|k) - x_r(k + i|k))^2 + (y(k + i|k) - y_r(k + i|k))^2 \right] + b \sum_{i=1}^{N_c-1} \left[ (\theta(k + i|k) - \theta_r(k + i|k))^2 \right] + c \sum_{i=0}^{N_c-1} \left[ (\Delta \delta(k + i|k))^2 \right],
\end{equation}
\begin{equation}
\tag{27}
\end{equation}

where \(a, b, \) and \(c\) are weighting factors; \(\Delta \delta(k + i|k) = \delta(k + i + 1|k) - \delta(k + i|k)\) is the control increment that represents the change of desired steering angle. The selection of weighting factors is important, and reliance on practical test is often needed. In
order to contrast with the traditional MPC method, the values of \(a, b,\) and \(c\) should be consistent with \(q\) and \(r\) in (10).

The objective function contains tracking errors in the position and heading at each prediction horizon, as well as the total variation of control variables.

4.4. Progress of MPC-GA for Path Tracking. The progress of the MPC-GA-based path tracking control method is shown in Figure 5.

Reference path is input from path planning module. With the given current position, the system chooses the nearest segment of reference path as the target path \(Y_r\). Through increment encoding method, the initial generation population is generated randomly, and each individual represents the control increment. The population can be converted to candidate solutions \(\{U_i\}\) after the decoding operation. Next, the predicted trajectory \(Y_p\) of each solution is obtained by substituting the control sequence into the nonlinear predictive model. According to the fitness function, the fitness of each candidate solution is calculated by comparing the predicted path and target path. Then, the algorithm enters into iterative optimization, and the iteration will stop within finite steps or when the deviation between the predicted trajectory and target path is less than a small value. After iteration, the best individual is selected, and the individual is decoded. Lastly, the first control quantity \(\delta_1\) is output to the actuator.

5. Simulation and Experiment

This section presents the simulations and on-field test of the proposed control method. In order to evaluate the control method, some different simulation scenarios are demonstrated: (1) tracking ability in straight way road is analyzed when the vehicle departs from the target path; (2) curve road with sinusoidal variation is then applied to test the adaptability of tracking path; (3) effectiveness of the proposed control method is further validated in complex situations such as curve road and circle road. In the experiment, the on-field test is conducted based on Kotei 2.0 autonomous platform.

5.1. Simulations. The first simulation is aimed to validate the deflection rectifying ability. The reference path is set as a straight line, and the initial position of the vehicle is set 2 meters apart from the path, while the heading is the same as reference path. The vehicle speed is set from low speed (20 km/h) to high speed (60 km/h) separately and is assumed to be constant. Simulation results are shown in Figure 6. According to the results, the MPC-GA controller can direct the vehicle to reach the reference path with initial departure. The adjustment process is smooth and safe. After the system is stabilized, the track deviation is less than 0.1 m.

Next, the road is set as a curve that the variation conforms to sinusoidal functions \((y = 3.5 \sin (x/30), y = 3.5 \sin (x/10))\), which simulates the lane changing path or obstacle avoidance path. The initial position is set to be near the path. Also, different vehicle speeds are tested in this driving scene. The tracking performance is shown in Figure 7. According to the results, the vehicle perfectly responds to the continuous change of road path. The track deviation is less than 0.1 m most of the time.
Then, we simulate two complex situations. The first situation gives a reference path with variation that conforms to function $y = x_{\text{sin}} (x/30)/10$ as shown in Figure 8. In this situation, an autonomous vehicle needs to respond to changing consecutive curves and sharp turns. The initial position of the vehicle is set 2 meters apart from the path. The initial speed of a vehicle is zero, and the target speed is set as $15\,\text{km/h}$. The second situation adopts circles as the reference path as shown in Figure 9. The vehicle starts at the points 2 meters apart from the circles.

The simulation results in Figures 8 and 9 reveal that the MPC-GA-based path tracking control strategy realizes the path tracking for the two complex situations. The displacement error keeps small all the time after a beginning’s chattering, and
the maximum error is about 0.15 m. The error fully meets the requirements of path tracking for autonomous vehicle.

Lastly, the performance of the MPC-GA method is compared with that of the traditional MPC method and PP control method in similar conditions. The reference path is set as a curve, and the variation conforms to the sinusoidal function \( y = 3.5 \sin\left(x/30\right)\). The original position is set 0.1 m apart from the reference path. The vehicle speed is set as 40 km/h. The executed paths and desired paths are shown in Figure 10. The tracking error and control quantities of the three methods are shown in Figures 11 and 12.

From the results, we can see that all three methods can drive the vehicle following the reference path. At the beginning, MPC method causes a larger overshoot than MPC-GA method and PP method. After entering the stable stage, the tracking error of MPC-GA method and MPC method is less than that of PP method, and tracking precision of MPC-GA method is no less than that of MPC method.

According to the simulation results, the MPC-GA is able to deal with both simple situations and complex situations. Even in the sharp turn situation, the controller generates a corresponding steering angle and drives the vehicle to follow the preset path. The tracking deviation is less than 0.15 m. Compared with the MPC method and PP control method, the path executed with the MPC-GA method is the closest to the desired path.

5.2. Experiment. In order to examine the feasibility and effectiveness of the proposed MPC-GA method, the on-field test is conducted based on Kotei 2.0 platform that is developed by Wuhan Goyu Intelligent Technology Corporation as shown in Figure 13(a). On-field test is conducted on park road of Wuhan Goyu Intelligent Technology Corporation. The satellite map image of the test field is shown in Figure 13(b), with a total area of about 200 m × 200 m. The park road consists of corners with different turning radius. The road width is 3.6 m approximately. First, we drive the car around the park road to collect GPS data (position in the XY plane) to get the global reference path.

Then, we start the autonomous system, and the vehicle moves automatically from the start point to the end point with the autonomous system. The travel path is recorded. Figure 14 shows the road map and travel path in the XY plane.

The vehicle passes by corner A, corner B, corner C, and corner D successively, and the local enlarged images of the four corners and two straight roads are shown in Figures 15 and 16. The tracking error of the entire process is depicted in Figure 17.

Figures 15–17 illustrate the vehicle’s performance when it travels from start point to end point. On the straight road, the travel path is almost perfectly consistent with the reference path. When encountering the corners, the vehicle will deviate from the reference path. With the self-adjusting ability of MPC-GA, the vehicle rapidly tracks the reference path again. For the straight road, the average tracking error is less than 0.05 m, while the maximum error is 0.1 m. For the curve road, the average tracking error is less than 0.2 m, while the maximum error is about 0.3 m.

The above simulations and experiment indicate that the MPC-GA method has realized the path tracking control for autonomous vehicle with straight road scene and curve road scene. It also shows a good self-adjusting performance when there is a sharp turn. The controller eliminates the tracking error and directs the vehicle to reach the reference path with a small deviation. Meanwhile, the MPC-GA method demonstrates better performance than traditional methods. Therefore, the proposed path tracking control method can ensure the vehicle driving in accordance with the reference path.

6. Conclusion

This study puts forward a path tracking control method for autonomous vehicle. The method combines the traditional MPC and GA to obtain the optimal solution. The proposed MPC-GA method can solve the nonlinear optimization problem without any compromise, and effectively deal with the objective function of any free-format equations. The control strategy is applied in the path tracking problem to implement the steering control. Simulations and on-field tests illustrate the performance of the proposed method. By the MPC-GA method, the autonomous vehicle can track the preset path both on straightway road and curve road at different speed. Compared with traditional MPC method and PP method, the proposed MPC-GA method is of higher accuracy. The control strategy can be considered as an
alternative method to solve the path tracking problem of autonomous vehicle.

This work needs to be further carried out. Subsequent research will focus on reducing algorithm complexity, in order to improve real-time performance. Currently, the time consumption of the proposed MPC-GA algorithm is almost similar to MPC. Reducing the GA population or decreasing evolution generations can further reduce the time complexity of MPC-GA, but at the same time leads to poor control performance. Therefore, trying new attempt and approach to reduce algorithm complexity without decreasing control performance will be our future work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the Hubei Key R&D Plan (2021BAA020) and Wuhan Science and Technology Plan (2020030603012331).

References

[1] K. Lee and D. Kum, “Collision avoidance/mitigation system: motion planning of autonomous vehicle via predictive occupancy map,” IEEE Access, vol. 7, pp. 52846–52857, 2019.

[2] H. Wang, X. Chen, and Y. Chen, “Trajectory tracking and speed control of cleaning vehicle based on improved pure Pursuit algorithm *,” in Proceedings of the 38th Chinese Control Conference, Guangzhou, China, 2019.

[3] J. Ahn, S. Shin, M. Kim, and J. Park, “Accurate path tracking by adjusting look-ahead point in pure pursuit method,” International Journal of Automotive Technology, vol. 22, no. 1, pp. 119–129, 2021.

[4] Q. Kong, L. Zhang, and X. Xu, “Control algorithm of path planning and path tracking for automated valet parking,” Journal of Physics: Conference Series, J. Phys.: Conf. Ser. vol. 2232, no. 1, Article ID 012018, 2022.

[5] B. Ariffin, B. Y. Suprapto, S. A. D. Prasetyowati, and Z. Nawawi, “Steering control in electric power steering autonomous vehicle using type-2 fuzzy logic control and PI control,” World Electric Vehicle Journal, vol. 13, no. 3, pp. 53–21, 2022.

[6] W. Hu, S. Zhang, and L. Niu, “Self-adaptive PID control based on RBF network for trajectory tracking of dual-mass servo system,” Journal de Physique: Conf. Ser. vol. 1871, no. 1, Article ID 012113, 2021.

[7] C. Nanding and C. Jing, “Design of intelligent tracing car based on infrared sensor,” IOP Conference Series: Materials Science and Engineering, vol. 563, no. 4, Article ID 042006, 2019.

[8] Y. Zhou, “Research on self-tracking car based on fuzzy-PID control algorithm,” Journal de Physique: Conf. Ser. vol. 2170, no. 1, Article ID 012012, 2022.

[9] L. Sun, J. Ma, and B. Yang, “Fuzzy PID design of vehicle attitude control systems,” in Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Heifei, China, 2020.

[10] S. Chen and H. Chen, “MPC-based path tracking with PID speed control for autonomous vehicles,” IOP Conference Series: Materials Science and Engineering, vol. 892, no. 1, Article ID 012034, 2020.

[11] M. Samuel, M. Hussein, and M. Binti, “A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle,” International Journal of Computer Application, vol. 135, no. 1, pp. 35–38, 2016.

[12] L. Yu, X. Yan, Z. Kuang, B. Chen, and Y. Zhao, “Driverless bus path tracking based on fuzzy pure Pursuit control with a front axle reference,” Applied Sciences, vol. 10, no. 1, pp. 230–319, 2019.

[13] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, “A novel pure Pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller,” IEEE Access, vol. 8, pp. 166525–166540, 2020.

[14] J. Li, Q. Wu, and J. Wang, “Autonomous tracking control for four-wheel independent steering robot based on improved pure Pursuit,” Journal of Beijing Institute of Technology (Social Sciences Edition), vol. 29, no. 4, pp. 466–473, 2020.

[15] J. Liu, Q. Dai, and H. Guo, “Human-oriented online driving authority optimization for driver-automation shared steering control,” IEEE T Intell Vehicl., 2022.

[16] Z. Wang, X. Zhou, and J. Wang, “Extremum-seeking-based adaptive model-free control and its application to automated vehicle path tracking,” IEEE-Asme T Mech, 2022.

[17] A. T. Nguyen, J. Rath, T. M. Guerra, R. Palhares, and H. Zhang, “Robust set-invariance based fuzzy output tracking control for vehicle autonomous driving under uncertain lateral forces and steering constraints,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 5849–5860, 2021.

[18] C. Zhang, D. Chu, S. Liu, Z. Deng, C. Wu, and X. Su, “Trajectory planning and tracking for autonomous vehicle based on state lattice and model predictive control,” IEEE Intelligent Transportation Systems Magazine, vol. 11, no. 2, pp. 29–40, 2019.

[19] G. Wang, L. Liu, Y. Meng, Q. Gu, and G. Bai, “Integrated Path tracking control of steering and braking based on holistic MPC,” IFAC-PapersOnLine, vol. 54, no. 2, pp. 45–50, 2021.

[20] A. Mitov, J. Kraliev, T. Slavov, and I. Angelov, “Comparison between model predictive (MPC) and model reference adaptive controllers (MRAC) for electrohydraulic steering system implemented as real-time simulink program,” IOP Conference Series: Materials Science and Engineering, vol. 1002, no. 1, Article ID 012034, 2020.

[21] M. Kim, D. Lee, J. Ahn, M. Kim, and J. Park, “Model predictive control method for autonomous vehicles using time-varying and non-uniformly spaced horizon,” IEEE Access, vol. 9, pp. 86475–86487, 2021.

[22] J. Liu, Z. Yang, and Z. Huang, “Simulation performance evaluation of pure Pursuit, Stanley, LQR, MPC controller for autonomous vehicles,” in Proceedings of the 2021 IEEE International Conference on Real-time Computing and Robotics, Xining China, 2021.

[23] J. Nan, B. Shang, W. Deng, B. Ren, and Y. Liu, “MPC-Based path tracking control with forward compensation for autonomous vehicle.”
autonomous driving,” IFAC-PapersOnLine, vol. 54, no. 10, pp. 443–448, 2021.

[24] S. Chen, H. Chen, and N. Dan, “Implementation of MPC-based trajectory tracking considering different fidelity vehicle models,” Journal of Beijing Institute of Technology (Social Sciences Edition), vol. 29, no. 3, pp. 303–316, 2020.

[25] A. Alessandretti, A. P. Aguiar, and C. N. Jones, “Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control,” in Proceedings of the Control Conference. 2013 European Control Conference, Zurich, Switzerland, 2013.

[26] 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.

[27] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, “An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles,” IEEE Access, vol. 8, pp. 51400–51413, 2020.

[28] P. Huang, X. Luo, and Z. Zhang, “Simulation of autonomous agricultural machine based on improved pure Pursuit model,” in Proceedings of the Third IFIP International Conference on Computer and Computing Technologies in Agriculture, Beijing, China, 2009.

[29] S. Banihani and S. De, “Development of a genetic algorithm-based lookup table approach for efficient numerical integration in the method of finite spheres with application to the solution of thin beam and plate problems,” International Journal for Numerical Methods in Engineering, vol. 67, no. 12, pp. 1700–1729, 2006.

[30] Z. Xu, W. Zhao, and C. Wang, “Local path planning and tracking control of vehicle collision avoidance system,” Transaction of Nanjing University of Aeronautics and Astronautics, vol. 35, no. 4, pp. 729–738, 2018.

[31] H. Wang, B. Liu, X. Ping, and Q. An, “Path tracking control for autonomous vehicles based on an improved MPC,” IEEE Access, vol. 7, pp. 161064–161073, 2019.

[32] K. Edlund, L. E. Sokoler, and J. B. Jorgensen, “A primal-dual interior-point linear programming algorithm for MPC,” in Proceedings of the 48th IEEE Conference on Decision & Control, Shanghai, China, 2009.

[33] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, “OSQP: an operator splitting solver for quadratic programs,” Mathematical Programming Computation, vol. 12, no. 4, pp. 637–672, 2020.

[34] Z. Wang, G. Li, H. Jiang, Q. Chen, and H. Zhang, “Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control,” IEEE, vol. 23, no. 3, pp. 1103–1113, 2018.

[35] L. Wang, Z. Zhai, Z. Zhu, and E. Mao, “Path tracking control of an autonomous tractor using improved Stanley controller optimized with multiple-population genetic algorithm,” Actuators, vol. 11, no. 1, pp. 22–14, 2022.

[36] J. Chen, H. Zhu, L. Zhang, and Y. Sun, “Research on fuzzy control of path tracking for underwater vehicle based on genetic algorithm optimization,” Ocean Engineering, vol. 156, pp. 217–223, 2018.

[37] T. Tometzki, O. Stursberg, and C. Sonntag, “Optimizing hybrid dynamic processes by embedding genetic algorithms into MPC,” in Proceedings of the International Symposium on Advanced Control of Chemical Processed, Gramado, Brazil, 2006.