2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

A Robust Look-ahead Distance Tuning Strategy for the Geometric Path Tracking Controllers

Longsheng Chen
School of Data
and Computer Science
Sun Yat-Sen University
Guangzhou, China
Email:
chenlsh25@mail2.sysu.edu.cn

Ni Liu
School of Data
and Computer Science
Sun Yat-Sen University
Guangzhou, China
Email:
liu39@mail2.sysu.edu.cn

Yunxiao Shan
School of Data
and Computer Science
Sun Yat-Sen University
Guangzhou, China
Email:
yunxiao.shan@gmail.com

Long Chen
(corresponding author)
School of Data
and Computer Science
Sun Yat-Sen University
Guangzhou, China
Email:
chenl46@mail.sysu.edu.cn

***Abstract*—Geometric path tracking controllers are very popular to be implemented in the autonomous vehicles because of its simple implementation and validation in the practical applications. However, it also suffered from the sensitivity to the selections of the look-ahead distance. Therefore, reasonable tuning strategy of the look-ahead distance is very important to the application of the geometric controllers. This paper proposed a more robust tuning strategy for all the geometric controllers based on the careful analysis of the previous methods. Instead of utilizing as much as possible elements, we just want to make use of only two variables, the distance from the current location to the reference path and the changing rate of it. Fuzzy logic was utilized to combine these two variables to tune the look-ahead distance. We will show that we just use two variables, but we actually consider another two variables' affections indirectly. Lastly, we will add our strategy to two different classical geometric controllers and compare the relative benefits of previous tuning strategies.**

***Index Terms*—Geometric controller, Path tracking, Fuzzy logic.**

I. Introduction

The autonomous vehicle technologies are becoming a hot topic in recent years. Some of them have been used for the public, such as LDWS(Lane departure warning system)[1], ACC(Adaptive Cruise Control)[2], etc. However, the navigation is still a challenging field considering the complex environment and real-time computation requirements. Normally, navigation includes two tasks, one is to plan a feasible and non-collision path, the other is to control the vehicle to track the path accurately to safely move forward. This paper focused on the second function of the navigation, path tracking technology.

Generally, there are just for two types of controllers, geometric controllers and controllers based on non-geometric relationship. Geometric controllers build a geometric relationship between the current state and target state in the traced path to calculate the steering angle. The relationship can be built by the circle[3], Clothoid curve[4], the screw theory[5] and etc. The geometric controller is easy to implement and robust enough to handle the tracking problem of the autonomous vehicle. As for the controllers based on non-geometric relationship, such as the feedback controllers[6], model based controllers[7][8]. The feedback controllers can be applied into any system without consideration of the model. It only calculates the controls by dealing with the output of the system. In contrast, the model based controllers rely on the vehicle's model. With the model and optimization method, the model based controllers can perform accurate and smooth control.

However, the non-geometric controllers are not widely used in the autonomous vehicles. The feedback controller suffers from the regulation of the parameters and the model based controllers' performance is affected by the model's fidelity to the real vehicle. As we all know, it is difficult to obtain a high fidelity vehicle model in the practical application. Moreover, the optimization methods used in the model based controllers may increase the computing complexity and then affect the real-time application. As for the geometric controller, its performance depends on the fitting method and the tuning mechanism of the look-ahead distance[9]. Especially for the tuning mechanism of the look-ahead distance, which is difficult to be decided because of too many influential factors, such as the current location of vehicle, the velocity, the tracking error and etc. In this paper, we try to design a robust and independent tuning strategy of the look-ahead distance. The detailed contribution of this paper is listed below:

- We design a fuzzy strategy using only two variables to tune the look-ahead distance. Besides, we will prove the employed variables can reflect more influential factors than themselves.
- According to the human drivers' experiences and the tests, we design the fuzzy rules and define the membership functions of the two input variables and also the output variable.
- We conduct a simulation test to apply the designed strategy on three classical geometric controllers and then compare their performances with the proposed tuning strategy with two other strategies under different velocities. The final results prove the tracking performance is improved by the controller with our fuzzy tuning

strategy.

This paper is organized as follows: we firstly review the recent developments of the geometric controllers applied for the autonomous vehicle. Secondly, we use fuzzy logic to design a tuning strategy and explain our modifications. Thirdly, we compare two classical geometric controllers to prove our proposed strategy is effective as an add-on. At the end of this paper, we have a conclusion from our method and look forward to the future development of the geometric controllers used for the autonomous vehicles.

II. Literature reviews

A. Geometric path tracking controllers

Geometric controllers are fundamentally built from the analysis of the geometric relationship between the vehicle model and the reference path, and how these vehicle models track the path. Geometric controllers utilize the location information and the reference path to compute the control command of the vehicle by the kinematic model.

The most classical geometric controller is the PP (Pure-Pursuit) controller which is commonly used in the autonomous vehicles. The first PP was designed to make the missile to track a target. PP can control the missile to tune the velocity direction to follow the moving target[9]. Based on this application, Wallace[11] proposed it to use in the autonomous vehicle, to follow a lane center got from the camera. Field tests prove PP can control the vehicle to track the defined path very accurately. Furthermore, Amidi[10] expands the applications of PP, which controls a little mobile robot to follow a path. However, previous application does not show how the specific process of the implantation. Coulter[3] detailed the implementation issue of PP. Since then, PP was widely applied for indoor[13] and outdoor robots. Nevertheless, the large tracking error makes PP could not satisfy some accurate applications when tracking a curved path. Jeff Wit proposed a method, named vector pursuit[5]. Although some field tests prove the vector pursuit is better than PP, The implantation and tuning are more difficult. Kristian[17] proposed a lateral controller, alice method. Citlalli[12] compared the alice method with several other geometric controllers and conclude alice method is better in tracking performance.

B. Look-ahead distance tuning strategy

Besides of the applications, some researchers also want to analyze the functional laws and then to optimize it. Ollero[14] analyzed how the look-ahead distance affects the performance and the stability of the PP. One of the most significant conclusions of [14] is that the look-ahead distance must be greater than 1 when the robot wants to follow a straight path. Moreover, when following a curved path, the look-ahead distance must increase with the curvature and the delay and also be greater than 1. In summary, we must design a tuning strategy for the selection of the look-ahead distance based on some variables used in tracking if we wish to optimize the controller.

Realizing the importance of the look-ahead distance, Anibal Ollero[15] employed fuzzy logic to tune the look-ahead distance. He considered almost all possible variables that may affect the selection of the look-ahead distance and designed a huge fuzzy strategy. To simplify the computation and to improve the efficiency, Kuwata proposed a simplified adaptive method that tuning the look-ahead distance from the velocity. The tuning method's parameters were obtained from a number of different experiences[16][17]. However, Kuwata's tuning strategy was only being tested for 30km/h, higher speed may require more experience to tune the parameters in the tuning strategy.

In this paper, we want to discuss the tuning strategy problem. In the literature[15], four variables were considered to take effect on the selection of the look-ahead distance. But in Kuwata method, only command velocity was considered. Excessive redundant variables will increase the designing difficulties, and also make the strategy more complex in tune. However, using command velocity directly to tune is also arguable, on the one hand, deriving a reasonable command velocity is not easy when considering the complex dynamic of the vehicle and the partially known environments. On the other hand, any oscillations in velocity will impact on the look-ahead distance, which may make the steering process unstable. Therefore, designing a robust strategy should hold this in mind, using less input variables, and making the variables function indirectly.

III. Tuning strategy based on fuzzy logic

We also design a strategy based on fuzzy logic, but will use less input variables than Ollero[14]. Firstly, we introduce the basic concept of fuzzy logic and how this technique is related to our strategy. And then, we explain the selection reason for the input variables based on some mathematical analysis. Finally, we show our fully proposed fuzzy strategy.

A. Fuzzy logic

Fuzzy logic is the theory that imitates the judgment or decision made by human being from the uncertainty using the degrees of truth as a mathematical model of vagueness, or a mathematical model of ignorance. Considering the uncertainty and imprecision, fuzzy logic uses a series of linguistic values to express the vagueness variables with adjectives or adverbs.

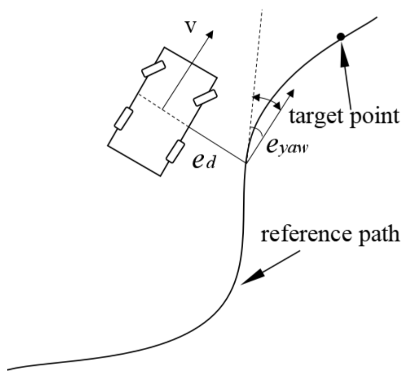

Fig. 2. Variables in path tracking model

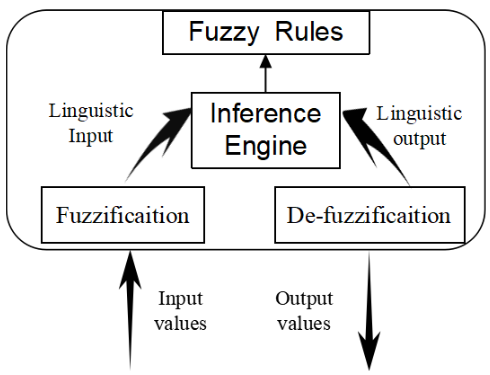

Fig. 1. The framework of fuzzy control

Fig. 1 shows the framework of fuzzy logic. Fuzzy logic has two steps to deal with inputs and outputs: firstly, fuzzification will map the input numerical values into the fuzzy membership function; secondly, defuzzification can be used to map a fuzzy output membership function to compute the output numerical values. In general, fuzzy logic uses a sequence of rules based on the experts' knowledge to make the "fuzzy" decision expressed by the linguistic values.

To our tuning strategy, there is not a robust and general math model required by accurate control method that has already proved helpful to reflect the relationships between the look-ahead distance and the input variables. We only have some experience in the controllers' performance and the look-ahead distance, like long look-ahead distance brings stability but bad accuracy and short look-ahead distance may improve the accuracy but make the controller unstable. Moreover, we also have some human driver's experience. Based on what we have, the fuzzy logic is the best choice for us. To design a fuzzy strategy, the most important thing is to choose appropriate input variables.

B. Choosing the input variables

As shown in Fig.2, v means the velocity of the vehicle, e_d the track error and e_{yaw} the heading error between the vehicle and the reference path. There are lots of variables available for us to choose to be the input variables in the fuzzy strategy. Almost every variable shown in Fig.2 is related to the look-ahead distance. However, choosing excessive input variables may increase the computation burden exponentially. To design a more efficient fuzzy strategy, we should use as fewer as possible input variables but with more impacts on the driving performance. With this in mind, we want to optimally select variables in Fig.2.

$$\begin{cases} \dot{x} = v \cdot cos\theta \\ \dot{y} = v \cdot sin\theta \\ \dot{\theta} = v \cdot k \end{cases} \tag{1}$$

As showed in Eq.(1) and Fig.3, θ means the heading(or yaw) of the vehicle, and k means the curvature of the motion. Fig.3 shows the geometric model of robotic vehicle, L means the weelbase of vehicle, R means the tuning radius, v means the velocity of vehicle, δ means e_{yaw} in Fig.2.

$$\begin{cases} e_d = \sqrt{\Delta x^2 + \Delta y^2} & (2) \\ \dot{e}_{d_x} = \dfrac{\Delta x \cdot \dot{x}}{e_d} & (3) \\ \dot{e}_{d_y} = \dfrac{\Delta y \cdot \dot{y}}{e_d} & (4) \\ \dot{e}_{d_x} = \dfrac{\Delta x \cdot v \cdot cos\theta}{e_d} & (5) \\ \dot{e}_{d_y} = \dfrac{\Delta y \cdot v \cdot sin\theta}{e_d} & (6) \end{cases}$$

In our proposed tuning strategy, we choose track-error(e_d) and the changing rate of the track-error($\dot{e}_d$). The reason to choose e_d is that it could be felt by the human drivers easily. When a driver wants to track a defined path, he will sense whether the vehicle is far away or close to the path. Therefore, e_d is a perfect variable to express human driver's experience. As for the reason to select $\dot{e}_d$, we prove that it could represent more impacts than itself.

According to the geometric model of the vehicle shown in Fig.3, and the motion model in Eq.(1), we did some deductions of Eq.(3) and Eq.(4), and substitute Eq.(1) into Eq.(5) and Eq.(6), we obtain the relation between the $\dot{e}_d$ and the velocity and the yaw. The $\dot{e}_d$ can influence itself, the velocity and the yaw. In addition, Δx and Δy mean the longitudinal and lateral distance between the current position and the nearest point on the path.

C. Designed fuzzy rules

After selecting the input variables, we need to design some rules that are from the rich experience of the human drivers and the tests to deal with the two input variables and then to infer the proper look-ahead distance.

The experience of human drivers and the normal tests:

- If the car is leaving away from the road, the driver should turn the steering wheel to make the car close to the road.
- If the car is moving close to the road, the driver should hold the steering wheel and turn it back to the forward direction as slow as possible.

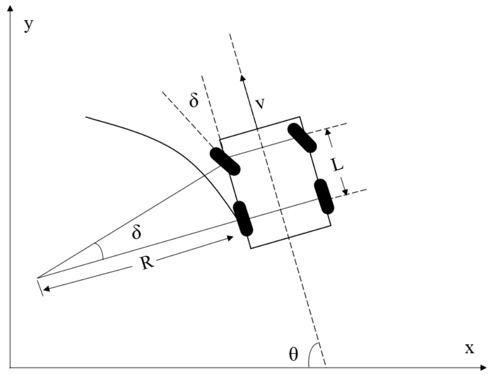

Fig. 3. The geometric model of robotic vehicle

- If the car is staying within the road, the driver should hold the steering wheel in the forward direction.
- If the car has a high velocity, the driver should hold the steering wheel in the forward direction and turn the steering wheel slightly to keep the car on the road.
- On the autonomous vehicle, it can accept e_d less than 0.2 m.
- The max value of the e_d should not be greater than 1 m.
- $\dot{e}_d$ lay between -0.1 and 0.1 m/s in the normal driving.

The Fuzzy rules: Translating driver's experience in fuzzy rules is the key process for our method. Because we take two factors into account for our tuning strategy, e_d and $\dot{e}_d$. When the value e_d and $\dot{e}_d$ are changing, it will output a proper look-ahead distance. The designed fuzzy rules to tune the look-ahead distance are illustrated as follows:

1) If e_d is smallsmall and $\dot{e}_d$ is negbig then l_d is great;
2) If e_d is smallsmall and $\dot{e}_d$ is negmiddle then l_d is middleplus;
3) If e_d is smallsmall and $\dot{e}_d$ is negmiddle then l_d is middle;
4) If e_d is smallsmall and $\dot{e}_d$ is posmiddle then l_d is middle;
5) If e_d is smallsmall and $\dot{e}_d$ is posgreat then l_d is small;
6) If e_d is small and $\dot{e}_d$ is negbig then l_d is middleplus;
7) If e_d is small and $\dot{e}_d$ is negmiddle then l_d is middle;
8) If e_d is small and $\dot{e}_d$ is zero then lookaheaddist is small;
9) If e_d is small and $\dot{e}_d$ is posmiddle then l_d is smallsmall;
10) If e_d is small and $\dot{e}_d$ is posgreat then l_d is smallsmall;
11) If e_d is middle and $\dot{e}_d$ is negbig then l_d is middleplus;
12) If e_d is middle and $\dot{e}_d$ is negmiddle then l_d is small;
13) If e_d is middle and $\dot{e}_d$ is zero then l_d is smallsmall;
14) If e_d is middle and $\dot{e}_d$ is posmiddle then l_d is smallsmall;
15) If e_d is middle and $\dot{e}_d$ is posgreat then l_d is smallsmallsmall;
16) If e_d is great and $\dot{e}_d$ is negbig then l_d is middle;
17) If e_d is great and $\dot{e}_d$ is negmiddle then l_d is small;
18) If e_d is great and $\dot{e}_d$ is zero then l_d is smallsmall;
19) If e_d is great and $\dot{e}_d$ is posmiddle then l_d is smallsmallsmall;
20) If e_d is great and $\dot{e}_d$ is posgreat then l_d is smallsmallsmall.

Fuzzy variables: As the discussion above, the input values and output values in the fuzzy logic will use linguistic values. Therefore, we name the two input variables track error and the rate of tracking error as e_d and $\dot{e}_d$. At the same time, the output variable is only the look-ahead distance which named as l_d in the fuzzy logic. As our definition, the fuzzy variable e_d has four associated linguistic values: $smallsmall$, $small$, $middle$ and $large$, while $\dot{e}_d$ has five associated linguistic values: $negbig$, $negmiddle$, $zero$, $posmiddle$ and $posgreat$. And the output fuzzy variable l_d has five associated linguistic values: $smallsmallsmall$, $smallsmall$, $small$, $middleplus$ and $great$. All the linguistic values are set to a reasonable range by a mass of experiments.

IV. EXPERIENCES

We use a high-fidelity simulator to prove our proposed strategy is effective. The simulator in Fig.4 is a complete and independent simulating environment for Udacity's self-driving car Nanodegree program.Theoretically, our strategy can be used in almost all the geometric controllers. In our experience, we choose two classical and practical controllers, one is the PP, the other is the Alice method[18]. We will test our method by combining it with these controllers. We firstly present the two compared geometric controllers simply.

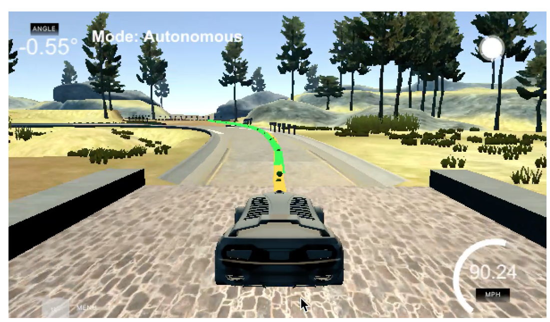

Fig. 4. Simulator

A. Two different geometric controllers

PP is one of the most basic and general methods used for calculating the steering angle to the autonomous vehicle. Given a reference path, according to pre-defined look-ahead distance, a goal point will be automatically determined by the calculation of the steering angle.

$$\delta = arctan\left(\frac{2Lsin\phi}{l_d}\right) \quad (7)$$

The Eq.(7) shows the calculating process. As in the PP, the steering angle only depends on the wheelbase L, the look-ahead distance l_d and the angle ϕ between the vehicle heading and the look-ahead heading.

Alice method[18] is a nonlinear control strategy for trajectory tracking developed by Team Caltech and implemented

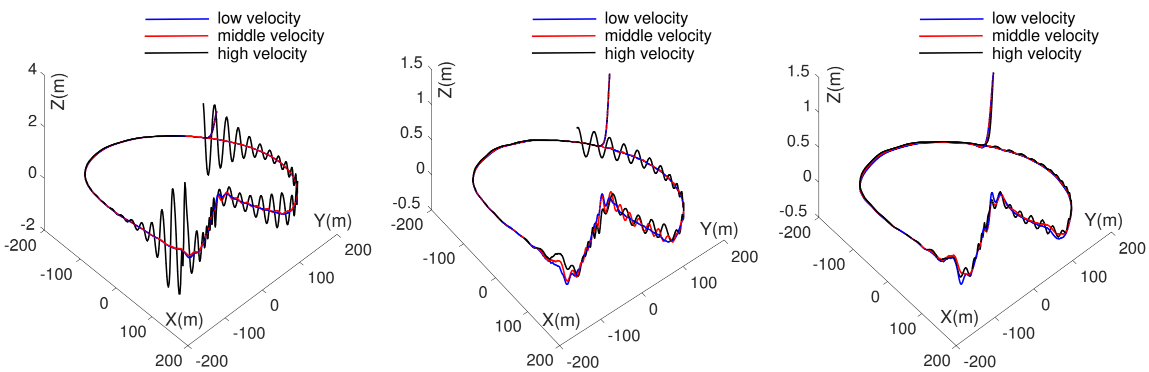

Fig. 5. The e_d results of using PP controller with the different look-ahead distance tuning methods. **Left:** constant look-ahead distance. **Middle:** MIT tuning strategy. **Right:** Our look-ahead tuning strategy.

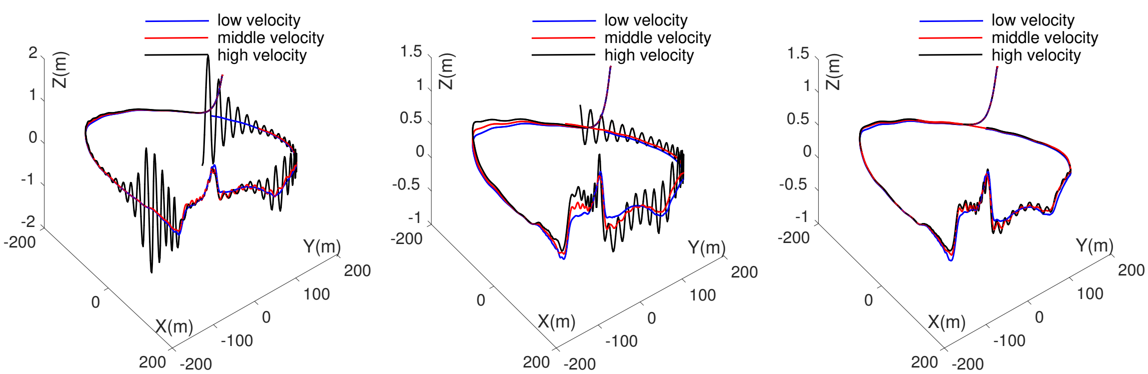

Fig. 6. The e_d results of using Alice controller with the different look-ahead distance tuning methods. **Left:** constant look-ahead distance. **Middle:** MIT tuning strategy. **Right:** Our look-ahead tuning strategy.

in the Alice autonomous car. Their autonomous vehicle competed for the 2005 DARPA autonomous vehicle challenge. Alice method uses Ackerman steering principle and the kinematics of the bicycle model. The control law is shown as follows:

$$\delta = arctan\left(\frac{-cos\left(e_\theta\right)e_d - \left(L + l_d\right)sin\left(e_\theta\right)}{L - \left(L + l_d\right)cos\left(e_\theta\right) + sins\left(e_\theta\right)e_d}\right) \quad (8)$$

As the Eq.(8) shown, the track error e_d is the distance between the rear axle and the nearest points of the reference path. The yaw error e_θ is the angle between the vehicle direction and the tangent of the projected vehicle position in the path. L is the vehicle wheelbase. l_d is the look-ahead distance. This method's purpose is to try its best to reduce the track error e_d and the yaw error e_θ.

$$l_d\left(v_{cmd}\right) = \begin{cases} 3 & if v_{cmd} < 1.34 \\ 2.24v_{cmd} & if 1.34 < v_{cmd} < 5.36 \\ 12 & otherwise \end{cases} \quad (9)$$

In order to demonstrate the performance of our tuning strategy, we will compare with the fixed value look-ahead distance and the strategy related to the velocity. The strategy with the velocity will use the MIT method shown in Eq.(9). l_d means the look-ahead distance, while v_{cmd} means the command velocity given by the setting parameter.

B. Comparing the results

In order to show the proposed strategy could accommodate velocity's change better than other methods. We will test our strategy with different velocities: the low velocity, the middle velocity and the high velocity. Therefore, parameters for our simulation that we have set are shown in Table.I. Different look-ahead distance strategies are applied to different controllers, and the e_d in the Fig.5 and Fig.6 show their performances in the tracking accuracy.

We can conclude from the experimental results, that the constant look-ahead distance can not accommodate the

TABLE I
PARAMETERS FOR SIMULATION

Acceleration	$\pm 1 m \cdot s^{-2}$
Low Velocity	$0 \sim 30 km \cdot h^{-1}$
Middle Velocity	$30 \sim 60 km \cdot h^{-1}$
High Velocity	$60 \sim 90 km \cdot h^{-1}$
Steering Angle	$\pm 25 \deg$
Wheel Base	$2.76 m$

velocity's change, and behave worst(Fig.5 **Left** and Fig.6 **Left**). Although MIT method uses velocity to tune the look-ahead distance, the oscillation problem is still unavoidable when applying the high velocity in the curved road(Fig.5 **Middle** and Fig.6 **Middle**). MIT method requires to design a good target velocity control law to keep good performance. However, it is difficult to design even just considering such a simple environment like in Fig.4(without any obstacles). Moreover, using the command velocity to tune the look-ahead distance directly will not only bring the effective impact of the velocity control law but also the bad effects. If the designed command velocity control law has any defects, no filters are employed to narrow it on the MIT method.

Our method has the best performance even in the highest velocity(Fig.5 **Right** and Fig.6 **Right**). Comparing with the MIT method, we actually could use Eq.(5) and Eq.(6) as a filter to the velocity and the yaw. When the command velocity changes, it firstly makes the vehicle tune the command velocity and then the real-time velocity is affected. Next, real-time velocity impacts on the $\dot{e}_d$. Through the indirect impact transformation, we weaken the velocity's impacts, but bring more impacts than MIT method, like the yaw and the $\dot{e}_d$. In the curved segment, MIT method could not tune the look-ahead distance properly, because we could not design a command velocity control law to accommodate the environment. However, our proposed strategy does not have this kind of requirement. We could obtain very good performance through tuning the look-ahead distance by the fuzzy logic, especially in the segments difficult to design a corresponding velocity controller.

V. CONCLUSION

In this paper, we proposed a more robust look-ahead distance tuning strategy for the geometric path tracking controllers so that the controllers can be adaptive to different velocities and perform more stable. The fuzzy logic is employed in our strategy and the reason to choose the input variables is explained in mathematics and actual driving tests. One of our innovations is that we prove that it is not necessary to add all the related variables in the fuzzy input, and two variables are sufficient to tune the look-ahead distance.

Although the proposed method in this paper performs well, the continuity of the control can't be guaranteed. We need to consider more in tuning the look-ahead distance in the future research, such as the previous controls and other real-time constraints.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61773414.

REFERENCES

[1] Wang, Wenshuo, et al. "A Learning-Based Approach for Lane Departure Warning Systems with a Personalized Driver Model." (2017).

[2] Bayar, Bilgehan, et al. "Impact of different spacing policies for adaptive cruise control on traffic and energy consumption of electric vehicles." Control and Automation IEEE, 2016:1349-1354.

[3] Coulter R C. Implementation of the pure pursuit path tracking algorithm. Carnegie-Mellon Univer sity Pittsburgh, Robotics Institute; 1992. CMU-RI- TR-92-01.

[4] Girbs V, Armesto L, Tornero J, Solanes J E. Continuous-curvature kinematic control for path following problems. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on 25-30 September 2011; San Francisco. IEEE; 2011. p.4335-4340.

[5] Wit J, Crane C D, Armstrong D. Autonomous ground vehicle path tracking. Journal of Robotic Systems. 2004;21(8):439-449.

[6] G Reynoso-MezaJ Carrillo-AhumadaY BoadaJ Pic. PID controller tuning for unstable processes using a multi-objective optimisation design procedure. IFAC-PapersOnLine, 2016, 49(7): 284-289.

[7] Yu, Ru and Guo, Hongyan and Sun, Zhenping and Chen, Hong. MPC-Based Regional Path Tracking Controller Design for Autonomous Ground Vehicles. IEEE International Conference on Systems, 2016:2510-2515.

[8] OH Dagci, UY OgrasU Ozguner. Path following controller design using sliding mode control theory. American Control Conference, 2003, 1:903-908 vol.1.

[9] Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li. CF-Pursuit: A Pursuit Method with a Clothoid Fitting and a Fuzzy Controller for Autonomous Vehicles. International Journal of Advanced Robotic Systems, vol. 12, no. 9, 2015.

[10] Amidi O, Thorpe C E. Integrated mobile robot control. Fibers '91. Boston: International Society for Optics and Photonics; 1991. p. 504-523.

[11] Wallace R, Stentz A, Thorpe C E, Moravec H, Whittaker W, Kanade T. First results in robot road- following. In Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI 85); August 1985; Los Angeles. 1985. 1:66-71.

[12] Citlalli Gamez Serna, Ruichek Yassine. Dynamic Speed Adaptation for Pat Tracking Based on Curvature Information and Speed Limits. Sensors 2017, 17, 1383.

[13] Morales J, Martnez J L, Martnez M A, Mandow A. Pure-pursuit reactive path tracking for nonholo nomic mobile robots with a 2D laser scanner. EURASIP Journal on Advances in Signal Process ing, 2009; 3.

[14] Ollero A, Heredia G. Stability analysis of mobile robot path tracking. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95); August 1995; Pittsburgh. 1995. (3):461-466.

[15] Ollero A, Garcia-Cerezo A J, Martinez J L. Fuzzy supervisory path tracking of mobile robots. Control Engineering Practice. 1994; 2(2):313-319.

[16] Kuwata Y, Teo J, Fiore G, Karaman S, Frazzoli E, How J P. Real-time motion planning with applica tions to autonomous urban driving. Control Systems Technology. IEEE Transactions on Control Systems Technology. 2009; 17(5):1105-1118.

[17] Kuwata Y, Teo J, Karaman S, Fiore G, Frazzoli E, How J P. Motion planning in complex environments using closed-loop prediction. In Proceedings of AIAA Guidance, Navigation, and Control Confer ence and Exhibit (AIAA 2008); August 2008; Honolulu. 2008.

[18] Linderoth, M. Soltesz, K. Murray, R.M. Nonlinear lateral control strategy for nonholonomic vehicles. In Proceedings of the 2008 American Control Conference, Seattle, WA, USA, 1113 June 2008; pp. 32193224.