Adaptive Pure Pursuit with Deviation Model Regulation for Trajectory Tracking in Small-Scale Racecars

Ralph Al Fata
Department of Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon
rea35@mail.aub.edu

Naseem Daher
Department of Electrical and Computer Engineering
Department of Mechanical Engineering
American University of Beirut
Beirut, Lebanon
nd38@aub.edu.lb

2025 International Conference on Control, Automation, and Instrumentation (IC2AI) | 979-8-3315-2960-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/IC2AI62984.2025.10932261

***Abstract*—Designing motion controllers for autonomous racecars allows researchers to study the safety and computational efficiency of their algorithms under extreme conditions, while also serving as a benchmark for conventional autonomous driving technologies. Small-scale autonomous racecars compel developers to use feeble processors or microcontrollers that are incapable of executing computationally-expensive control algorithms, like Model Predictive Control (MPC), in real time. On the other hand, geometric controllers are derived from kinematic models, where the steering angle is computed based on simple trigonometric equations, which significantly simplifies their implementation. While geometric controllers achieve acceptable tracking performance at low-to-medium velocities, their accuracy decreases as the vehicle velocity increases, which is due to ignored dynamics, large tire slip, and unmodeled nonlinearities that dominate at high speeds. In this work, we propose an adaptive pure pursuit controller that calculates a variable look-ahead distance based on curvature in a feedforward fashion, complemented by a regulated deviation (error-based) model in the feedback loop, which generates a corrective steering angle that accounts for nonlinearities at higher speeds. A comparative simulation study is presented to demonstrate the superiority of the proposed controller over the traditional pure pursuit controller, which is evidenced by 85.69% improvement in lateral position tracking, 19.75% improvement in yaw angle tracking, and 8% improvement in lap time.**

***Index Terms*—Adaptive Pure Pursuit, Autonomous Racing, Deviation Model, NXP-CUP, Trajectory Tracking**

I. Introduction

Autonomous vehicle control has been the focus of research for years, and has shown numerous developments since the demand for safe, reliable, and efficient navigation in all kinds of environments increases. Solid progress has been made in the motion control field, enabling vehicles to work effectively under different scenarios. However, challenges still remain in path tracking at the limits of handling. Numerous works have tackled the problem of motion control for autonomous vehicles. State-of-the-art controllers can be mainly categorized into two parts: model-based controllers, and geometric controllers.

Model-based controllers use mathematical models to predict the outcome of a potential control signal, where the performance of these signals is assessed through solving an optimization problem based on an objective function. For instance, Linear Quadratic Regulator (LQR) is a popular model-based design that is used due to its simplicity and effectiveness in minimizing a quadratic cost function that balances system state deviations and control effort. Liu *et al.* [1] used an LQR algorithm for lateral control of an autonomous car, and they employed a look-up table to accelerate the computation for varying longitudinal velocity.

Another widely used model-based controller is Model Predictive Control (MPC), which unlike LQR, introduces system constraints directly into the optimization problem. This makes it particularly effective in scenarios requiring adherence to safety and operational limits. Various adaptations of MPC have been proposed to address specific challenges. A tube-based MPC that combines a nominal model and a second-order super twisting sliding mode controller (STSMC) was designed in [2] for trajectory tracking under uncertainty. A. Jain *et al.* [3] proposed a nonlinear MPC based on the extended kinematic model, where unmodeled dynamics are compensated by using a Gaussian Process (GP), which iteratively refines the model for better approximation of the actual vehicle dynamics, and thus results in improving the tracking performance. G. Williams *et al.* [4] introduced the Model Predictive Path Integral Control (MPPI) algorithm, which is a sampling-based approach that leverages path integral control to optimize stochastic trajectories. R. Verschuere *et al.* [5] designed a nonlinear MPC with a minimum-time objective, putting much emphasis on rapid and efficient execution of the trajectory. M. Brunner *et al.* [6] proposed a Learning Model Predictive Control (LMPC) framework that leverages data from prior iterations to optimize the control performance over time. It guarantees better performance with each iteration and successfully applies the method to aggressive autonomous racing.

979-8-3315-2960-4/25/$31.00 ©2025 IEEE

Recent works addressed autonomous racing using AI based methods. For instance, K. G¨uc¸kıran and B. Bolat [7], compared two advanced Deep Reinforcement Learning (DRL) methods, which are Soft Actor-Critic (SAC) and Rainbow DQN, to develop near-optimal agents capable of high speed navigation. The study is conducted using TORCS simulator. Results proved that the SAC has superior performance in continuous action spaces, enabling smoother and more stable driving behaviour. This method relies on the availability of data, which is scarce in this application.

Other works have addressed vehicle control at the limits of handling using feedforward-feedback architectures. In [8] and [9], the feedforward part computes control inputs using a linearized vehicle dynamics model with nonlinear tire behavior. Kritayakirana and Gerdes [8] combined this with a full-state feedback controller. Kapania [9] developed a feedback controller that minimized tracking error at a look-ahead distance and included sideslip information in the feedforward component to further improve performance at the limits. On the other hand, Laurense *et al.* [10] developed a slip-angle-based framework for steering and longitudinal speed control in path tracking at the friction limit with a reduced dependency on high accuracy friction estimates.

While model-based controllers yield high performance and good tracking, they are computationally expensive, and cannot be implemented in a low-cost board, such as those used in small scale racecars. Geometric controllers, such as pure pursuit [11] and Stanley [12], demonstrate good performance at low speeds. However, since they are based on simplified kinematic models, their accuracy decreases at higher speeds where dynamic effects, such as inertia, tire slip, and external forces, play a more significant role in influencing the vehicle dynamics. To compensate for poor performance at higher velocities, several works have implemented an adaptive pure-pursuit algorithm, where they vary the look-ahead distance based on preset parameters or conditions. For instance, Huang *et al.* [13] implemented an adaptive pure pursuit (APP) controller, where the look-ahead distance is a linear combination of the longitudinal velocity and the curvature of the path, further stabilized using a PID controller to ensure smooth steering angle tracking. Other works, such as [14], propose generating the look-ahead distance using minimum and maximum bounds, weights that sum to one, and the deflection angle of the vehicle.

Park *et al.* [15] introduced an adaptive pure-pursuit algorithm with a piecewise linear look-ahead distance based on vehicle speed, incorporating a proportional-integral (PI) controller, with gains that are functions of the road curvature, to address corner cutting and tracking errors. In [16], they further improve the steering system of the car, by employing a servo motor that can transmit the steering forces through a belt and pulley. A lead-compensator is then designed along with a PID controller for the steering system, to accommodate for dead-band, which causes reduced path tracking and vehicle stability.

However, the adaptive pure pursuit algorithms discussed above were designed for general driving tasks, not for racing applications. Hence, they were not specifically tailored to handle the extreme operating conditions in high-performance racing scenarios, where the vehicles are constantly pushed to operate near the tire-road adhesion limits. The racing problem using adaptive pure pursuit was addressed by Becker *et al.* in [17], where they developed an adaptive pure pursuit controller that linearly adjusts the look-ahead distance based on the vehicle's velocity. To handle high-speed dynamics, they incorporated the Pacejka Magic Formula to model nonlinear tire slip and utilized a Lookup Table (LUT) to map desired lateral accelerations to the appropriate steering angles, thus ensuring precise trajectory tracking under challenging conditions. In experimentation, the track was known to the car, which allowed for the look-ahead distance to be as long as needed. In this work, the track is assumed to be unknown. Further details will be discussed in Section IV.

In this paper, we present an adaptive pure pursuit controller that computes a variable look-ahead distance based on the curvature of the desired path. To compensate for unmodeled nonlinearities and parametric uncertainties, an output feedback loop is implemented with an error-based deviation model that produces a corrective steering angle, in addition to the steering angle commanded by the APP controller, in order to compensate for the unmodeled vehicle dynamics. This controller design combines computational efficiency by using a geometric controller, with the ability to handle nonlinear vehicle dynamics via the feedback loop, which makes it suitable for computationally-constrained applications in autonomous racing.

The rest of the paper is organized as follows. In Section II, the vehicle dynamics model is introduced. Section III provides a basic understanding of pure pursuit (PP), adaptive pure pursuit (APP), the error-based deviation model regulator (DMR), and the proposed control scheme (APP+DMR). The control system design is validated via numerical simulations in Section IV. Finally, a conclusion and an outlook towards future work are provided in Section V.

II. Vehicle Dynamics Model

The planar vehicle dynamics are modeled using the bicycle model, which is illustrated in Fig. 1, where ℓ_f and ℓ_r are respectively the distances between the front and rear wheels to the vehicle's center of gravity (CG), δ is the steering angle at the front tire, β is the vehicle's side-slip angle, x and y are the longitudinal and lateral positions of the vehicle's body frame, X and Y are the absolute vehicle position in the inertial frame of reference, and F_{yf} and F_{yr} are the lateral tire forces of the front and rear wheels, respectively. In this work, the longitudinal dynamics are modeled as a first-order transfer function to capture the delay between the actual and desired longitudinal speeds, which influence the lateral dynamics by feeding the longitudinal speed (V_x) as a signal in the dynamic model.

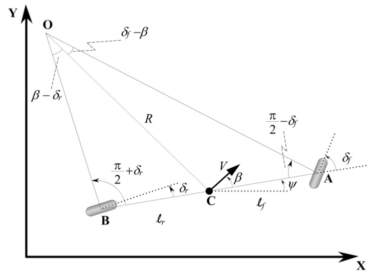


Fig. 1: The simplified vehicle dynamics bicycle model [18]

The lateral motion dynamics of the vehicle are given by [18]:

$$m(\ddot{y} + \dot{\psi} V_x) = F_{yf} + F_{yr}, \tag{1}$$

and the rotational (yaw) dynamics are given by:

$$I_z \ddot{\psi} = \ell_f F_{yf} - \ell_r F_{yr}. \tag{2}$$

A linear model is adopted for computing the front and rear tire lateral forces, F_{yf} and F_{yr}, which are given by [18]:

$$F_{y_f} = C_f \left(\delta - \frac{V_y + \ell_f \dot{\psi}}{\dot{x}} \right), \tag{3}$$

$$F_{y_r} = C_r \left(\frac{V_y + \ell_r \dot{\psi}}{V_x} \right). \tag{4}$$

Hence, the lateral dynamics model can be expressed in state-space format as follows [18]:

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}\ell_f - 2C_{\alpha r}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2\ell_f C_{\alpha f} - 2\ell_r C_{\alpha r}}{I_z V_x} & 0 & -\frac{2\ell_f^2 C_{\alpha f} + 2\ell_r^2 C_{\alpha r}}{I_z V_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2\ell_f C_{\alpha f}}{I_z} \end{bmatrix} \delta. \tag{5}$$

III. Controller Design

In this section, a basic description of the standard pure pursuit (PP) controller is discussed, followed by the design of an adaptive pure pursuit (APP) controller, and ending with the derivation and design of the proposed adaptive pure pursuit with deviation model regulation (APP+DMR) control system.

A. Adaptive Pure Pursuit

The pure pursuit controller is a geometric controller that commands the vehicle to follow a certain look-ahead point on the track.As seen in Fig. 2, a target point on the reference path is chosen at a look-ahead distance (l_d) [19]. An arc is fitted between the rear wheel and the target point to determine the relationship between the vehicle's position and the reference path [19].

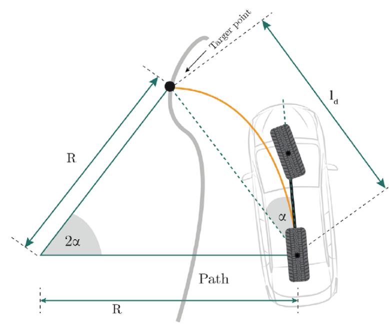


Fig. 2: Pure pursuit geometry for a bicycle vehicle model [19]

The feedforward steering angle δ_{FF} is calculated as [19]:

$$\delta_{FF} = \tan^{-1}\left(\frac{2l \sin\alpha}{l_d}\right), \tag{6}$$

where α is the angle between vehicle forward vector and the vector connecting the target point and the rear axle of the vehicle, it can be found as [19]:

$$\alpha = \sin^{-1}\frac{e_y}{l_d}. \tag{7}$$

Figure 3 illustrates the importance of having the correct l_d: if l_d is too small, the car would oscillate about the desired trajectory, whereas if l_d is too large, the vehicle would take longer to converge to the desired track or may eventually fail to do so.

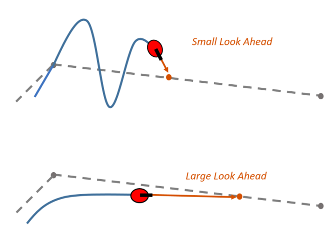


Fig. 3: Influence of l_d on path tracking [20]

Hence, it is critical to set appropriate values for l_d, which strike a balance between precision and fluctuation on curved (small l_d) and straight (large l_d) sections of the track. As such, in this work, we propose to dynamically calculate the look-ahead distance as follows:

$$l_d = \begin{cases} l_{d_{min}} & \text{if } l_d < l_{d_{min}} \\ l_{d_{max}} - k\rho & \text{if } l_{d_{min}} < l_d < l_{d_{max}} \\ l_{d_{max}} & \text{if } l_d > l_{d_{max}} \end{cases} \tag{8}$$

$$\frac{d}{dt}\begin{bmatrix} e_y \\ e_{\dot{y}} \\ e_\psi \\ e_{\dot{\psi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_\text{f}+2C_{\alpha r}l_\text{r}}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_\text{f}-2C_{\alpha r}l_\text{r}}{I_z V_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_\text{f}^2+2C_{\alpha r}l_\text{r}^2}{I_z V_x} \end{bmatrix} \begin{bmatrix} e_y \\ e_{\dot{y}} \\ e_\psi \\ e_{\dot{\psi}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_\text{f}}{I_z} \end{bmatrix}\delta + \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_\text{f}-2C_{\alpha r}l_\text{r}}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_\text{f}^2+2C_{\alpha r}l_\text{r}^2}{I_z V_x} \end{bmatrix}\dot{\psi}_\text{des} \tag{9}$$

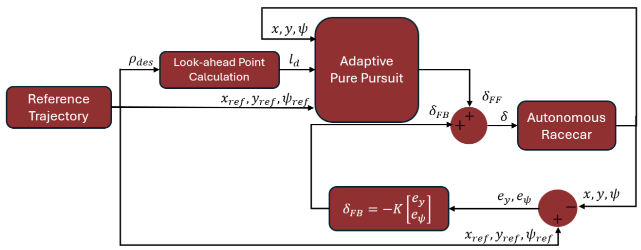


Fig. 4: Block diagram of the proposed APP+DMR controller

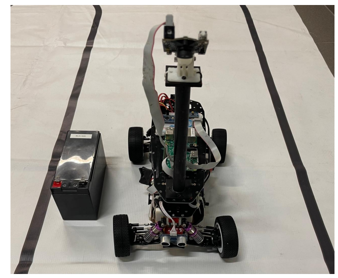


Fig. 5: MR-Buggy3 NXP Car

where k is a positive constant. It is noteworthy to mention that the look-ahead distance is constrained in a specified range of values since, in this work, we assume that the track is unknown, which is due to the fact that the sensory input from the environment around the racecar (perception subsystem) is limited to a certain range. When the racecar is on a straight section of the track, i.e. it is operating at maximum speed, the look-ahead point must be placed further from the car to prevent oscillations. Conversely, as the car approaches a corner, and the speed decreases, the look-ahead point decreases as the curvature increases, hence enabling the racecar to negotiate sharper turns.

B. *Deviation Model Regulator Design*

As previously mentioned, the pure pursuit algorithm is based on the kinematic bicycle model, which is only valid for operating in low-to-medium velocities. While the APP algorithm enhances the tracking performance of the vehicle at higher speeds, it still has inconsistencies due to unmodeled dynamics. Therefore, the error-based deviation model regulator system in the feedback loop generates a corrective steering angle, which is meant to compensate for those unmodeled nonlinearities that dominate at higher velocities. The error-based deviation model can be written as shown in Eq. (9) [18].

The system is regulated using an output feedback proportional controller, hence generating a control signal calculated as:

$$\delta_{FB} = -K\begin{bmatrix} e_y \\ e_\psi \end{bmatrix}. \tag{10}$$

Hence, the total control signal generated by the APP+DMR controller, as depicted in Fig. 4, is given by:

$$\delta = \delta_{FF} + \delta_{FB}. \tag{11}$$

IV. Simulation Results and Discussions

In this section, we provide a comparative study of the tracking performance for the proposed APP+DMR controller against the traditional pure pursuit and the adaptive pure pursuit algorithms. The root mean square (RMS) values of the lateral error e_y, the yaw angle error e_ψ, and lap time are chosen as the performance metrics. Numerical simulations were conducted using MATLAB using vehicle parameters that correspond to the MR-Buggy3 kit used in the NXP Cup (shown in Fig. 5), a 1:18 scale model car.

TABLE I: Vehicle Parameters

Parameter	Symbol	Unit	Value
Distance to front axle	l_front	[m]	0.12
Distance to rear axle	l_rear	[m]	0.16
Front cornering stiffness	C_front	[N/rad]	17.0
Mass of the vehicle	m	[kg]	1.36
Rear cornering stiffness	C_rear	[N/rad]	17.8
Yaw moment of inertia	I_z	[kg $\cdot$ m^2]	0.015

The reference path was designed using the Driving Scenario Design MATLAB Toolbox, which mimics a desired race line. A velocity profile was also generated for the given path, using the formula $V_x = \sqrt{\mu g \rho}$, where μ is the friction coefficient (assumed to be unity), and g is the gravity constant. To increase the simulation fidelity, the vehicle's longitudinal dynamics were modeled as a first-order transfer function that captures the relationship between the desired speed, V_{x_des}, and the actual speed of the racecar, V_x. The autonomous racecar is equipped with a servo motor that turns from 0 to 60° in $0.05s$. Therefore, a $0.0375s$ time delay was employed to account for the steering actuator dynamics, and the steering angle was constrained to a maximum of $\frac{\pi}{4}$.

The NXP Cup is an autonomous racing competition, where the autonomous racecar has to complete a lap with

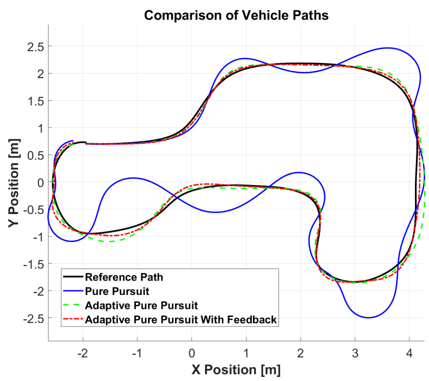


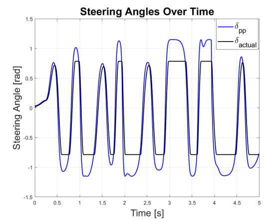
Fig. 6: Comparison of the path tracking performance for PP, APP, and APP+DMR

the fastest time without any previous knowledge of the track. Therefore, the challenge in the competition lies in the fact that the track is unknown to the contestants a priori, hence the knowledge of the track is limited to sensory input. In the presented buggy kit, the perception system (vision-based Pixy camera) is limited between $0.25m$ and $1m$ ahead of the racecar. Therefore, the thresholds $l_{d_{min}}$ and $l_{d_{max}}$ for the look-ahead distance of the adaptive pure pursuit algorithm were set to 0.25 and 1.0, respectively.

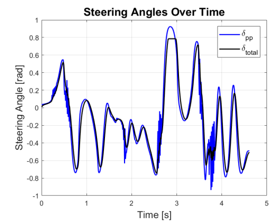
Figure 6 illustrates the different trajectories followed by the vehicle when using different controllers. It can be seen that the pure pursuit algorithm fails to perfectly track the path. When tuning the look-ahead distance, a compromise has to be made between avoiding oscillations at high speeds and converging fast enough to the track. In the given complex path with multiple consecutive corners, the pure pursuit algorithm failed to track the path since the required speed was higher than what the vehicle could handle. The APP presented better performance, since the look-ahead distance was changing along with the curvature of the path, hence enabling sharper turns while avoiding oscillations in the straights. The proposed APP+DMR controller shows even better performance since it generates additional corrective steering angles that compensate for the unmodeled nonlinearities and parametric uncertainties.

Figure 7 shows the different steering angles computed by the three controllers along the given path. It can be observed that APP exhibits fewer oscillations compared to pure pursuit, with an even further reduction when using DMR+APP. In Figure 7 (c), it is clear that the feedback controller is generating only small corrective steering angles, allowing for more aggressive corrections, and hence smaller errors and more precise path tracking. Fig. 8 shows that the APP+DMT controller was still able to achieve speeds that are comparable to PP (up to $7m/s$), which indicates that the racing performance was not compromised by the tighter trajectory tracking.

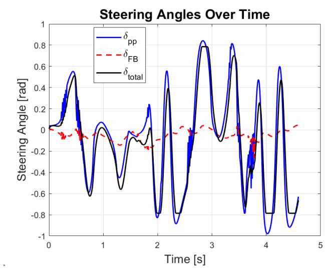
Table II shows the performance metrics for each of the controllers. The APP+DMT controller shows a 85.69% improvement in lateral tracking, a 19.75% improvement in yaw angle tracking, and 8% improvement in lap time, with respect to the traditional pure pursuit algorithm. However, it can be seen that APP+DMR exhibits larger e_{ψ}^{μ} than APP, which is due to its aggressive corrections that yield a 31% improvement in lateral tracking, which is a desirable feature in racing with limited vision to keep up with the optimal race line.



(a) PP steering angles



(b) APP steering angles



(c) APP+DMR steering angles

Fig. 7: Comparison of steering angles across different methods.

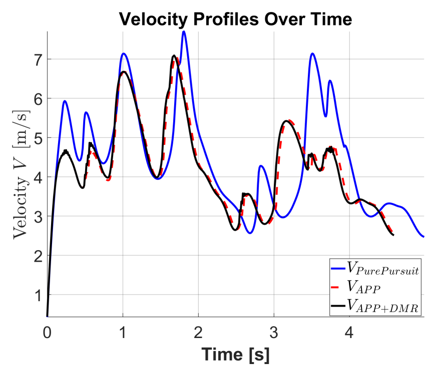


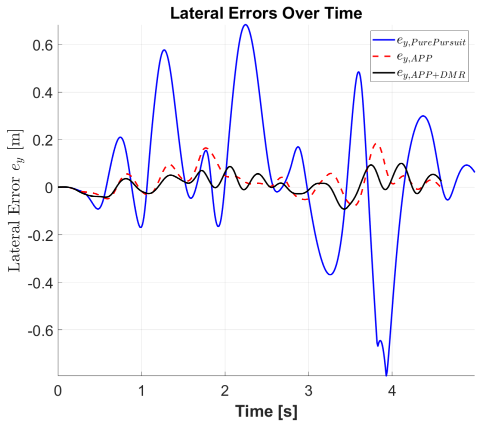
Fig. 8: Velocity profile while using different controllers

TABLE II: Performance Comparison of Controllers

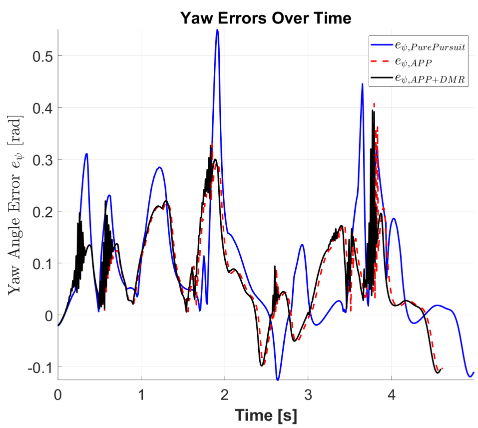
Controller	Lap Time [s]	$\mathbf{e_y^{\mu}}$ [m]	$\mathbf{e_y^{max}}$ [m]	$\mathbf{e_{\psi}^{\mu}}$ [rad]	$\mathbf{e_{\psi}^{max}}$ [rad]
Pure Pursuit	4.99	0.2852	0.7944	0.1463	0.5503
APP	4.61	0.0592	0.1846	0.1169	0.4081
PP+DMR	4.59	0.0408	0.1004	0.1174	0.3952

V. Conclusion and Future Work

This paper presented the design of a lateral motion controller for small-scale autonomous racecars. The proposed



(a) Lateral error



(b) Yaw angle error

Fig. 9: Lateral and Yaw angle errors for PP, APP, and APP+DMR

controller combines an adaptive pure pursuit controller in a feedforward configuration with an error-based deviation model regulator system in the feedback loop.

The look-ahead distance in the pure pursuit controller is calculated based on the path curvature to generate a steering angle command. The feedback regulator system generates a corrective steering angle that compensates for the unmodeled nonlinearities that dominate at high speeds.

Future work includes the development of a longitudinal controller that complements the proposed lateral controller, creating a MIMO system that allows both lateral and longitudinal trajectory tracking. Also, the proposed controller is to be experimentally validated on a low-cost board of the NXP MR-Buggy3 kit.

REFERENCES

[1] Z. Liu, D. Le, K. Zhang and B. Zhang, "Real-time Motion Control with Iterative Optimization and Robustness Analysis for Autonomous Driving," 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, China, 2019, pp. 1385-1390.

[2] H. El Atwi and N. Daher, "A Composite Model Predictive and Super Twisting Sliding Mode Controller for Stable and Robust Trajectory Tracking of Autonomous Ground Vehicles," 2021 IEEE 3rd International Multidisciplinary Conference on Engineering Technology (IMCET), Beirut, Lebanon, 2021, pp. 107-112.

[3] A. Jain, M. O'Kelly, P. Chaudhari, and M. Morari, "BayesRace: Learning to race autonomously using prior experience," arXiv preprint, arXiv:2005.04755, 2020.

[4] G. Williams, P. Drews, B. Goldfain, J. M. Rehg and E. A. Theodorou, "Aggressive driving with model predictive path integral control," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1433-1440.

[5] R. Verschueren, M. Zanon, R. Quirynen and M. Diehl, "Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm," 2016 European Control Conference (ECC), Aalborg, Denmark, 2016, pp. 141-147.

[6] M. Brunner, U. Rosolia, J. Gonzales and F. Borrelli, "Repetitive learning model predictive control: An autonomous racing example," 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 2017, pp. 2545-2550.

[7] K. Güçkıran and B. Bolat, "Autonomous Car Racing in Simulation Environment Using Deep Reinforcement Learning," 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 2019, pp. 1-6.

[8] K. Kritayakirana and J. C. Gerdes, "Using the centre of Percussion to design a steering controller for an autonomous race car," Vehicle System Dynamics, vol. 50, no. sup1, pp. 33–51, Jan. 2012.

[9] N. R. Kapania, "Trajectory Planning and Control for an Autonomous Race Vehicle." Order No. 28118585, Stanford University, United States – California, 2016.

[10] V. A. Laurense, J. Y. Goh and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," 2017 American Control Conference (ACC), Seattle, WA, USA, 2017, pp. 5586-5591.

[11] Tech. Report, CMU-RI-TR-09-08, Robotics Institute, Carnegie Mellon University, February, 2009.

[12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," in *The 2005 DARPA Grand Challenge: The Great Robot Race*, M. Buehler, K. Iagnemma, and S. Singh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–43.

[13] Y. Huang, Z. Tian, Q. Jiang and J. Xu, "Path Tracking Based on Improved Pure Pursuit Model and PID," 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT, Weihai, China, 2020, pp. 359-364.

[14] S. Qinpeng, W. Zhonghua, L. Meng, L. Bin, C. Jin and T. Jiaxiang, "Path Tracking Control of Wheeled Mobile Robot Based on Improved Pure Pursuit Algorithm," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 4239-4244.

[15] M. -W. Park, S. -W. Lee and W. -Y. Han, "Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm," 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Gyeonggi-do, Korea (South), 2014, pp. 1443-1447.

[16] M. Park, S. Lee, and W. Han, "Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm," ETRI Journal, vol. 37, no. 3, pp. 617–625, Jun. 2015.

[17] J. Becker, N. Imholz, L. Schwarzenbach, E. Ghignone, N. Baumann and M. Magno, "Model- and Acceleration-based Pursuit Controller for High-Performance Autonomous Racing," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 5276-5283.

[18] Rajamani, Vehicle dynamics and control. 2011.

[19] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Review and Performance Evaluation of path tracking controllers of Autonomous Vehicles," IET Intelligent Transport Systems, vol. 15, no. 5, pp. 646–670, Mar. 2021.

[20] MathWorks, "Pure pursuit controller," MATLAB & Simulink, [Online]. Available: https://www.mathworks.com/help/nav/ug/pure-pursuit-controller.html. [Accessed: Dec. 6, 2024].