Context and Orientation Aware Path Tracking

Nicholas Michael Bünger*[1], Sahil Panjwani*[2,3], Malika Meghjani*[2,3],
Zefan Huang[3], Marcelo H. Ang Jr.[4] and Daniela Rus[5]

***Abstract*— Autonomous vehicles on city roads and especially in pedestrian environments require agility to navigate narrow passages and turn in tight spaces, leading to the need for a real-time, robust and adaptable controller. In this paper, we present orientation and context aware controllers for autonomous vehicles that can closely track the reference path wit alh respect to the current state of the vehicle, environmental properties, and the desired target orientation at the desired target location. Our proposed controllers are derived from the widely used pure pursuit controller. We validate our proposed controllers with respect to the baseline pure pursuit controller in simulation and on a full-size autonomous vehicle in a pedestrian environment. Our experimental results suggest significant improvements in adaptability and tracking performance compared to the pure pursuit controller.**

I. INTRODUCTION

Urban environments such as city roads and pedestrian walkways present challenging problems for autonomous vehicles to navigate on narrow pathways, maneuver sharp turns and park in tight spaces. This requires autonomous vehicle controllers to be adaptable in real-time to a diverse range of scenarios while attempting to realize the desired path and most importantly the desired target orientation. Traditionally, geometric path tracking algorithms have been used for motion control of autonomous vehicles. These algorithms primarily derive a geometric relation between the vehicle pose and its reference path without weighing in the kinematics and dynamics of the vehicle. The simplicity, soundness, and low computational cost of these algorithms contribute to their prevalence in autonomous vehicles for path tracking.

The earliest and most routinely used path tracking controller is the pure pursuit controller [1] which owes the name to its conceptualisation. Given that an autonomous vehicle is tasked to chase a moving point on a reference path based on its current location, it is said to be in a constant 'pursuit' of the goal-point, where the goal-point is selected based on a fixed look-ahead distance. This is comparable to the well known 'follow-the-carrot' controller [2], wherein the autonomous vehicle is directly steered towards a chosen carrot-point within one look-ahead distance. In contrast, the pure pursuit controller entails calculating a curvature to be followed between the current location of the vehicle and its instantaneous goal-point. As a consequence, an angular velocity is calculated for the autonomous vehicle to attain its instantaneous goal at a certain look-ahead distance from it. The process is repeated until the vehicle reaches the target goal point on its reference path.

The pure pursuit controller is computationally simple and robust to large disturbances. However, given the controller's dependency on only the basic attributes of the geometric model, it gives rise to certain weaknesses. These are mainly attributed to the improper selection of the look-ahead distance which results in poor path tracking performance. Specifically, the controller lacks in following a given reference path due to the constant need to tune the look-ahead distance to optimise its performance and converge to the path. Thus, leading to steering latency. This further results in cutting corners, as the controller does not explicitly consider the actual curvature of the path.

In this paper, we propose two novel path tracking controllers which not only achieve the desired path, similar to the pure pursuit controller, but also obtain the target orientation at each point. The contributions of this paper are: (a) a novel orientation-aware path tracking controller which accounts for the required target orientation, (b) a novel context-aware path tracking controller with dynamically varying look-ahead distance based on the road contextual information and the vehicle state, (c) a theoretical analysis of the proposed orientation-aware controller, and (d) a validation of our proposed controllers in challenging simulated and real-world scenarios and their comparison with the baseline pure pursuit controller.

II. RELATED WORK

The geometric controllers, such as pure pursuit [1] and follow-the-carrot [2] account only for the desired target position in order to determine the vehicle motion for path tracking. Unlike these controllers, the vector pursuit controller [3] and proportional path tracking controller [4] consider the required target orientation in addition to the required target position. Specifically, vector pursuit controller is based on the theory of screws [5] to provide the desired turning radius based on the vehicle's current position and orientation. It has been observed that the vector pursuit controller is more robust and accurate in path tracking performance than the

*authors contributed equally.
Nicholas M. Bünger is with the [1]Swiss Institute of Technology, Zürich, Switzerland buengern@student.ethz.ch

Sahil Panjwani and Malika Meghjani are with the [2]Singapore University of Technology and Design, [3]Singapore-MIT Alliance for Research and Technology, Singapore sahil_panjwani@sutd.edu.sg, malika_meghjani@sutd.edu.sg

Zefan Huang is also with [3]Singapore-MIT Alliance for Research and Technology, Singapore huang.zefan@smart.mit.edu

Marcelo H. Ang Jr. is with the [4] National University of Singapore, Singapore mpeangh@nus.edu.sg

Daniela Rus is with the [5]Massachusetts Institute of Technology, Cambridge, MA, USA rus@csail.mit.edu

traditional pure pursuit controller. However, the controller's performance is traded-off with the computational complexity of the algorithm. Similar to vector pursuit and proportional path tracking controllers, our proposed controllers also incorporate orientation information. In addition to being robust to the sensitivity of the look-ahead distance and the ability to handle sudden large position and heading errors, our proposed algorithm is computationally tractable and therefore suitable for real-time implementation.

The performance of the aforementioned geometric controllers are highly dependent on the appropriate selection of the look-ahead distance. Thus, several variants [6]-[13] of the pure pursuit algorithm have been proposed to primarily address the problem of appropriate selection of the look-ahead distance based on different independent parameters such as vehicle's velocity and curvature of the road. Compared to these studies, our work proposes to dynamically select the look-ahead distance by accounting for both the vehicle status and environmental properties to guide the controller for a more steady and conformable path tracking solution.

In addition to the geometric controllers, there exist other classes of controllers such as dynamic, model-based, and classic controllers [14]. The dynamic controllers [15]-[16] account for the dynamic effects i.e., internal forces, energy, or momentum within the system to describe the motion of the vehicle. This necessitates the need for costly and dedicated sensors or extra data-processing steps for integration of dynamic feedback. Also, similar to the pure pursuit controller, the dynamic controller performance is dependent on the appropriate selection of the look-ahead distance. The model-based controllers have garnered growing interest as they consider whole vehicle model to derive its control law [17]. The recent developments of novel model-predictive controllers based on time-elastic-bands [18] have the added flexibility of temporal discretization for time-optimal control. Such approaches have improved the run-time performance over the conventional model-based controllers at the trade-off of accuracy. Lastly, the well-known classical controllers such as PID [19]-[20] and sliding mode controllers [21]-[22] are known for their simplicity in design, however, they require parameter-optimisation for different operating conditions. Also, they are often vulnerable to chattering and dead bands which further influence their performance and stability [23].

Each of the aforementioned path tracking algorithms have their respective strengths and weaknesses. The selection of an appropriate algorithm is influenced by the environmental conditions, computing resources, and controller parameters that are required to be regulated. An adaptive geometric controller such as the one proposed in this paper offers a simple, efficient, and computationally tractable solution by virtue of an easy implementation and regulation of basic state variables. Therefore, the focus of this work is on geometric controllers and the proposed controllers will be bench-marked against the original pure-pursuit controller.

III. Pure Pursuit Controller

We propose two path tracking controllers which overcome the drawbacks of the classic *Pure Pursuit* (PP) controller. In order to introduce the contributions of our proposed controllers, we first analyse the PP controller.

Given the current state vector $X(t)$ and the target state vector $X_{ref}(t)$, the objective of the PP controller is to find the control vector, $u(t)$ such that $X(t) - X_{ref}(t)$ is minimized. We define, $X(t) = [x_t, y_t, \theta_t]$ as the state vector representing the pose of the vehicle and $u(t)$ as the one dimensional control vector representing the steering angle, $\delta(t)$. Thus, $u(t) = \delta(t)$ where, $\delta(t)$ is constrained by the maximum allowable steering angle, such that, $-\delta_{max} \leq \delta(t) \leq \delta_{max}$. For the sake of simplicity, the dependency of time will not be specified in the following equations. We first analyze the constrained capabilities of the PP controller based on the following two observations.

Observation 1: The PP algorithm outputs a unique curvature based on its control law to reach a specific point on the path but it does not consider the actual curvature of the path.

The PP algorithm tends to follow a point regardless of the orientation i.e. it follows a position rather than a pose. This can become particularly problematic when high curvatures have to be pursued along the path, as it would happen often for vehicles that are traveling in pedestrian environments where sudden changes of trajectories (in form of high curvature bents) would be required for avoiding pedestrians that might appear on the way. The PP algorithm geometrically plans for the target point (x_{ref}, y_{ref}) with given steering angle δ and a constant wheelbase, L, using a bicycle model, as represented in Eq. 1 and illustrated in Fig. 1a.

$$\delta = tan^{-1}\left(\frac{2Lsin(\eta)}{l_{ahead}}\right) \tag{1}$$

where, l_{ahead} is the look-ahead distance between the current position of the vehicle (x, y) and the target point (x_{ref}, y_{ref}). η is the look-ahead angle, i.e. the angle between the current heading of the vehicle and the line-of-sight direction of the look-ahead target point. The look-ahead angle, η and distance, l_{ahead} are the only two variables required by the PP algorithm.

In Fig. 1a, the current heading is straight ahead at the current position of the vehicle (x, y) and redrawn at the target point (x_{ref}, y_{ref}) for reference. The current position of the vehicle (x, y) is set to intersect with the center of rear axle. The target orientation annotated at the target point represent the required orientation of the vehicle after following the motion along the arc length. With a constant steering angle, δ, the vehicle will consequently follow the arc connecting the current and target points, having radius R, represented by Eq. 2, where γ is the curvature of the path.

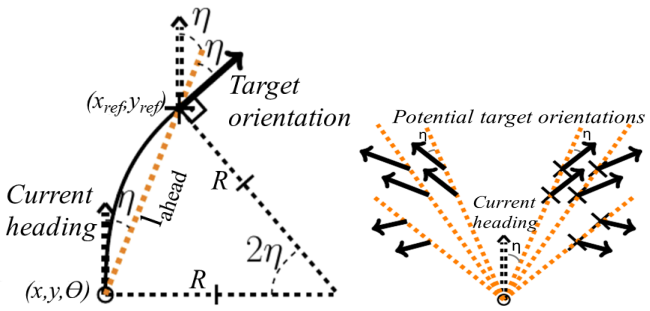
$$R = \frac{1}{\gamma} = \frac{l_{ahead}}{2sin(\eta)} \tag{2}$$

An analysis of angles is fundamental for understanding the structure of this geometry. First, the closed triangle is

isosceles as two sides have the same length R. The arc connecting (x, y) with (x_{ref}, y_{ref}) will always have an internal angle 2η given the geometry of the problem. Lastly, the angle between the direction of look-ahead distance and the target orientation of the vehicle is also η. This means that the difference in angle between current heading and target orientation will be equal to 2η. Based on the geometrical analysis illustrated in Fig. 1a, an important property of the algorithm can be observed which we describe below.

Observation 2: Any point lying on a straight line starting from the current position is reached with the target orientation angle 2η with respect to the current heading, if controlled by *PP algorithm*.

The consistency of the potential target orientation angles defined by *Observation 2* reveals how the geometry of the algorithm works. A graphical illustration of the same is presented in Fig. 1b. This characteristic of the PP algorithm confirms *Observation 1*. Specifically, the PP algorithm allows us to *apriori* identify the orientation of each potential target point achievable from the current location of the vehicle which is independent of the curvature of the path. In fact, the *Observation 2* shows that according to PP algorithm, each point has its foretold orientation defined exclusively by η. Hence, we prove that *Observation 2* induces *Observation 1*.



(a) Geometry (b) Property

Fig. 1: Pure Pursuit controller

IV. PROPOSED APPROACH

We apply the observations from Section III to derive two novel path tracking algorithms: (a) orientation aware pure pursuit (OPP) and (b) context aware pure pursuit (CPP). Consider rotating the heading direction illustrated in Fig. 1b by a certain angle. This action will change the target orientation of the target point. If the target orientation of the arc obtained from the PP algorithm and the desired target orientation do not match, an alternative current heading would guarantee that the right curve would meet both the requirements of matching the coordinates and the orientation of the reference point. It is, however, important to emphasize that non-holonomic constraints should not be violated.

Fig. 2 shows an optimal heading which would allow the vehicle to not only reach the reference point (x_{ref}, y_{ref}), but also to match the target orientation θ_{ref}. The general intuition while visualizing oneself in the current heading position and being asked to reach the reference point with a specific target orientation, would be to spontaneously first

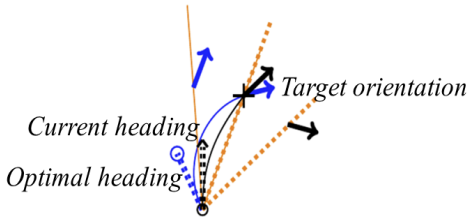
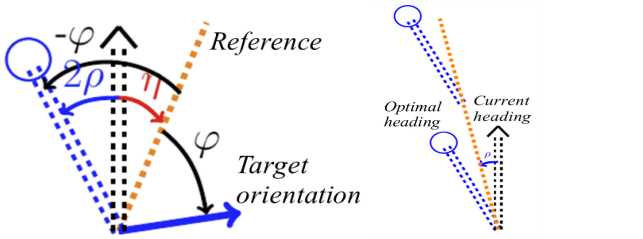


Fig. 2: Illustration of curvature required for target orientation.

turn left in direction of optimal heading and then right in order to achieve the target pose. This thought has to be defined mathematically using a systematic control law. Hence, we propose an orientation-aware controller.

A. Orientation-Aware Pure Pursuit Controller

Given the target orientation, if the ideal blue arc path in Fig. 2 is too late to be achieved, then we can use *Observation 2* for defining the optimal heading.



(a) Orientation analysis (b) New look-ahead angle

Fig. 3: Orientation-aware controller

The key point for achieving the optimal heading for the new desired target orientation is based on first observing that the line connecting the current position and look-ahead point in Fig. 3a is constant and can be considered as the reference. We introduce a new variable φ which is the angle between the reference and the new desired target orientation. Considering *Observation 2*, we can determine the angle between the current and the optimal headings as 2ρ, given by Eq. 3 and illustrated in Fig. 3a.

$$2\rho = -\varphi - \eta \tag{3}$$

Fig. 3b shows the optimal heading and the new look-ahead angle ρ with respect to the reference.

B. Control law

We derive a general control law to obtain the steering angle, δ, for the given current poses (x, y, θ) and reference poses $(x_{ref}, y_{ref}, \theta_{ref})$. The optimal heading is obtained based on the observations, from Fig. 1b and Fig. 3a, such that the target orientation can be reached at any point lying on a particular straight line which has the new look-ahead angle, ρ. The correction for the optimal heading will happen when the new look-ahead distance, l_{new}, is chosen. Consequently the *modified control law* is given by Eq. 4 where l_{new} has to be tuned.

$$\delta = tan^{-1}\left(\frac{2Lsin(\rho)}{l_{new}}\right) \tag{4}$$

The l_{new} parameter represents the intensity of the current trajectory's correction. If l_{new} is small, it means that the current position will be corrected much earlier in order to reach the target point with the target orientation. This produces more oscillations on the movement of steering wheel with the same principle as the PP algorithm. A bigger l_{new} makes the system more stable having fewer oscillations but will make it converge to the ideal position with an increased number of iterations while cutting corners. We empirically observe that there is a specific ideal ratio between l_{new} and l_{ahead}. This will require us to tune only one parameter, l_{ahead}, similar to the PP controller. In contrast, the angle ρ is unique for the current state vector (x, y, θ) and the target state vector, $(x_{ref}, y_{ref}, \theta_{ref})$, as presented in Eq. 3.

C. Context-aware Pure Pursuit Controller

The performance of geometric path tracking controllers depends on the appropriate selection of the look-ahead distance such that it can dynamically instruct the vehicle about the extent to track the reference path. Hence, the look-ahead distance needs to be adaptable to a variety of roads and vehicle states.

We propose a novel scaling function for dynamically selecting the look-ahead distance for an improved compliance, maneuverability, and steadiness. This variable look-ahead function helps to overcome the shortcomings of a constant look-ahead distance. In addition, it attenuates the over-sensitivity of the orientation-aware algorithm for avoidable oscillations along its path. Therefore, we keep the desirable attributes of both the PP and OPP controllers while minimising their associated inconveniences. In order to make the look-ahead distance adaptable, we propose a dynamic l_{new} distance referred as $l_{\text{new\_var}}$ which is determined using six parameters. These parameters are maximum deceleration for braking distance (BD), vehicle's velocity (v), vehicle's steering angle (δ), heading difference angle (2ρ) from the reference path, cross-track error (e), and curvature of the road (ω) being traversed. The $l_{\text{new\_var}}$ is calculated using Eq. 5 where k_1, k_2, k_3, k_4, and k_5 are tuning parameters with the first one being positive and others negative.

$$l_{\text{new\_var}} = k_1 v + BD + (k_2\delta + k_3\omega + k_4 e + k_5(2\rho)) \quad (5)$$

The value of k_1 was fixed at 0.7 for all simulation and real-world experiments based on the empirical analysis of convergence to the reference path and vehicle stability. The braking distance (BD) was measured conservatively at a maximum deceleration of $0.25g$ for the upper limit of the operating velocity while testing in simulations and on a full size autonomous vehicle. The maximum values of all the negative tuning parameters (k_2, k_3, k_4, k_5) that moderate the look-ahead distance were determined such that they always contribute a net value equal to the vehicle's braking distance, which in turn hinges on maximum deceleration to traverse the path safely. The real time values for the above tuning parameters (k_2, k_3, k_4, k_5) were calculated proportionately to their maximum computed values and current steering angle (δ). The velocity (v) was made variable as a function of steering, with higher values being applied for smaller steering angles and lower ones for sharper steering inputs. This helps the planner slow the vehicle down preemptively for a sharp turn. The speed is further smoothed with a first-order low-pass filter to filter out any erroneous measurements for preventing slipping and dynamic roll-overs and inducing better stability to the vehicle's motion. It thereby helps minimise abrupt transitions of linear and angular accelerations and improve continuity in motion. Therefore, as a consequence of Eq. 5, the controller looks farther onto the reference path when the vehicle speeds up as it drives the positive factor higher and looks closer when the steering input, road curvature, cross-track error, and/or heading difference increase as the net negative value of $l_{\text{new\_var}}$ grows. These additional tuning parameters allow us a more configurable and adaptable controller in unstructured environments and altering traffic conditions as illustrated both qualitatively and quantitatively in the subsequent sections.

V. EXPERIMENTS

We validate our proposed path tracking algorithms in simulation as well as on a full size autonomous vehicle. Specifically, we compare the OPP and CPP with the original PP controller as the baseline method. The input to the controller is the reference path and the outputs of the controller are throttle and steering commands to guide the vehicle both laterally and longitudinally. The metrics used for validation of the controller are the cross-track error of the vehicle relative to the reference path and angular jerks experienced while negotiating the curved path segments. The cross-track error is used to validate the safety of the controller and the jerk for the ride comfort. The stability of the controller is assessed with respect to the oscillations resulting from the changes in steering angle which is also quantified in terms of cross-track error and angular jerks.

A. Simulation Results

We validate the robustness and adaptability of our proposed controllers on two significantly challenging paths that require the vehicle to maneuver discontinuous paths which are specifically challenging for the PP controller. These tracks are U-shaped and 8-shaped tracks, as seen in Fig. 4. They are developed based on inspiration from human car driving license test [24].

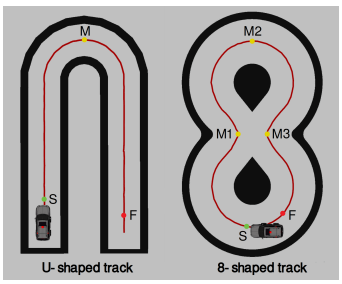


Fig. 4: Test tracks annotated with start point (S), middle points (M, M1, M2, M3) and finish point (F)

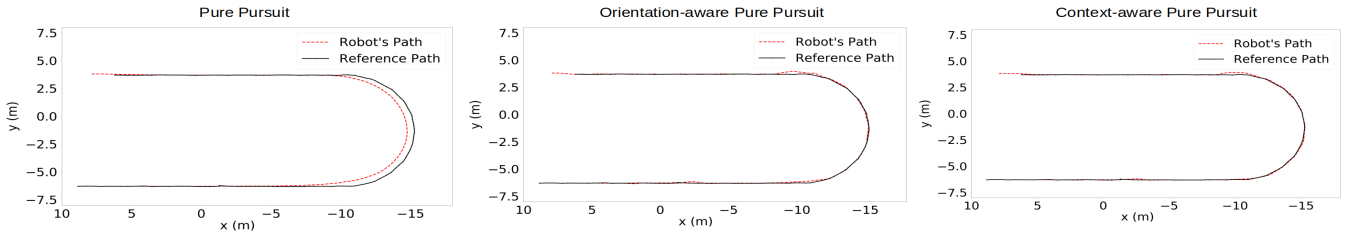


Fig. 5: Performance Comparison on U-shaped track

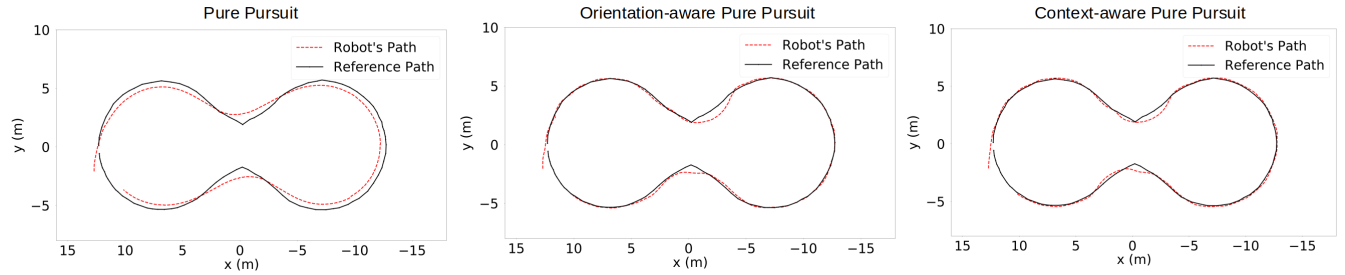


Fig. 6: Performance Comparison on 8-shaped track

Unlike the standard 8-shaped track that requires a cross-over point, we require the vehicle to keep on the same side of the path, which leads to a demand on the steering to transition from one extreme yaw angle to the other within an unusually tight space. This helps us in assessing the agility of the controller to negotiate demanding tracks which is critical for safe navigation around obstacles.

We build the simulation environment using ROS and Gazebo [25]. We developed a modified CATvehicle model [26] to simulate an autonomous vehicle. The simulated vehicle is four-wheeled, enabled with Ackerman steering based on Ford Escape model, as observed in Fig. 4. The simulated vehicle and the real autonomous electrical golf buggy are both controlled through ROS. The braking distance was calculated to be 0.45m. We performed three iterations for each track and their mean shape profiles along with the mean cross track error and jerk values are reported in the following sections.

1) Shape Profile: We qualitatively validated the controllers by analyzing the shape of their respective tracked paths on the two aforementioned test tracks. The tracked paths by the individual controllers are overlaid on their respective reference paths as shown in Fig. 5 and Fig. 6. The PP controller performed well on the straight sections of the path but while negotiating a turn, both on U-shaped and 8-shaped tracks, it tended to cut corners and consistently deviated away from the ideal path throughout all the curved sections. Whereas, the OPP performed a lot better, maintaining its followed path along the curved sections as it considers the required pose at each point on the path. However, the controller deviated while maneuvering the changeover points of yaw angle at middle points, M1 and M3 in Fig. 4 of the 8-shaped track, which is also the most crucial and difficult juncture in the entire track. Also, for the U-shaped track, the controller strays a little to the left just before engaging with the high curvature towards the right near the middle point, M. This is due to the control law from Eq. 4. The CPP controller recorded a similar performance to the OPP controller along the curves. It showcased the best overall path tracking behavior while staying the longest on the path and deviating the least.

2) Cross-track error: The next study was for quantifying the lateral deviations of the vehicle from its reference path. We measure the integrated area of deviation and the maximum variation from the reference path using *ImageJ* software [27], giving us mean and maximum cross-track error as a performance metric. Table I demonstrates a significant reduction in the cross-track error from PP to OPP controller. The CPP controller further reduces the cross-track error. The improvement in maximum deviation using CPP is much more evident on 8-shaped track given its combination of orientation-awareness and contextual-awareness, rendering it proficient in tracking a variety of paths more effectively. The CPP controller offers good lateral and longitudinal adaptability to different path conditions and its performance is consistent regardless of the test track, unlike the PP controller [28]. A controller with low cross-track error is most valued in highly constrained environments involving sharp turns or multiple obstacles where it would be desirable for the vehicle to not cross the defined operational environment boundaries or bump into obstacles.

3) Varying look-ahead: We measure the variation in cross-track error with an increase of the look-ahead distance for the PP and OPP algorithms which use fixed look-ahead distance. We used the '8' shaped test track as it was more challenging compared to the 'U' shaped track.

As it can be observed from the plot in Fig. 7, although the cross-track error increased significantly for both controllers with an increase in look-ahead distance, the rate of increase of the error with look-ahead distance is noticeably higher in

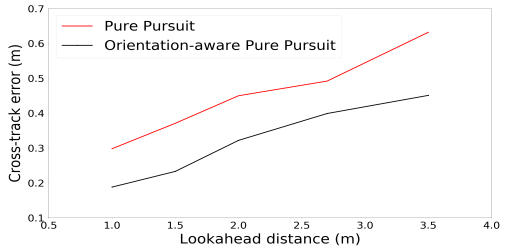


Fig. 7: Varying look-ahead on 8-shaped track

the case of PP. This indicates a reduced dependence of OPP on the look-ahead distance. This study also demonstrates the need for a smarter and dynamic look-ahead which contextualises both the vehicle and the environment to make its path tracking decisions.

4) Jerks: Lastly, the study to quantify ride quality involved measuring the angular jerks by virtue of steering changes for a comparison between OPP and CPP. As observed from Table I, OPP reported a maximum jerk value of 2.279 m/s^3 whereas CPP showcased an improvement in ride comfort with a reported maximum jerk value of 1.03 m/s^3 on the figure of 8. Although the maximum jerk here sees an increase from PP to CPP, this was only observed along high curvatures while staying near the comfort threshold [29]. The increase in the jerk value along the curve was observed due to a higher compliance to the actual path curvature unlike in PP where the actual curvature is not considered.

Controller	Max. Error (m)		Mean Error (m)		Max. Jerk (m/s^3)	
	Utrack	8track	Utrack	8track	Utrack	8track
PP	0.581	0.983	0.161	0.338	0.519	0.693
OPP	0.276	0.955	0.031	0.105	1.3	2.279
CPP	0.249	0.688	0.019	0.101	0.621	1.03

TABLE I: Performance study for simulations

B. Field trials

The real-world experiments are implemented on an autonomous golf buggy [30] developed by Singapore-MIT Alliance for Research and Technology (SMART) which is shown in Fig. 8. The vehicle is equipped with two LiDARs, one Inertial Measurement Unit (IMU), and wheel encoders for pose estimation and is localised using the Adaptive Monte Carlo Localisation [31]-[32]. The golf buggy has an on-board computer with Intel Core i7 processor, 16 GB RAM and Solid State Drives. All experiments reported in this section were executed on it in real-time.

Fig. 8: Autonomous golf buggy

The vehicle was tested in autonomous mode with a remote safety driver in a pedestrian environment of a university campus. The shape profile and cross-track error are presented in Fig. 9 and Table II, respectively. The cross-track error is observed to be the largest for PP and comparable for the other two controllers whereas the maximum angular jerk is observed to be the highest for OPP controller at 2.05 m/s^3, followed by 0.92m/s^3 and 0.672m/s^3 for CPP and PP controllers, respectively. Though the maximum angular jerk is the least for the PP controller, it narrowly escaped an obstacle by a margin of 8.3 cm distance. The obstacle is illustrated by a red dot in Fig. 9 toward the beginning of the turning path. In comparison, OPP and CPP controllers allowed the test vehicle a distance of 31.1 cm and 33.7 cm respectively helping it maneuver around that segment more safely.

These results are qualitatively consistent with the simulation results. There is improvement in the cross-track error from PP to OPP controller and in jerks from OPP to CPP. A compilation of the results presented in this section are provided at the following online video link, `https://youtu.be/Q_T_H6-yhiM`.

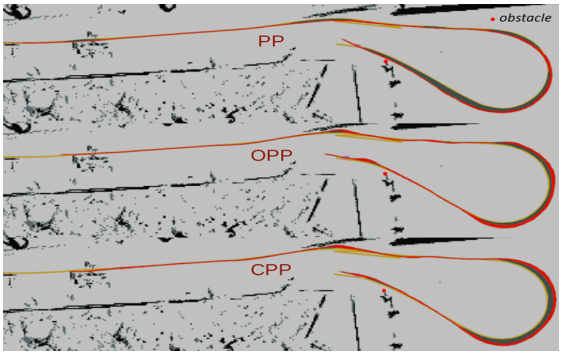


Fig. 9: Shape profiles for PP, OPP, and CPP respectively. yellow: reference path, red: tracked path, shaded: cross-track error

Controller	Max. Error (m)	Mean Error (m)	Max. Jerk (m/s^3)
PP	0.417	0.105	0.672
OPP	0.252	0.049	2.05
CPP	0.361	0.067	0.92

TABLE II: Performance study with real world data

VI. DISCUSSION AND CONCLUSION

We proposed two novel path tracking algorithms, OPP and CPP, for autonomous vehicles to navigate narrow passages and tight turns. Both algorithms consider not only the target position but also the target orientation of the vehicle for path tracking. OPP controller reduced the average maximum cross-track error by 75% compared to the PP baseline in simulation and 53.3% during field trials. The CPP controller used the contextual information of the vehicle state and the environment to obtain dynamic look-ahead distance. The average maximum cross track error for CPP was significantly

reduced by 79% in simulation and 36% for field trials. However, the enhanced conformity to the path by the two controllers was traded-off by an increased cost for maximum angular jerks. The maximum jerk for CPP was increased by 34% in simulation and 37% for field trials when compared to the orientation ignorant PP algorithm, although staying near comfort threshold. However, mean jerk reported for all controllers were insignificant. In conclusion, our proposed orientation and context aware path tracking controllers have provided promising results for improving the widely-known PP controller.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation, Prime Minister's Office, Singapore, under its CREATE programme, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG and the Singapore University of Technology and Design (SUTD) Startup Research Grant.

REFERENCES

[1] Omead Amidi and Chuck E Thorpe. Integrated mobile robot control. In *Mobile Robots V*, volume 1388, pages 504–523. International Society for Optics and Photonics, 1991.

[2] Arturo L Rankin, Carl D Crane III, David G Armstrong II, Allen D Nease, and H Edward Brown. Autonomous path-planning navigation system for site characterization. In *Navigation and Control Technologies for Unmanned Systems*, volume 2738, pages 176–186. International Society for Optics and Photonics, 1996.

[3] Jeff Wit, Carl D Crane III, and David Armstrong. Autonomous ground vehicle path tracking. *Journal of Robotic Systems*, 21(8):439–449, 2004.

[4] JA Marchant, T Hague, and ND Tillett. Row-following accuracy of an autonomous vision-guided agricultural vehicle. *Computers and electronics in agriculture*, 16(2):165–175, 1997.

[5] RS Ball. The theory of screws, cambridge, uk, 1900.

[6] Anthony Stentz, Cristian Dima, Carl Wellington, Herman Herman, and David Stager. A system for semi-autonomous tractor operations. *Autonomous Robots*, 13(1):87–104, 2002.

[7] Hiroki Ohta, Naoki Akai, Eijiro Takeuchi, Shinpei Kato, and Masato Edahiro. Pure pursuit revisited: Field testing of autonomous vehicles in urban areas. In *2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, pages 7–12. IEEE, 2016.

[8] Yunxiao Shan, Wei Yang, Cheng Chen, Jian Zhou, Ling Zheng, and Bijun Li. Cf-pursuit: a pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *International Journal of Advanced Robotic Systems*, 12(9):134, 2015.

[9] Yuanpeng Chen, Yunxiao Shan, Long Chen, Kai Huang, and Dongpu Cao. Optimization of pure pursuit controller based on pid controller and low-pass filter. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3294–3299. IEEE, 2018.

[10] Longsheng Chen, Ni Liu, Yunxiao Shan, and Long Chen. A robust look-ahead distance tuning strategy for the geometric path tracking controllers. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 262–267. IEEE, 2018.

[11] Ernő Horváth, Csaba Hajdu, and Péter Kőrös. Novel pure-pursuit trajectory following approaches and their practical applications. In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 000597–000602. IEEE, 2019.

[12] Ernő Horváth, Csaba Hajdu, and Péter Kőrös. Enhancement of pure-pursuit path-tracking algorithm with multi-goal selection. In *2019 1st IEEE International Conference on Gridding and Polytope Based Modeling and Control (GPMC)*, pages 13–18. IEEE, 2019.

[13] Derek J Chopp, Nate Spike, Jeremy Bos, and Darrell Robinette. Multi point pure pursuit. In *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2020*, volume 11415, page 1141505. International Society for Optics and Photonics, 2020.

[14] Noor Hafizah Amer, Hairi Zamzuri, Khisbullah Hudha, and Zulkifli Abdul Kadir. Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. *Journal of intelligent & robotic systems*, 86(2):225–254, 2017.

[15] Rafael Fierro and Frank L Lewis. Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. *Journal of robotic systems*, 14(3):149–163, 1997.

[16] Eric J Rossetter. *A potential field framework for active vehicle lanekeeping assistance*. PhD thesis, stanford university Palo Alto (CA, 2003.

[17] A Ollero and O Amidi. Predictive path tracking of mobile robots. application to the cmu navlab. In *Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments, ICAR*, volume 91, pages 1081–1086, 1991.

[18] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In *2015 european control conference (ECC)*, pages 3352–3357. IEEE, 2015.

[19] Gabriel M Hoffmann, Claire J Tomlin, Michael Montemerlo, and Sebastian Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *2007 American control conference*, pages 2296–2301. IEEE, 2007.

[20] Pan Zhao, Jiajia Chen, Yan Song, Xiang Tao, Tiejuan Xu, and Tao Mei. Design of a control system for an autonomous vehicle based on adaptive-pid. *International Journal of Advanced Robotic Systems*, 9(2):44, 2012.

[21] Alan SI Zinober. Deterministic control of uncertain systems. In *Proceedings. ICCON IEEE International Conference on Control and Applications*, pages 645–650. IEEE, 1989.

[22] Jinkun Liu and Xinhua Wang. *Advanced sliding mode control for mechanical systems*. Springer, 2012.

[23] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[24] Howard Dugoff, RD Ervin, and Leonard Segell. *Vehicle handling test procedures*. 1970.

[25] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. IEEE.

[26] Rahul Kumar Bhadani, Jonathan Sprinkle, and Matthew Bunting. The cat vehicle testbed: A simulator with hardware in the loop for autonomous vehicle applications. *arXiv preprint arXiv:1804.04347*, 2018.

[27] Curtis T Rueden, Johannes Schindelin, Mark C Hiner, Barry E DeZonia, Alison E Walter, Ellen T Arena, and Kevin W Eliceiri. Imagej2: Imagej for the next generation of scientific image data. *BMC bioinformatics*, 18(1):529, 2017.

[28] Jarrod M Snider et al. Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.

[29] Lars Svensson and Jenny Eriksson. Tuning for ride quality in autonomous vehicle: Application to linear quadratic path planning algorithm, 2015.

[30] ZJ Chong, B Qin, T Bandyopadhyay, T Wongpiromsarn, ES Rankin, MH Ang, E Frazzoli, D Rus, D Hsu, and KH Low. Autonomous personal vehicle for the first-and last-mile transportation services. In *2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)*, pages 253–260. IEEE, 2011.

[31] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[32] Dieter Fox. Kld-sampling: Adaptive particle filters and mobile robot localization. *Advances in Neural Information Processing Systems (NIPS)*, 14(1):26–32, 2001.