Letter of Response

Manuscript Number: DS-21-1096

Title: Low-speed vehicle path-tracking algorithm based on model predictive control using QPKWIK solver

Date: 30/05/2021

Dear Editor(s) of Journal of Dynamic Systems, Measurement and Control,

We are very glad to receive the reviewers' comments on our manuscript. The reviewers' comments are very helpful to improve our paper. We have revised the manuscript according to the reviewers' comments, and the changes are highlighted in red in the revised manuscript. Our responses to the reviewers' comments are shown below:

Response to Reviewer #1:

***Comment 1**: Comparison of multiple algorithms performance.*

***Response 1**:* Thanks for the reviewer's comment. A comparison of multiple algorithms tracking performance has been made in figure 1. The comparison between QPKWIK solver and QP solver is shown in Figure 5. The revision has been made in line 101 of the revised manuscript as:

"Taking the S curve as the reference path, the tracking performance of each algorithm is shown in Figure 1."

"To set the reference paths, the algorithm calls the $QUADPROG()$ or $QPKWIK()$ function and obtains the driving trajectory of the vehicle (see Fig. 2)."

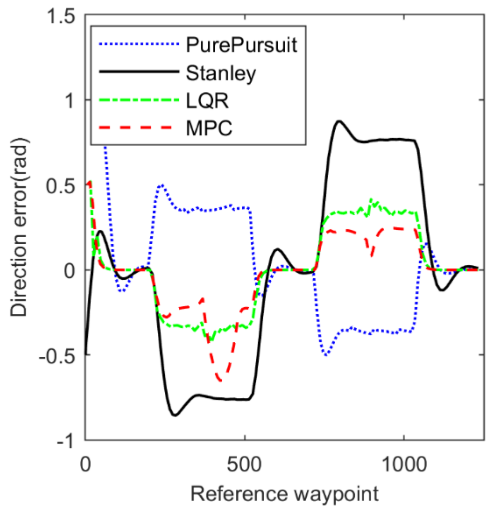

Fig. 1 Comparison of multiple algorithms tracking performance

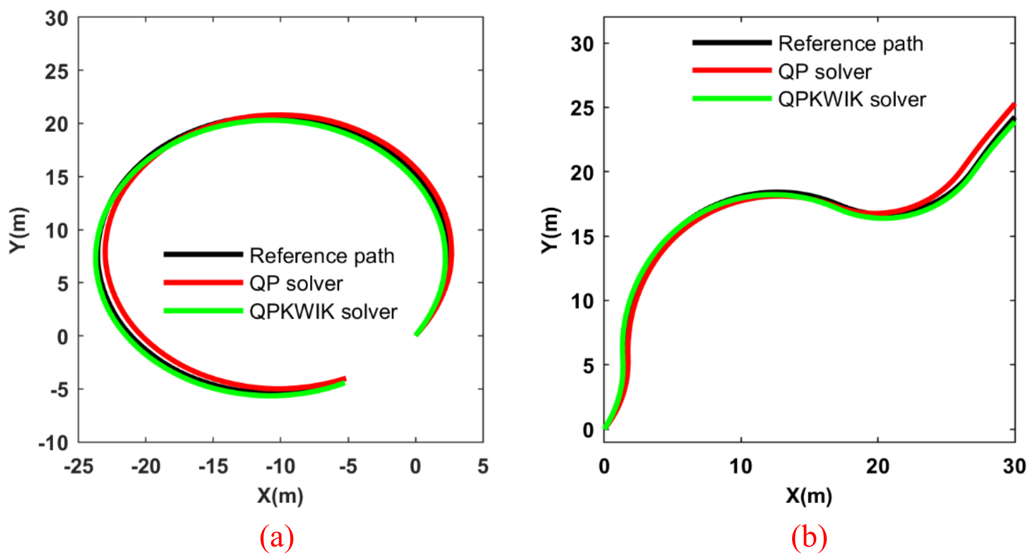

Fig. 2 Comparison of tracking results along the reference paths: (a) circular curve and (b) gradual curve

***Comment 2**: Driving scenario should include obstacles closer to the real world and not just a controlled environment like controlled parking lot.*

***Response 2**:* Thanks for the reviewer's comment. The low-speed path tracking algorithm in this paper is mainly used for automatic valet parking. The main application scenario is the parking lot, so other experimental scenarios such as urban roads and high-speed roads are not set up. In this work, the driving scene of the parking lot is simpler than that of urban roads, with no multiple lanes, less traffic, and fewer traffic participants. Therefore, four working conditions are set up in Section 5: straight driving, right-angle turning, rectangular steering, and obstacle avoidance. These four working conditions represent part of the situation in a general parking lot. Thanks again.

Comment 3*: Demonstrate how the algorithm is better than state of art? Not just limit themselves to implementation of considered algorithm.*

Response 3*:* Thanks for the reviewer’s comment. As mentioned in line 106, Model predictive control (MPC) is very suitable for vehicle path tracking. There are many researches on path tracking based on model predictive control, and many studies [5,31] prove that the tracking performance is better. Based on these studies, a QPKWIK solver applied to the MPC algorithm is proposed. The simulation results in section 4.2 show that the solution time of the QPKWIK solver is 25% less than that of the QP solver, and the tracking error is 41% less than that of the QP solver.

Reference:

[5] Wang, H., Liu, B., Ping, X., and An, Q., 2019, “Path Tracking Control for Autonomous Vehicles Based on an Improved MPC”, IEEE Access, **7**, pp. 161064-161073. DOI: 10.1109/access.2019.2944894

[31] Guo, H., Cao, D., Chen, H., Sun, Z., and Hu, Y., 2019, “Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification”, Mech. Syst. Signal Proc., 118, pp. 41-60. DOI: 10.1016/j.ymssp.2018.08.028

Comment 4*: Demonstrate if the change in operating conditions effects the tracking performance i.e. robustness to disturbance.*

Response 4*:* Thanks for the reviewer’s comment. There exists a couple of extra uncertainness in the dynamics of external disturbances applied to the vehicle. Although we use a linear tire model, the change in cornering stiffness caused by tire nonlinearity is also one of the external disturbances. However, MPC has its robustness to some extent [29,30], the controller can stabilize the system after operating conditions change, and the experimental results show that the robustness of the designed controller can achieve good tracking performance. Therefore, this article does not discuss robustness in depth. More in-depth research on robustness is one of our future research directions. Thanks again.

Reference:

[29] Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M., 2000, "Constrained model predictive control: Stability and optimality", Automatica, **36** (6), pp. 789-814. DOI: 10.1016/S0005-1098(99)00214-9

[30] Sun, C., Zhang, X., Zhou, Q., and Tian, Y., 2019 "A Model Predictive Controller With Switched Tracking Error for Autonomous Vehicle Path Tracking", IEEE Access, 7, pp. 53103-53114. DOI: 10.1109/access.2019.2912094

***Comment 5**: How does the algorithm perform when the path tracking is dynamically updating? This is realistic as available parking position can change in real-time. Also, parking lot is a good example for slow-speed application.*

***Response 5**:* Thanks for the reviewer's comment. In Section 5.4, the dynamic path update scenario was tested. The vehicle detects an obstacle ahead and re-plans a collision-free path through the path planning module. After generating the new path, the path tracking controller designed in this paper can continue to track the new reference path and control the error within a small range. This process is solved in real-time using the QPKWIK solver in the path tracking algorithm, which can track the reference path stably. If the parking position changes in real-time, similar to obstacle avoidance, the map module and the path planning module need to work together to generate a new path. The path tracking algorithm in this paper can also track the new reference path. The real-time update is also one of the advantages of the model predictive control algorithm. Since a series of prediction sequences and control sequences are generated at each moment, the proposed algorithm can adapt to such changes for the dynamic update path. Thanks again.

***Comment 6**: Can the authors describe the effect of different horizon length on the* *results?*

Response 6*:* Thanks for the reviewer's comment. Since the horizon length affects computational burden, We set $Np = 60$ and $Nc = 30$. As for the effect of different horizon lengths on tracking performance and computational burden, no quantitative research is done in this article, but qualitative analysis is done. That is, the larger the horizon length, the higher the computational burden. Quantitative analysis of the horizon length to tracking performance will be one of our future works. Revision has been made in line 342 of the revised manuscript as:

"Since the length of the horizon affects the computational burden, the length of the horizon is too long, resulting in high computational complexity[41]. In this article, we set $Np = 60$ and $Nc = 30$. Under this condition, the tracking error of the vehicle is the smallest and the computational burden is not high."

Reference:

[41] Cannon, M., Liao, W., and Kouvaritakis, B., 2008 "Efficient MPC optimization using Pontryagin's minimum principle", Int. J. Robust Nonlinear Control, **18** (8), pp. 831-844. DOI: 10.1002/rnc.1247

Response to Reviewer #2:

***Comment 1**: QPKWIK long form should be specified and referenced for readers to understand what this entails.*

Response 1: Thanks for the reviewer's comment. The longform of QPKWIK is quadratic programming knows what it knows. Revision has been made in line 135 of the revised manuscript as:

"To further reduce computational burden and tracking error, the present paper proposes a vehicle path-tracking algorithm based on model predictive control using the quadratic programming knows what it knows (QPKWIK) solver [34]."

Reference:

[34] Goldfarb, D., and Idnani, A., 1983, "A numerically stable dual method for solving strictly convex quadratic programs", Math. Program., **27** (1), pp. 1-33. DOI: 10.1007/BF02591962

***Comment 2**: Inserting hyperlinks to references, figures, tables, equations will be helpful so that clicking will open the specific page.*

***Response 2**:* Thanks for the reviewer's comment. The hyperlinks to references, figures, tables, equations have been inserted into this article. Thanks again for your kind suggestion.

***Comment 3**: In Fig 4, it will be helpful if input to the objective function and output of the predictive model is described.*

***Response 3**:* Thanks for the reviewer's comment. The input to the objective function is "Predictive path and Reference path," and the output of the predictive model is "Predictive path." These have been added to Fig. 4. The following figure has been updated in line 790 of the revised manuscript.

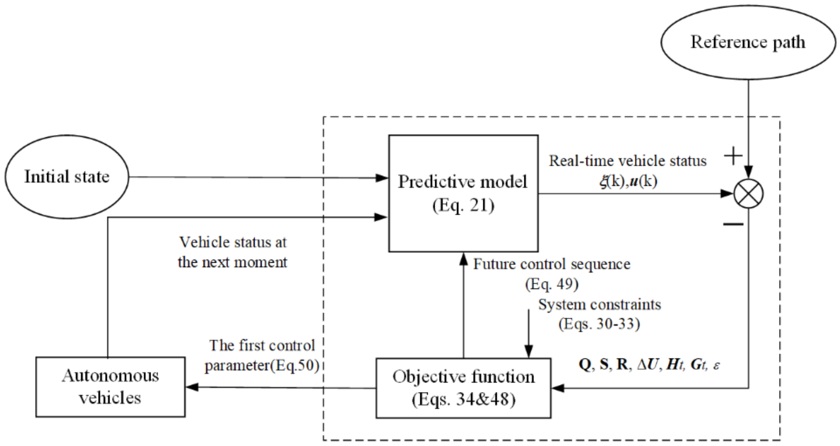

Fig. 4 Diagram of the path-tracking controller

Comment 4*: In Model Predictive Controller Design, specifically, section 3.1, Predictive Model, all the equations mentioned should be elaborated as to how they co-relate to the actual vehicle operation. Shedding some light in this aspect will be useful instead of just mentioning bunch of mathematical equations as it is important to translate them into what effect model predictive controls will have on the vehicle or vehicle functioning.*

Response 4*:* Thanks for the reviewer's comment. In section 3.1 Predictive Model, some unnecessary equations (Eq. 17, 25) have been added to the appendix. This simplifies the equations in section 3.1 and expresses the established vehicle motion state equation and output variables more clearly. Eqs. (15)-(22) are used to establish the new state-space equations and output control variables of the vehicle. Thus, the state and output variables in the predicted time domain can be obtained through the current state and control variables of the system. The statement has been made in line 290 and the appendix. B (Already mentioned in the response to reviewer #3 minor comment 6) has been added in line 577 of the revised manuscript.

"From Eq. (21), one observes that in the system prediction time domain, the system-state and system-control output can be calculated through $\boldsymbol{\psi}(k|t)$ and $\boldsymbol{\Delta U}(k)$, thus realizing a 'system future-state prediction' function. Combining Eq. (23) and Eq. (15), we can predict the motion and system states of the vehicle in the time domain. The vehicle control is optimized by optimizing the solution of $\boldsymbol{\Delta U}(k)$ in the control time domain.”

Comment 5: *“As the control variable of the path-following controller is unknown, we must set a regular objective function”; line 287 - Which control variable is being referred to here? This should be clarified and explicitly mentioned or pointed to.*

Response 5: Thanks for the reviewer’s comment. The control variables include the heading angle and the speed of the vehicle. Revision has been made in line 329 of the revised manuscript as:

“As the control variable of the path-following controller is unknown, we must set a regular objective function. The control variables include the heading angle, and the speed of the vehicle.

Comment 6: *“The objective function must ensure fast and stable tracking of the vehicle along the reference path.”; line 289 – How is it ensured that objective function has met this: “ensure fast and stable tracking”? Also, has the designed objective function achieved this? If not, what are the consequences?*

Response 6: Thanks for the reviewer’s comment. It can be seen from the construction of the objective function that the process of using the QPKWIK solver to minimize the objective function to obtain the result is a process to ensure fast and stable tracking. The $\sum_{i=1}^{Np} \|\eta(t+i|t) - \eta_{ref}(t+i|t)\|_Q^2$ in the objective function represents the deviation between the reference output of the system and the actual output. Whether the system can track quickly depends on the time to solve the objective function and the response time of the system. From the experimental results, the system can achieve the goal of fast and stable tracking, as in section 5.2, in the early stage of steering, the system response lag will cause position deviation, and then the controller will continue to work so that the vehicle can achieve good tracking performance.

Comment 7: *While introducing section 4.2, the motivation of the new quadratic* *programming solver, QPKIWK should be explained in detail. Things like what current solutions lacks, what the new proposed solution offers and what impact does it have on the vehicle path tracking.*

***Response** 7:* Thanks for the reviewer's comment. Most of the solutions in MPC are transformed into QP problems. It causes a high computational burden. The proposed QPKWIK solver can reduce the computational burden and tracking error. A careful revision has been made in line 441 and 115 of the revised manuscript as:

"Reducing the computational burden and reducing tracking errors are the main ways to improve MPC performance. Many studies transform solutions in MPC into QP problems and use QP solvers to solve them [44]. Since MPC needs to perform real-time rolling optimization, the QP solver will also cause a high computational burden, which is of great significance to the optimization of the solver. To solve this sub-problem effectively and reduce the computational burden, we proposed a new quadratic programming solver, QPKWIK, based on the successive quadratic programming problem [45] and dual algorithm [34]."

"The MPC was generally implemented using a nonlinear vehicle model and linear vehicle model. Thus, the solution of MPC can be transformed into a nonlinear programming problem. This problem could be solved by the Interior-Point method [32] or Sequential Quadratic Programming (SQP) method [33]. Real-time implementation of nonlinear MPC is limited owing to the computational complexity of the nonlinear programming problem. Dealing with this problem, linearization techniques can be used to replace the nonlinear programming problem, and finally converted it into quadratic programming (QP) problem. Beginning with the unconstrained minimum of the objective function, the authors of [34] proposed an effective and numerically stable dual modus for positive definite quadratic programming. The authors of [35] proposed a simplified successive quadratic programming method that satisfactorily solves large-scale processing optimization problems with multiple variables, multiple constraints, and few degrees of freedom at faster iteration speeds than other methods. For the issue of online calculation burden and complex tuning process, the online parameter selection method based on a look-up table is used to help the vehicle track the reference path under the constraints of stability and computing power. It realizes the online tuning of the controller and obtains excellent tracking performance [28]. In fact, MPC is greatly affected by computational complexity. Therefore, reducing the computational burden has become a key factor in the realization of the MPC method.

Regardless of whether linear models or nonlinear models are used, most solutions in the MPC method are transformed into QP problems for further optimization. To further reduce computational burden and tracking error, the present paper proposes a vehicle path-tracking algorithm based on model predictive control using the quadratic programming knows what it knows (QPKWIK) solver [34]. The contributions of this work are presented as follows:

- A kinematic model was established to satisfy the driving of the vehicle, and the vehicle motion state was described by a discrete state-space equation constructed through discretization processing, which obtained the input parameters of the path-tracking controller.
- Based on the vehicle motion equation, the vehicle constraints are added to the controller, the linear model is used to establish the model predictive controller, and the tracking error and heading error are used as the main influencing factors of the objective function
- The model predictive controller solver is further optimized, and the QPKWIK solver is used to achieve a smaller computational cost and smaller tracking error. Experimental results show that the controller has better tracking performance."

Reference:

[28] Wang, Z., Bai, Y., Wang, J., and Wang, X., 2019, "Vehicle Path-Tracking Linear-Time-Varying Model Predictive Control Controller Parameter Selection Considering Central Process Unit Computational Load", ASME J. Dyn. Sys., Meas., Control., **141** (5). DOI: 10.1115/1.4042196

[32] Vasantharajan, S., Viswanathan, J., and Biegler, L.T., 1990, "Reduced successive quadratic programming implementation for large-scale optimization problems with smaller degrees of freedom", Comput. Chem. Eng., **14** (8), pp. 907-915. DOI: 10.1016/0098-1354(90)87045-Q

[33] Falcone, P., Eric Tseng, H., Borrelli, F., Asgari, J., and Hrovat, D.,2008 "MPC-based yaw and lateral stabilisation via active front steering and braking", Veh. Syst. Dyn., **46** (sup1), pp. 611-628. DOI: 10.1080/00423110802018297

[34] Goldfarb, D., and Idnani, A., 1983, "A numerically stable dual method for solving strictly convex quadratic programs", Math. Program., **27** (1), pp. 1-33. DOI: 10.1007/BF02591962

[35] Schmid, C., and Biegler, L.T., 1994, "Reduced Hessian successive quadratic programming for realtime optimization", Advanced Control of Chemical Processes, pp. 173-178. DOI: 10.1016/B978-0-08-042229-9.50032-X

[44] Rokonuzzaman, M., Mohajer, N., Nahavandi, S., and Mohamed, S., 2021 "Review and performance evaluation of path tracking controllers of autonomous vehicles", IET Intell. Transp. Syst., **15** (5), pp. 646-670. DOI: 10.1049/itr2.12051

[45] Schmid, C., and Biegler, L.T., 1994, "Quadratic programming methods for reduced hessian SQP", Comput. Chem. Eng., **18** (9), pp. 817-832. DOI: 10.1016/0098-1354(94)E0001-4

***Comment 8**: "The experimental results of tracking along an L right-angled road are shown in Figure 10.";line 459 - I guess Fig 8 is being referenced over here and not 10, right? Please correct it.*

***Response 8**:* Thanks for the reviewer's kind suggestions and pointing this out. The Fig 10 has been changed to Fig. 8 in the line 514 of the revised manuscript:

"The experimental results of tracking along an L right-angled road are shown in Fig. 8."

Response to Reviewer #3:

Major comments

Comment 1: *It is a little bit confusing when you talk about the solver. It is vague weather authors developed solver or only applied solver in this application.*

Response 1: Thanks for the reviewer's comment. By improving the existing QPKWIK solver, the H matrix is combined with the dual algorithm for Cholesky decomposition, forming the new QPKWIK solver mentioned in this article. And then apply it to the MPC algorithm. A careful revision has been made in line 445 of the revised manuscript. Thanks again.

"To solve this sub-problem effectively and reduce the computational burden, we proposed a new quadratic programming solver, QPKWIK, based on the successive quadratic programming problem [45] and dual algorithm [34]."

Reference:

[34] Goldfarb, D., and Idnani, A., 1983, "A numerically stable dual method for solving strictly convex quadratic programs", Math. Program., **27** (1), pp. 1-33. DOI: 10.1007/BF02591962

[45] Schmid, C., and Biegler, L.T., 1994, "Quadratic programming methods for reduced hessian SQP", Comput. Chem. Eng., **18** (9), pp. 817-832. DOI: 10.1016/0098-1354(94)E0001-4

Comment 2: *Motivation of this study is not clear at all. Why do we need study low speed tracking control? Why MPC and developed solver is the best option? How proposed method differ from existing works?*

Response 2: Thanks for the reviewer's comment. The motivation has been revised in the introduction. The literature review has been updated to provide evidence for the motivation. In the automatic valet parking scenario, the vehicle is moving at a low speed because it is driving in the parking lot at this time. Compared with other methods, MPC is suitable for dealing with vehicle stability constraints as well as changing vehicle and tire dynamics due to the ability to systematically including system constraints and future predictions. Most of the solutions in the MPC method are transformed into QP problems, resulting in a high computational burden and high tracking error. To solve this problem, a QPKWIK solver combined with a dual algorithm to further optimize the solver has been proposed. The introduction has been updated in the revised manuscript as：

"To track the predetermined path accurately and stably, the tracking error has become a key indicator to evaluate the tracking effect. Several kinds of tracking errors have been used in the path tracking algorithm. In simple geometric tracking controllers, such as pure pursuit [15], the steering angle is directly determined from the lateral deviation through the geometrical relationship between the vehicle and the desired path. But the main shortcoming of Pure Pursuit is in the selection of look-ahead distance. Another shortcoming of this algorithm is that it may not be able to negotiate discontinuous paths [16]. It is a common problem in geometric controllers. Improvement of this method has been proposed previously. An analytical method to tune the look-ahead distance base on a linearized model was proposed for path tracking control by Campbell [17]. Another geometric controller is "Stanley". It is an autonomous vehicle developed by Stanford University that won the second DARPA Grand Challenge in 2005. This method has been referenced and implemented in many studies. A nonlinear Stanley controller has been implemented on Hardware-in-the-loop testing and compares it against a linear method of double-loop control with feed-forward load disturbance compensation [18]. The results show that the Stanley controller was more superior with the better transient response and lower steady-state errors in curves. Moreover, unlike pure pursuit control, a look-ahead distance is not required for designing the controller. Linear quadratic regulator (LQR) is one of the most popular optimal controllers where the controller gain was determined using a linear quadratic optimization approach. Fan et al. [19] adopted LQR optimal control to achieve the closed-loop control of vehicle linear system to guarantee its stability and fast convergence property in Automatic Parking System. However, due to the absence of path feedback, the LQR controller performed badly, it undesirably provides steady-state errors when negotiating curvature paths [20].

Sliding mode control also has been successfully employed in many types of research on path tracking control for autonomous vehicles to ensure the robustness of the control system and to handle the nonlinear vehicle dynamics [21,22]. Wang et al. [22] proposed saturation function in the control law to solve the Chattering issues, but the chattering phenomenon still exists. The author in [23] proposed a robust H_∞ output-feedback control strategy combined with mixed genetic algorithms (GA) /linear matrix inequality (LMI) was used to obtain proper static output-feedback gains. Besides, machine learning methods, with their capability to handle nonlinear systems, provide a new approach to deal with the nonlinear vehicle dynamic system problem. A control architecture combined traditional methods and deep reinforcement learning methods was proposed to correct control errors and reduce the time and effort of testing and tuning [24]. Shan et al. [25] proposed an adaptive Reinforcement Learning-based weight-adjustment model to adjust the weight in PP_PID to trade-off between the effects of PP and PID. This method can better deal with tracking errors and adapt to various tracking scenarios. Nevertheless, training a machine learning model adapted to real vehicles is still hard and needs high computational costs. Taking the S curve as the reference path, the tracking performance of each algorithm is shown in Figure 1.

From above, a variety of control algorithms have been used in path tracking design, however, there are still many problems in balancing the robustness and tracking error. It would be of special interest to solve these problems for the path tracking control of autonomous vehicles. Model predictive control (MPC) is very suitable for dealing with vehicle stability constraints as well as changing vehicle and tire dynamics due to the ability to systematically including system constraints and future predictions [26-28]. Also, the inherent robustness of MPC guarantees the system robustness to some extent [29,30]. Constrained optimization problems can be solved by model predictive control (MPC) in an optimal controller, which predicts the future state of the system and can handle state and control variables [31]. In practice, vehicle path-tracking by a model predictive controller is mathematically formulated in the design stage, but the system output is solved in real-time on a computing platform.

The MPC was generally implemented using a nonlinear vehicle model and linear vehicle model. Thus, the solution of MPC can be transformed into a nonlinear programming problem. This problem could be solved by the Interior-Point method [32] or Sequential Quadratic Programming (SQP) method [33]. Real-time implementation of nonlinear MPC is limited owing to the computational complexity of the nonlinear programming problem. Dealing with this problem, linearization techniques can be used to replace the nonlinear programming problem, and finally converted it into quadratic programming (QP) problem. Beginning with the unconstrained minimum of the objective function, the authors of [34] proposed an effective and numerically stable dual modus for positive definite quadratic programming. The authors of [35] proposed a simplified successive quadratic programming method that satisfactorily solves large-scale processing optimization problems with multiple variables, multiple constraints, and few degrees of freedom at faster iteration speeds than other methods. For the issue of online calculation burden and complex tuning process, the online parameter selection method based on a look-up table is used to help the vehicle track the reference path under the constraints of stability and computing power. It realizes the online tuning of the controller and obtains excellent tracking performance [28]. In fact, MPC is greatly affected by computational complexity. Therefore, reducing the computational burden has become a key factor in the realization of the MPC method."

Reference:

[15] Elbanhawi, M., Simic, M., and Jazar, R., 2016, "Receding horizon lateral vehicle control for pure pursuit path tracking", J. Vib. Control., **24** (3), pp. 619-642. DOI: 10.1177/1077546316646906

[16] Amer, N.H., Zamzuri, H., Hudha, K., and Kadir, Z.A., 2016, "Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges", J. Intell. Robot. Syst., **86** (2), pp. 225-254. DOI: 10.1007/s10846-016-0442-0

[17] Campbell, S.F., 2007, “Steering control of an autonomous ground vehicle with application to the DARPA urban challenge”, Massachusetts Institute of Technology, Ph.D. thesis. DOI: 1721.1/42301

[18] Törő, O., Bécsi, T., and Aradi, S., 2016, “Design of Lane Keeping Algorithm of Autonomous Vehicle”, Period. Polytech. Transp. Eng., **44** (1), pp. 60-68. DOI: 10.3311/PPtr.8177

[19] Fan, Z., and Chen, H., 2016, “Study on Path Following Control Method for Automatic Parking System Based on LQR”, SAE Int. J. Passenger Cars Electron. Electr. Syst., **10** (1), pp. 41-49. DOI: 10.4271/2016-01-1881

[20] Cordeiro, R.A., Azinheira, J.R., de Paiva, E.C., and Bueno, S.S., 2013, “Dynamic modeling and bio-inspired LQR approach for off-road robotic vehicle path tracking”, 16th Int. Conf. Adv.Robot. (ICAR), pp. 1-6. DOI: 10.1109/ICAR.2013.6766549

[21] Solea, R., and Nunes, U., 2006, “Trajectory planning and sliding-mode control based trajectory-tracking for cybercars”, Integr. Comput.-Aided Eng., **14** (1), pp. 33-47. DOI: 10.3233/ICA-2007-14104

[22] Wang, J., Steiber, J., and Surampudi, B., 2009, “Autonomous ground vehicle control system for high-speed and safe operation”, Int. J. Veh. Auton. Syst., **7** (1-2), pp. 18-35. DOI: 10.1504/IJVAS.2009.027965

[23] Hu, C., Jing, H., Wang, R., Yan, F., and Chadli, M., 2016, “Robust H∞ output-feedback control for path following of autonomous ground vehicles”, Mech. Syst. Signal Proc., **70-71**, pp. 414-427. DOI: 10.1016/j.ymssp.2015.09.017

[24] Chen, I.M., and Chan, C.-Y., 2020. “Deep reinforcement learning based path tracking controller for autonomous vehicle”, Proc. Inst. Mech. Eng. Part D-J. Automob. Eng., **235** (2-3), pp. 541-551. DOI: 10.1177/0954407020954591

[25] Shan, Y., Zheng, B., Chen, L., Chen, L., and Chen, D., 2020, “A Reinforcement Learning-Based Adaptive Path Tracking Approach for Autonomous Driving”, IEEE Trans. Veh. Technol., **69** (10), pp. 10581-10595. DOI: 10.1109/tvt.2020.3014628

[26] Brown, M., Funke, J., Erlien, S., and Gerdes, J.C., 2017, "Safe driving envelopes for path tracking in autonomous vehicles", Control Eng. Practice, **61**, pp. 307-316. DOI: 10.1016/j.conengprac.2016.04.013

[27] Jalali, M., Khajepour, A., Chen, S.-k., and Litkouhi, B., 2017, "Handling Delays in Yaw Rate Control of Electric Vehicles Using Model Predictive Control With Experimental Verification", ASME J. Dyn. Sys., Meas., Control., **139** (12). DOI: 10.1115/1.4037166

[28] Wang, Z., Bai, Y., Wang, J., and Wang, X., 2019, "Vehicle Path-Tracking Linear-Time-Varying Model Predictive Control Controller Parameter Selection Considering Central Process Unit Computational Load", ASME J. Dyn. Sys., Meas., Control., **141** (5). DOI: 10.1115/1.4042196

[29] Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M., 2000, "Constrained model predictive control: Stability and optimality", Automatica, **36** (6), pp. 789-814. DOI: 10.1016/S0005-1098(99)00214-9

[30] Sun, C., Zhang, X., Zhou, Q., and Tian, Y., 2019 "A Model Predictive Controller With Switched Tracking Error for Autonomous Vehicle Path Tracking", IEEE Access, **7**, pp. 53103-53114. DOI: 10.1109/access.2019.2912094

[31] Guo, H., Cao, D., Chen, H., Sun, Z., and Hu, Y., 2019, "Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification", Mech. Syst. Signal Proc., **118**, pp. 41-60. DOI: 10.1016/j.ymssp.2018.08.028

[32] Vasantharajan, S., Viswanathan, J., and Biegler, L.T., 1990, "Reduced successive quadratic programming implementation for large-scale optimization problems with smaller degrees of freedom", Comput. Chem. Eng., **14** (8), pp. 907-915. DOI: 10.1016/0098-1354(90)87045-Q

[33] Falcone, P., Eric Tseng, H., Borrelli, F., Asgari, J., and Hrovat, D.,2008 "MPC-based yaw and lateral stabilisation via active front steering and braking", Veh. Syst. Dyn., **46** (sup1), pp. 611-628. DOI: 10.1080/00423110802018297

[34] Goldfarb, D., and Idnani, A., 1983, "A numerically stable dual method for solving strictly convex quadratic programs", Math. Program., **27** (1), pp. 1-33. DOI: 10.1007/BF02591962

[35] Schmid, C., and Biegler, L.T., 1994, "Reduced Hessian successive quadratic programming for realtime optimization", Advanced Control of Chemical Processes, pp. 173-178. DOI: 10.1016/B978-0-08-042229-9.50032-X

Comment 3: *Discussion regarding tire model for bicycle model is missing. I am assuming authors using linear model; however, adding extensive discussion regarding tire model is highly recommended. You can see "tire model" section in this study: Lateral control of an autonomous vehicle using integrated backstepping and sliding mode controller.*

Response 3: Thanks for the reviewer's comment. The discussion regarding tire model has been added in line 184 of the revised manuscript as:

"The two-DOF bicycle model has been used for designing the model predictive controller. The bicycle model has been built on a linear tyre model [38]. The controller has been applied to the vehicle by using Matlab/Simulink. The tyer model is integrated into Matlab/Simulink as one of the modules. Also, Pacejka magic formula tyre model [39] and brush tire model [40] can be used in the simulation."

Reference:

[38] Norouzi, A., Masoumi, M., Barari, A., and Farrokhpour Sani, S., 2018, "Lateral control of an autonomous vehicle using integrated backstepping and sliding mode controller", Proc. Inst. Mech Eng Pt K-J Multi-Body Dyn., **233** (1), pp. 141-151. DOI: 10.1177/1464419318797051

[39] Pacejka, H.B., and Bakker, E., 1992, "The Magic Formula Tyre Model", Veh. Syst. Dyn., **21**, (sup001), pp. 1-18. DOI: 10.1080/00423119208969994

[40] Kapania, N.R., and Gerdes, J.C., 2015, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling", Veh. Syst. Dyn., **53**, (12), pp. 1687-1704. DOI: 10.1080/00423114.2015.1055279

Comment 4: *It is unclear that how Eq. 31, 32, and 33 has been drive. Does it based on any standard or reference or you drive it based on your system specification.*

Response 4: Thanks for the reviewer's comment. The front-wheel angle increment in Eq. (25) is obtained through vehicle experiments. Since the structure of the steering mechanism of each vehicle is not the same, the data is measured by experiments. Because it is driving in a low-speed environment, the speed of the car cannot change too much, and at the same time, it needs to meet the requirements of human comfort. After vehicle debugging, the acceleration range of this experiment is determined, so the acceleration range in Eq. (26) is also obtained through experiments Value, in this range, the vehicle tracking performance is best, and the human body feels the most comfortable. Eq. (27) is to ensure that the car does not collide with the edge of the road, so a minimum distance between the corner of the vehicle and the edge of the road is set. Considering obstacle avoidance and sensor accuracy, the value is 30 cm after the experiment. This is the minimum value that the experimental vehicle in this article can guarantee to avoid the collision.

In summary, In Eq. (25), (26), and (27), the data are all measured by experiment. A careful revision has been made in line 308 of the revised manuscript as follows:

"To ensure the stability and safety of the driving vehicle, excessive steering must be prevented by constraining the increment of the front-wheel angle. Due to the differences in the structure of the steering system of different vehicles, the front wheel angle increment needs to be measured through experiments. Through experiments, the front-wheel angle increment on a flat road is limited to:

$$-0.5^{\circ} \leq \Delta\delta_f \leq 0.5^{\circ} \tag{25}$$

Since the vehicle is running in a low-speed environment, in order to prevent the tracking performance from being affected by the instantaneous increase or decrease of the vehicle speed, and at the same time, considering the comfort of passengers, the acceleration range is determined. Through experiments, the vehicle tracking performance and passenger comfort are the best in the following acceleration ranges.

$$-0.06m/s^2 \le a \le 0.06m/s^2 \quad (26)$$

To ensure that the vehicle does not collide with the edge of the road during driving, the minimum distance between the corner point of the vehicle and the edge of the road is set. The vehicle is subject to the following constraints:

$$\|(X_{i,k+1}, Y_{i,k+1})\| \le \|(X_{b,k+1}, Y_{b,k+1})\| - D_s \quad (27)$$

where $(X_{i,k+1}, Y_{i,k+1})$ is the coordinate of the corner i of the vehicle at time $k+1$, $(X_{b,k+1}, Y_{b,k+1})$ is the road boundary of the vehicle's position at time $k+1$, and D_s is the anti-collision safety distances. Considering obstacle avoidance and sensor accuracy, D_s is set at 30cm after experiment."

Comment 5: *I wondered how author define reference path as it is important part of autonompoise vehicle. I recommend exclusive literature review in that part. e.g. Path planning and re-planning of lane change manoeuvres in dynamic traffic environments.*

Response 5: Thanks for the reviewer's comment. The reference path was provided by path planning module. The literature review has been added in line 45 of the revised manuscript as:

"The predetermined path can be obtained by the path planning module. The path planning module is responsible for computing a safe, comfortable, and dynamically feasible path from the vehicle's current configuration to the goal configuration provided by the decision-making module [6]. Path planning in autonomous vehicles has been studied for years. Many authors divide the problem into global and local planning. Dijkstra algorithm is a graph searching algorithm that finds the single-source shortest path in the graph [7]. It was widely used in global path planning. A* algorithm is an extension of the Dijkstra algorithm. It enables a fast node search due to the implementation of heuristics [8]. The cost function in the A* algorithm defines the weights of the nodes. It is suitable for searching spaces mostly known a prior by the vehicle, however, it needs high computational cost. Rapidly-Exploring Random Tree (RRT) allows fast planning in semi-structured spaces, often used in local path planning [9]. Besides, the clothoid curves [10,11], polynomial curves [12], Bezier curves [13], and splines curves [14] were also applied in local path planning. The author in [12] proposed a novel path for lane change maneuvers based on a quantic function. The simulation proves that the lane change path generated by the polynomial curve is smooth and safe. The path planning module provides the desired path of the vehicle."

Reference:

[6] Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E., 2016, "A survey of motion planning and control techniques for self-driving urban vehicles", IEEE Trans. Intell. Vehicles, **1** (1), pp. 33-55. DOI: 10.1109/TIV.2016.2578706

[7] Zheng, Y., and Özgüner;, Ü., 2007, "A Composite Model for Vehicle Formation and Path Selection on a Cellular Structured Map", ASME J. Dyn. Sys., Meas., Control., **129** (5), pp. 644-653. DOI: 10.1115/1.2764506

[8] Likhachev, M., and Ferguson, D., 2009, "Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles", Int. J. Robot. Res., **28** (8), pp. 933-945. DOI: 10.1177/0278364909340445

[9] Karaman, S., and Frazzoli, E.,2011, "Sampling-based algorithms for optimal motion planning", Int. J. Robot. Res., **30** (7), pp. 846-894. DOI: 10.1177/0278364911406761

[10] Brezak, M., and Petrovi, I., 2014, "Real-time Approximation of Clothoids With Bounded Error for Path Planning Applications", IEEE Trans. Robot., **30** (2), pp. 507-515. DOI: 10.1109/TRO.2013.2283928

[11] Funke, J., and Christian Gerdes, J., 2015, "Simple Clothoid Lane Change Trajectories for Automated Vehicles Incorporating Friction Constraints", ASME J. Dyn. Sys., Meas., Control., **138** (2). DOI: 10.1115/1.4032033

[12] Norouzi, A., Kazemi, R., and Abbassi, O.R., 2019, "Path planning and re-planning of lane change manoeuvres in dynamic traffic environments", Int. J. Veh. Auton. Syst., **14** (3), pp. 239-264. DOI: 10.1504/IJVAS.2019.099831

[13] Chen, Y., Cai, Y., Zheng, J., and Thalmann, D., 2017, "Accurate and Efficient Approximation of Clothoids Using Bezier Curves for Path Planning", IEEE Trans. Robot., **33** (5), pp. 1242-1247. DOI: 10.1109/TRO.2017.2699670

[14] Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M., and Soderkvist, I., 2010, "Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles", IEEE Trans. Autom. Sci. Eng., **7** (1), pp. 167-172. DOI: 10.1109/TASE.2009.2015886

Comment 6: *What is reason of high error in turing (Fig. 8); however, in Fig. 9 which has couple of turning, the tracking performance is better?*

Response 6: Thanks for the reviewer's comment. In Fig. 8, the maximum position and heading deviations were 91.5 cm and 0.40 rad, respectively. The deviations were largest at NOC = [500, 550], the early stage of turning the curve. Due to the delay, the steering mechanism of the vehicle has not yet performed the steering action at this time, so the error at this stage is high. As the controller works, the heading is adjusted quickly and the error is reduced. In Fig. 9, the maximum deviations in the position and heading directions were 88.6 cm and 0.58 rad, respectively. From these two indicators, the tracking effect of Figure 8 and Figure 9 is not much different, which is also one of the goals of this algorithm, which can maintain good tracking performance under multiple working conditions. Explanations have been added to line 517 of the revised manuscript as:

"The deviations were largest in at NOC = [500, 550], the early stage of turning the curve. At this time, the steering mechanism of the vehicle has not yet performed the steering action, and there is a delay, so the error at this stage is high. The positional deviation reduced as the heading was quickly adjusted after the controller working."

Comment 7: *It is not clear weather all the results is experimental results or not? Table 1 caption is "Comparative analysis of the tracking simulation results", so it seems all the results is simulation. If so, what is the reason of discuss about experimental setup?*

Response 7: Thanks for the reviewer's comment. Table 1 in Section 4 is the simulation results. Section 5 is the experimental results. In Section 4, an arbitrary curve is generated and these two solvers are used for path tracking respectively. The simulation results in Table 1 are the comparison of the tracking effect between the QPKWIK solver and the general QP solver. The simulation results show that the QPKWIK solver has better performance. Therefore, a real-vehicle experiment was carried out in Section 5, and the performance of the solver in the actual situation was further verified through the experiment. Thanks again.

Minor comments

Comment 1: *The abstract needs to be revised as it is too general. To enhance the article, the problem formulation and motivation, as well as the novelty of the methodology, should be already stated in the abstract. Also, superior results should already be pointed out more clearly.*

Response 1: Thanks for the reviewer's comment. A careful revision has been made in Abstract.

"Automated valet parking is a part of autonomous vehicles. Path tracking is a vital capability of autonomous vehicles. In the scenario of automatic valet parking, the existing control algorithm will produce a high tracking error and a high computational burden. This paper proposes a path-tracking algorithm based on model predictive control to adapt to low-speed driving. By using the model predictive control method and vehicle kinematics model, a path tracking controller is designed. Combining the dual algorithm to further optimize the solver, a new QPKWIK solver is proposed. The simulation results show that the solution time of the QPKWIK solver is 25% less than that of the QP solver, and the tracking error is reduced by up to 41% compared with the QP solver. In the parking lot, the tracking performance is tested under four common scenarios, and the experimental results show that this controller has better tracking performance."

Comment 2: *Literature review needs to be updated and covered recent developments in the field. Using Machine learning methods is a new trend in this field and acknowledging these methods in literature is crucial.*

Response 2: Thanks for the reviewer's comment. Literature review has been updated and covered the machine learning methods. A careful statement has been made in line

92 of the revised manuscript as:

"Besides, machine learning methods, with their capability to handle nonlinear systems, provide a new approach to deal with the nonlinear vehicle dynamic system problem. A control architecture combined traditional methods and deep reinforcement learning methods was proposed to correct control errors and reduce the time and effort of testing and tuning [24]. Shan et al. [25] proposed an adaptive Reinforcement Learning-based weight-adjustment model to adjust the weight in PP_PID to trade-off between the effects of PP and PID. This method can better deal with tracking errors and adapt to various tracking scenarios. Nevertheless, training a machine learning model adapted to real vehicles is still hard and needs high computational costs."

Reference:

[24] Chen, I.M., and Chan, C.-Y., 2020. "Deep reinforcement learning based path tracking controller for autonomous vehicle", Proc. Inst. Mech. Eng. Part D-J. Automob. Eng., **235** (2-3), pp. 541-551. DOI: 10.1177/0954407020954591

[25] Shan, Y., Zheng, B., Chen, L., Chen, L., and Chen, D., 2020, "A Reinforcement Learning-Based Adaptive Path Tracking Approach for Autonomous Driving", IEEE Trans. Veh. Technol., **69** (10), pp. 10581-10595. DOI: 10.1109/tvt.2020.3014628

Comment 3: *Fig. 1. need to be redesigned professionally. I would like to introduce Vehicle dynamics and control" by Rajesh Rajaman, where you can find a better graphical representation of the bicycle model.*

Response 3: Thanks for the reviewer's comment. Based on the previous assumptions, refer to Vehicle dynamics and control, the bicycle model is used to represent the operating state of the vehicle. A careful revision about Fig. 2 (*i.e.,* Fig.1 in the previous manuscript) and has been made in line 174 and line 780 of the revised manuscript:

"The vehicle kinematics of the bicycle model is based on the Ackerman steering principle under the above-mentioned assumptions [36].

The bicycle model of the vehicle is shown in Fig. 2. In the inertial coordinate system $O-XY$, (X_f, Y_f) and (X_r, Y_r) are the centre coordinates of the front and rear axles of the vehicle, respectively, φ is the yaw angle of the vehicle body, v_r and v_f are the speeds of the centre rear and front axles of the vehicle, respectively, L is the wheelbase of the vehicle. M and N denote the axes of the rear and front axles, respectively, δ is the equivalent front-wheel turning angle, and δ is calculated from the turning angles of the inner and outer steering wheels [37].

”

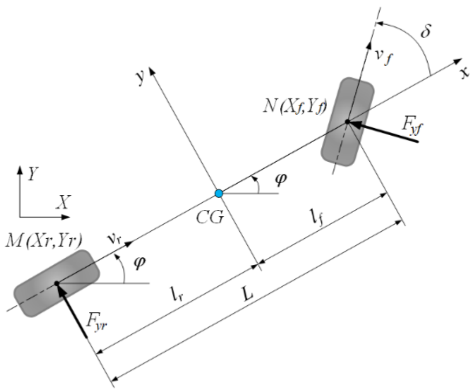

Fig. 2 Kinematics of the bicycle model

“The steering radius and equivalent front-wheel angle are obtained from the vehicle yaw rate and vehicle speed, respectively:

$$\begin{cases} R = \frac{v_r}{\omega} \\ \delta = \arctan(\frac{L}{R}) \end{cases} \tag{8}$$

Where R is the turning radius of the rear wheels.”

Reference:

[36] Rajamani, R., 2011, “Vehicle dynamics and control”, Springer Science & Business Media, Berlin, GER. ISBN: 1461414326

[37] Shekh, M., Umrao, O., and Singh, D., 2020, "Kinematic Analysis of Steering Mechanism: A Review", Proc. Int. Conf. Mech. Energy. Tech., pp. 529-540. DOI: 10.1007/978-981-15-2647-3_48

Comment 4: *Fig. 2 is redundant and citing a paper regarding this matter would be enough.*

Response 4: Thanks for the reviewer's comment. Figure 2 has been deleted, and the serial numbers of other figures have been updated. New references have been applied. Revision has been made in line 181 of the revised manuscript.

"M and N denote the axes of the rear and front axles, respectively, δ is the equivalent front-wheel turning angle, and δ is calculated from the turning angles of the inner and outer steering wheels [37]."

Reference:

[37] Shekh, M., Umrao, O., and Singh, D., 2020, "Kinematic Analysis of Steering Mechanism: A Review", Proc. Int. Conf. Mech. Energy. Tech., pp. 529-540. DOI: 10.1007/978-981-15-2647-3_48

Comment 5: *Fig. 4 is fine but it is too general and not representative of your work. I recommend adding equation number inside each block to make it more clear what is happening in your control strategies.*

Response 5: Thanks for the reviewer's comment. A careful revision has been made in the figure. Equation number has been added to each block. The predictive model denotes Eq. 15, Eq. 43 denotes the future sequence, Eqs. 24-27 denote system constraints, the objective function can be described by Eq. 28 and 42, and the first control parameter is described by Eq. 44. Also, the output of the predictive model and the input of the objective function was added to Fig. 4. Revision in line 789 of the revised manuscript.

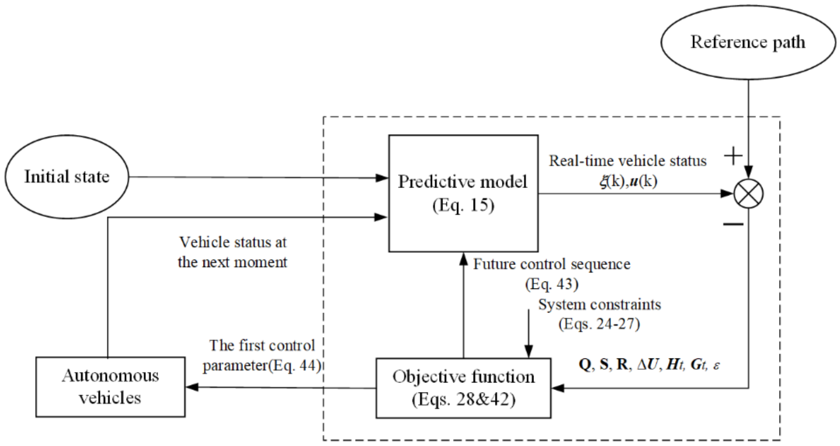

Fig. 4 Diagram of the path-tracking controller

Comment 6: *I recommend adding some unnecessary equation to the appendix such as Eq. 17, 25, and so on.*

Response 6: Thanks for the reviewer's comment. Eqs. 11-14, Eq. 17, 25 have been added to Appendix. Appendix. A presents the coordinates of the vehicle corner point, appendix. B denotes Taylor expansion of Eq. 16 and appendix. C describes the calculation of the state space. Revision has been made in line 577 of the revised manuscript.

"**Appendix**

A. The coordinates of the vehicle corner point. The coordinates of points of A, B, C and D can be calculated as follows:

Point A (X_A, Y_A) at the front-left corner of the vehicle:

$$\begin{cases} X_A = X_r + \cos\left(\varphi + \arctan\left(\frac{W}{2(L+l_f)}\right)\right) * \sqrt{\left(\frac{W}{2}\right)^2 + (L+l_f)^2} \\ Y_A = Y_r + \sin\left(\varphi + \arctan\left(\frac{W}{2(L+l_f)}\right)\right) * \sqrt{\left(\frac{W}{2}\right)^2 + (L+l_f)^2} \end{cases} \quad \text{(A. 1)}$$

Point B (X_B, Y_B) at the front-right corner of the vehicle:

$$\begin{cases} X_B = X_r + \cos\left(\varphi - \arctan\left(\frac{W}{2(L+l_f)}\right)\right) * \sqrt{\left(\frac{W}{2}\right)^2 + (L+l_f)^2} \\ Y_B = Y_r + \sin\left(\varphi - \arctan\left(\frac{W}{2(L+l_f)}\right)\right) * \sqrt{\left(\frac{W}{2}\right)^2 + (L+l_f)^2} \end{cases} \quad \text{(A. 2)}$$

Point C (X_C, Y_C) at the right-rear corner of the vehicle:

$$\begin{cases} X_C = X_r + \sin\left(\varphi - \arctan(\frac{2l_r}{W})\right) * \sqrt{(\frac{W}{2})^2 + {l_r}^2} \\ Y_C = Y_r - \cos\left(\varphi - \arctan(\frac{2l_r}{W})\right) * \sqrt{(\frac{W}{2})^2 + {l_r}^2} \end{cases} \tag{A. 3}$$

Point D (X_D, Y_D) at the rear-left corner of the vehicle:

$$\begin{cases} X_D = X_r - \left(\sin\varphi + \arctan(\frac{2l_r}{W})\right) * \sqrt{(\frac{W}{2})^2 + {l_r}^2} \\ Y_D = Y_r + \left(\cos\varphi + \arctan(\frac{2l_r}{W})\right) * \sqrt{(\frac{W}{2})^2 + {l_r}^2} \end{cases} \tag{A. 4}$$

B. Taylor expansion of Eq. (16). Expanding Eq. 12 as a Taylor series around the path reference point, we get:

$$\begin{aligned} \dot{\boldsymbol{\xi}}_r &= f(\boldsymbol{\xi}_r, \boldsymbol{u}_r) + \left.\frac{\partial f(\xi,u)}{\partial \xi}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (\xi - \xi_r) + \left.\frac{\partial f(\xi,u)}{\partial u}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (u - u_r) \\ &+ \frac{1}{2!} \times \left.\frac{\partial^2 f(\xi,u)}{\partial \xi^2}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (\xi - \xi_r)^2 + \frac{1}{2!} \times \left.\frac{\partial^2 f(\xi,u)}{\partial u^2}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (u - u_r)^2 \\ &+ \cdots + \frac{1}{n!} \times \left.\frac{\partial^n f(\xi,u)}{\partial \xi^n}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (\xi - \xi_r)^n + \frac{1}{n!} \times \left.\frac{\partial^n f(\xi,u)}{\partial u^n}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (u - u_r)^n + R_n(\xi_r, u_r) \end{aligned} \tag{B. 1}$$

Ignoring the higher-order terms, the following can be obtained.

$$\dot{\boldsymbol{\xi}}_r = f(\boldsymbol{\xi}_r, \boldsymbol{u}_r) + \left.\frac{\partial f(\xi,u)}{\partial \xi}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (\xi - \xi_r) + \left.\frac{\partial f(\xi,u)}{\partial u}\right|_{\substack{\xi=\xi_r \\ u=u_r}} (u - u_r) \tag{B. 2}$$

From Eq. 12 and (B. 2), $\dot{\tilde{\xi}}$ is obtained as

$$\dot{\tilde{\boldsymbol{\xi}}}_r = \begin{bmatrix} \dot{X} - \dot{X}_r \\ \dot{Y} - \dot{Y}_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r \sin\varphi_r \\ 0 & 0 & v_r \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X - X_r \\ Y - Y_r \\ \varphi - \varphi_r \end{bmatrix} + \begin{bmatrix} \cos\varphi_r & 0 \\ \sin\varphi_r & 0 \\ \frac{\tan\delta_r}{l} & \frac{v_r}{l\cos^2\delta_r} \end{bmatrix} \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix} \tag{B. 3}$$

Turning it into a general form, we get:

$$\dot{\tilde{\boldsymbol{\xi}}}_{\boldsymbol{r}} = \begin{bmatrix} \dot{x} - \dot{x}_r \\ \dot{y} - \dot{y}_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v_r \sin\varphi_r \\ 0 & 0 & v_r \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x - x_r \\ y - y_r \\ \varphi - \varphi_r \end{bmatrix} + \begin{bmatrix} \cos\varphi_r & 0 \\ \sin\varphi_r & 0 \\ \frac{\tan\delta_r}{l} & \frac{v_r}{l\cos^2\delta_r} \end{bmatrix} \begin{bmatrix} v - v_r \\ \delta - \delta_r \end{bmatrix} \quad \text{(B.4)}$$

C. State space calculation. By substituting Eq. (18) into Eq. (17), the space state $\boldsymbol{\psi}(k)$ is shown as follows:

$$\boldsymbol{\psi}(1) = \tilde{\boldsymbol{A}}_t \psi(0) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0)$$

$$\boldsymbol{\psi}(2) = \tilde{\boldsymbol{A}}_t \boldsymbol{\psi}(1) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1) = \tilde{\boldsymbol{A}}_t \left(\tilde{\boldsymbol{A}}_t \boldsymbol{\psi}(0) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0) \right) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1)$$

$$= \tilde{\boldsymbol{A}}_t^2 \boldsymbol{\psi}(0) + \tilde{\boldsymbol{A}}_t \tilde{\boldsymbol{B}}_t \boldsymbol{\psi}(0) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1)$$

$$\boldsymbol{\psi}(3) = \tilde{\boldsymbol{A}}_t \boldsymbol{\psi}(2) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(2) = \tilde{\boldsymbol{A}}_t^3 \boldsymbol{\psi}(0) + \tilde{\boldsymbol{A}}_t^2 \tilde{\boldsymbol{B}}_t \boldsymbol{\psi}(0) + \tilde{\boldsymbol{A}}_t \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(2)$$

$$\dots\dots$$

$$\boldsymbol{\psi}(k) = \tilde{\boldsymbol{A}}_t^k \boldsymbol{\psi}(0) + \tilde{\boldsymbol{A}}_t^{k-1} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0) + \tilde{\boldsymbol{A}}_t^{k-2} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1)$$

$$+ \dots + \tilde{\boldsymbol{A}}_t \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(k-2) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(k-1)$$

$$= \tilde{\boldsymbol{A}}_t^k \boldsymbol{\psi}(0) + \sum_{j=0}^{k-1} \left(\tilde{\boldsymbol{A}}_t^{k-1-j} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(j) \right) \quad \text{(C. 1)}$$

For the control variable $\boldsymbol{\eta}(k)$:

$$\boldsymbol{\eta}(1) = \tilde{\boldsymbol{C}}_{k,t} \boldsymbol{\psi}(1) = \tilde{\boldsymbol{A}}_t \boldsymbol{\psi}(0) + \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0)$$

$$\boldsymbol{\eta}(2) = \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \boldsymbol{\psi}(1) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1)$$

$$= \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^{k^2} \boldsymbol{\psi}(0) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1)$$

$$\boldsymbol{\eta}(3) = \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \boldsymbol{\psi}(2) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(2)$$

$$= \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^{k^3} \boldsymbol{\psi}(0) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^{k^2} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(0) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(1) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(2)$$

$$\dots\dots$$

$$\boldsymbol{\eta}(k) = \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{A}}_t^k \boldsymbol{\psi}(k + j - 1) + \tilde{\boldsymbol{C}}_{k,t} \tilde{\boldsymbol{B}}_t \boldsymbol{\Delta u}(\mathrm{k} + \mathrm{j} - 1)$$

$$= \widetilde{\boldsymbol{C}}_{k,t}\widetilde{\boldsymbol{A}}_t^{kj}\boldsymbol{\psi}(0) + \widetilde{\boldsymbol{C}}_{k,t}\widetilde{\boldsymbol{A}}_t^{kj-1}\widetilde{\boldsymbol{B}}_t\boldsymbol{\Delta u}(0) + \cdots + \widetilde{\boldsymbol{C}}_{k,t}\widetilde{\boldsymbol{B}}_t\boldsymbol{\Delta u}(\mathrm{k}+\mathrm{j}-1)$$

$$= \widetilde{\boldsymbol{C}}_{k,t}\boldsymbol{\psi}(\mathrm{k}) + \widetilde{\boldsymbol{A}}_t^{k}\boldsymbol{\psi}(0) + \widetilde{\boldsymbol{B}}_t\boldsymbol{\Delta u}(\boldsymbol{0}) = \widetilde{\boldsymbol{C}}_{k,t}\left(\widetilde{\boldsymbol{A}}_t^{k}\boldsymbol{\psi}(0) + \sum_{\boldsymbol{j=0}}^{\boldsymbol{k-1}}\left(\widetilde{\boldsymbol{A}}_t^{k-1-j}\widetilde{\boldsymbol{B}}_t\boldsymbol{\Delta u}(\boldsymbol{j})\right)\right)$$

$$= \widetilde{\boldsymbol{C}}_{k,t}\widetilde{\boldsymbol{A}}_t^{k}\boldsymbol{\psi}(0) + \sum_{j=0}^{k-1}\widetilde{\boldsymbol{C}}_{k,t}\widetilde{\boldsymbol{A}}_t^{k-1-j}\widetilde{\boldsymbol{B}}_t\boldsymbol{\Delta u}(\boldsymbol{j})$$

(C. 2)”

Comment *7: Quation notation is not clear. Please use bolc to distinguish between vector and integer.*

Response *7*: Thanks for the reviewer’s comment. A careful revision has been made in Eqs. 11-43 form lines 236-413 of the revised manuscript using bold to distinguish the vector and matrix. Thanks again for your kind suggestion.

Comment 8: *Adding nomenclature is highly recommended.*

Response 8: Thanks for the reviewer’s comment. Nomenclature has been added to the article in line 574 of the revised manuscript.

Nomenclature

v_f	speed of the centre front axles of the vehicle
v_r	speed of the centre rear axles of the vehicle
δ	steering wheel angle
m	total mass of vehicle
R	turning radius of the rear wheels
l_f	longitudinal distance from c.g. to front tires
l_r	longitudinal distance from c.g. to rear tires
L	total wheel base ($l_f + l_r$)

F_{yf}	lateral tire force on front tires
F_{yr}	lateral tire force on rear tires
X, Y	global axes
CG	the center of gravity of the vehicle
φ	yaw angle of the vehicle body in global axes
(X_f, Y_f)	the centre coordinates of the front axles of the vehicle
(X_r, X_r)	the centre coordinates of the rear axles of the vehicle
ϕ_{steer}	the vehicle steering-wheel angle
W	width of the vehicle
M	the axes of the rear axles
N	the axes of the front axles
α	steering angle of outer wheels
β	steering angle of inner wheels
ω	the vehicle yaw rate
$\dot{\xi}$	the vehicle state space
u	the control variable

Comment 9: *Please add flowchart/ pseudo code for cleridication in Solver section (line 394).*

Response 9: Thanks for the reviewer's comment. The QPKWIK algorithm pseudo code has been added in line 458 of the revised manuscript. Also, the pseudo code of MPC algorithm using QPKWIK slover has been added to clearly describe the proposed MPC algorithm in line 264 of the revised manuscript.

"The QPKWIK algorithm is defined as the Algorithm 2"

Algorithm 2 QPKWIK algorithm

Input: Objective function,all system constraints,
Output: Optimal solution for future control sequence

Step 1 Perform Cholesky decomposition of the Hessian matrix and find the inverse matrix through $chol()$ and $inv()$
Step 2 Use KKT conditions to solve unconstrained solutions
Step 3 Add one constrain to the solution
Step 4 Determine the search direction in the dual sapce
Step 5 Compute the step length in dual space and take step optimization
if Solution satisfies all the constraints **then**
    break
else
    run **Step 1** to **Step 5**
end if

"The MPC algorithm using QPKWIK solver is presented in the Algorithm 1."

Algorithm 1 MPC algorithm using QPKWIK solver

Input: Initial state,Reference path,system constraints
Output: Control varable $\mathbf{u(t)}$

Step 1 Calculate vehicle real state error and control error
Step 2 Update the state space equation $\boldsymbol{\xi(k)}$
Step 3 Calculate $\boldsymbol{\Delta U}$,$\boldsymbol{H_t}$,$\boldsymbol{G_t}$,$\boldsymbol{Q}$,$\boldsymbol{S}$,$\boldsymbol{R}$,$\boldsymbol{\varepsilon}$
Step 4 Minimize objective function using QPKWIK solver under system constraints and **Step 3** output
Step 5 Generate the future control sequence and apply the first control parameter to vehicle
Step 6 Update the next moment status to the predictive model
if Vehicle reachs to the end **then**
    break
else
    run **Step 1** to **Step 6**
end if

Comment 10: *A Please make all the results consistency. Greed colour should be reference for all subplot of Figures.*

Response 10: Thanks for the reviewer's comment. A careful revision has been made in the revised manuscript to make sure that all results are consistent and the color of all subplot of Figures are consistent. Thanks again for your kind suggestion.

Comment 11: *Discussion about results is not comprehensive. I recommend adding more* *about RMSE of error discussion, reason of outperforming, and high error in different references.*

Response 11: Thanks for the reviewer's comment. The RMSE of position and direction has been added to the revised manuscript in Table 2. A careful explanation for the reason of outperforming and high error in different reference has been made in line 506, 516, 530, 545 of the revised manuscript as:

Table 2 The RMSE of position and direction

Driving condition	Error type	RMSE
Straight driving	Position	38.8397 cm
	Direction	0.0393 rad
Right-angle turning	Position	28.2931 cm
	Direction	0.0792 rad
Rectangular driving	Position	49.2486 cm
	Direction	1.4453 rad
Obstacle avoidance driving	Position	41.0903 cm
	Direction	0.1223 rad

"The RMSE of position and direction were 38.8397 cm and 0.393 rad, respectively"

"The RMSE of position and direction were 28.2931 cm and 0.0792 rad. The deviations were largest in at NOC = [500, 550], the early stage of turning the curve. At this time, the steering mechanism of the vehicle has not yet performed the steering action, and there is a delay, so the error at this stage is high. The positional deviation reduced as the heading was quickly adjusted after the controller working."

"The RMSE of position and direction were 49.2486 cm and 1.4453 rad. The main reason is the same as that in section 5.2. The steering mechanism is lagging, resulting in a large deviation."

“The RMSE of position and direction were 41.0903 cm and 0.1223 rad. Similarly, the vehicle has gone through five turning processes, and the lag of the steering mechanism causes a large deviation during the steering process, which is also one of the main reasons for the position error”

After an in-depth amendment to the manuscript, I would like to ask for your further review. We would be very grateful if you could kindly accept this paper for publication in Journal of Dynamic Systems, Measurement and Control.

Many thanks with regards,
Corresponding Authors