2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)

Iterative Trajectory Optimization for Real-Time Motion Planner of Autonomous Driving

Zhichao Liu
University of South Carolina
Columbia, SC, USA
zhichao@email.sc.edu

Duong Le
American Haval Motor Technology
Farmington Hills, MI, USA
duong.le@havalus.com

Kai Zhang
Argo AI
Pittsburgh, PA, USA
carlzhangk@gmail.com

Renxiang Chen
Chongqing Jiao Tong University
Chongqing, China
manlou.yue@126.com

Darong Huang
Chongqing Jiao Tong University
Chongqing, China
drhuang@cqjtu.edu.cn

Bin Zhang
University of South Carolina
Columbia, SC, USA
zhangbin@cec.sc.edu

***Abstract*—For autonomous driving, motion planning and motion control are two significant modules. In this paper, an iterative trajectory optimization (ITO) method is proposed to improve the motion controller's tracking performance and increase the physical and operational feasibility of the motion planning trajectory. Based on the proposed scheme, a feedback connection is built from the motion controller to the motion planner. Different from traditional motion controllers, the motion controller in the proposed scheme is divided into two sub-modules, iterative motion simulator, and motion control operator. In the iterative motion simulator, the vehicle trajectory response is simulated while updating a trajectory offset iteratively. This offset will correct the vehicle response closer to the reference trajectory. After the iterative trajectory adjustment finishes, the simulated tracking error and trajectory offset are sent back to motion planner. The motion planner will first evaluate the simulated trajectory in terms of the response effectiveness, and then send the trajectory with the offset to the motion control operator to calculate the vehicle control maneuver. Comparing with tradition motion planning schemes, the proposed ITO approach can guarantee the trajectory's physical and operational feasibility, make the motion controller have better tracking performance, and have a predictive evaluation on the vehicle response. The simulation results demonstrate the effectiveness of the proposed method.**

***Index Terms*—Iterative optimization, trajectory optimization, motion planning, motion control, and autonomous driving**

I. INTRODUCTION

Autonomous driving has been developing rapidly over the last few years, and such technology offers great promising benefits. Motion planning and control are two important modules in the autonomous driving system. Motion planning needs to configure a collision-free, physically and operationally feasible, and comfortable motion trajectory to enable the vehicle to operate safely and efficiently in traffic; for the motion controller, it is responsible to drive the vehicle close to the motion planning trajectory and keeps the vehicle in safety and comfort under different kinds of disturbance.

The traditional motion planner mainly consists of two parts: trajectory generation and trajectory optimization, as shown in Fig. 1. Trajectory generation first generates motion trajectory candidates while considering road shape, obstacle collision avoidance, and vehicle physical dynamics. Various techniques have been proposed and implemented for trajectory generation including the graph search based method [1], [2], incremental search method (e.g. RRT* [3], [4] and fast marching tree [5]), and optimization based method (e.g. sequential quadratic programming [6] and dynamic programming methods [7]). For trajectory optimization, it makes the trajectory safer, smoother, and more comfortable, and then selects the optimal trajectory based on the cost function. In [8], a conjugate gradient method is presented to smooth the path result; and [9] proposes the separate optimization for both path and speed space; in [10], [11], the model predictive based method is applied to optimize the selected trajectory; in addition, some geometric and polynomial methods are applied to further smooth the trajectory [12]–[14].

However, traditional trajectory optimization methods have several issues. First, the trajectory generation only applies a simple vehicle model to save computational cost. Therefore, it cannot guarantee the generated trajectory being accurately tracked especially at the winding road scenario. Second, traditional motion planners do not provide any compensation to decrease the potential tracking error of the motion controller. Third, because of the unidirectional signal flow between the motion planner and the motion controller, the motion planner does not have a prediction on the deviation of the motion controller tracking. When the deviation accumulates and leads to potential collisions, the motion planner will not have enough time to regenerate a motion trajectory.

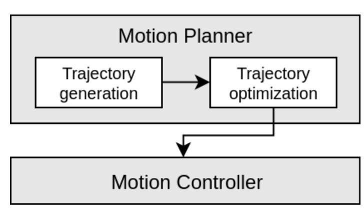

Fig. 1. Traditional structure of motion planner and motion controller

In this paper, an iterative trajectory optimization (ITO)

978-1-7281-0199-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SDPC.2019.00166

approach is proposed by building a feedback connection from the motion controller to the motion planner. In this method, the motion controller is divided into two parts, motion simulator, and motion control operator. Both two parts are based on the same vehicle model and control algorithm design. First, the generated motion trajectory is sent to motion simulator. In this iterative simulation, the trajectory offset is updated from previous simulation cycle to reduce trajectory tracking error, and the tracking error is used to adjust the trajectory response in the current simulation such that the trajectory response will be close to the reference trajectory. After the iterative motion simulation finishes, the final simulator trajectory and trajectory offset are saved in ITO module. The motion planner needs to verify the effectiveness of the final simulator trajectory. If it meets the requirement, the motion planner sends the initial trajectory with the final trajectory offset to motion control operator to calculate the vehicle control maneuver. Otherwise, the motion planner needs to regenerate a motion trajectory and do the process again.

The rest of this paper is organized as follows: Section II provides a brief description of the vehicle dynamic model and LQR motion controller design; Section III introduces the feedback motion planning structure, ITO, simulation evaluation, and real-time application strategy in detail; one simulation result and a number of numerical results are provided to show how the motion planning strategy improves the tracking performance, and in one scenario, the proposed motion planner predicts an operational unfeasible trajectory. The conclusion is summarized in Section V.

II. VEHICLE MODEL AND OPTIMAL CONTROLLER DESIGN

A. Vehicle dynamic model

To simplify the vehicle modeling complexity, the bicycle model as shown in Fig. 2 is used for motion controller design, where l_f and l_r are distances of front and rear wheels from center of gravity (CG), δ is the steering angle, v is vehicle velocity based on CG, β is the slip angle of the vehicle, x and y denote the longitudinal and lateral direction of the vehicle body frame, and X and Y denote the absolute car position inertial coordinates.

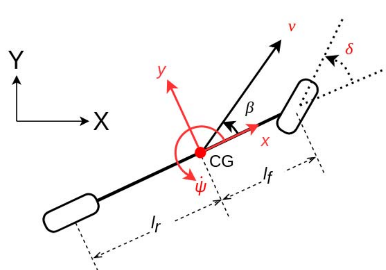

Fig. 2. The simplified vehicle dynamic model

The moving direction of each wheel will be no longer towards their heading directions especially at high speed, so the dynamic model represents the vehicle motion more accurately than the kinematic model [15]. The dynamics of the bicycle model [16] are built from the lateral, longitudinal, and turning degrees of freedom as:

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{xf} + 2F_{xr} \tag{1}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{yf} + 2F_{yr} \tag{2}$$

$$I_z\ddot{\psi} = 2l_fF_{yf} - 2l_rF_{yr} \tag{3}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocity in the vehicle body frame, $\ddot{x}$ and $\ddot{y}$ denote the longitudinal and lateral acceleration, m and I_z denote the vehicle's mass and yaw inertia, and F denotes the force on the vehicle body in which the subscripts x and y denote the longitudinal and lateral directions and f and r denote the front and rear wheels.

To effectively drive the vehicle to the desired motion, the vehicle motion controller is divided into two sub-controllers, lateral controller and longitudinal controller. For the longitudinal dynamic control, a simple PI controller has provided good enough longitudinal velocity and distance control. However, it is still a challenging task for accurate control of lateral dynamic.

B. Vehicle lateral dynamics

Considering vehicle dynamics on lateral direction, the front and rear tire forces are:

$$F_{yf} = C_f(\delta - \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}) \tag{4}$$

$$F_{yr} = C_r(\frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}) \tag{5}$$

where C_f and C_r are the cornering stiffness of front and each rear tire.

The vehicle dynamic model can be converted to the state space based on the lateral error e_1 and heading error e_2 with respect to the desired lateral position y_{des} and the desired heading ψ_{des} as [17]:

$$\dot{x} = Ax + B_1\delta + B_2\dot{\psi}_{des} \tag{6}$$

where

$$x = \begin{bmatrix} e_1 & \dot{e}_1 & e_2 & \dot{e}_2 \end{bmatrix}^T;$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2(C_f+C_r)}{m\dot{x}} & \frac{2(C_f+C_r)}{m} & \frac{2(C_rl_r-C_fl_f)}{m\dot{x}} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2(C_rl_r-C_fl_f)}{I_z\dot{x}} & \frac{2(C_fl_f-C_rl_r)}{I_z} & -\frac{2(C_fl_f^2+C_rl_r^2)}{I_z\dot{x}} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & \frac{2C_f}{m} & 0 & \frac{2C_fl_f}{I_z} \end{bmatrix}^T$$

$$B_2 = \begin{bmatrix} 0 & \frac{C_rl_r-C_fl_f}{m\dot{x}} - \dot{x} & 0 & -\frac{2(C_fl_f^2+C_rl_r^2)}{I_z\dot{x}} \end{bmatrix}^T$$

C. Optimal lateral controller design

Based on the state space model (6), the optimal state feedback control has the form of:

$$\delta^* = -Kx = -(k_1e_1 + k_2\dot{e}_1 + k_3e_2 + k_4\dot{e}_2) \tag{7}$$

To optimize the controller design, a discrete time finite-horizon linear quadratic regulator (LQR) is utilized with a cost function over an n-step horizon window [18]:

$$J = \sum_{i=0}^{n} \left(x_{i+1}^T Q x_{i+1} + u_i^T R u_i \right) \quad (8)$$

where x_i is the discrete state of x, u_i is the discrete input δ in model (6) with the sampling time t_s, and Q and R are diagonal weighting matrices. The optimal feedback gain K is derived as:

$$K = (R + B_{1d}^T P B_{1d})^{-1} B_{1d}^T P A_d \quad (9)$$

and P is the unique positive definite solution to the discrete time algebraic Riccati equation:

$$P = A_d^T P A_d - A_d^T P B_{1d} (R + B_{1d}^T P B_{1d})^{-1} B_{1d}^T P A_d + Q \quad (10)$$

where A_d and B_{1d} are the discrete time state space of A and B_1 in model (6).

In addition, another feed-forward controller will be added to eliminate the influence of the constant term $B_2\dot{\psi}_{des}$ and to cancel the steady-state tracking error [19].

III. ITERATIVE TRAJECTORY OPTIMIZATION MOTION PLANNING AND CONTROL

A. Feedback motion planning structure

As shown in Fig. 1, tradition motion planners mainly contain two steps: trajectory generation and trajectory optimization shown in Fig. 1. In motion planning, a set of path and trajectory are generated first, and then the trajectory optimization chooses the best trajectory based on the cost function among the trajectory candidates and makes the selected trajectory smoother and satisfy a number of constraints.

The traditional data flow between the motion planner and motion controller is unidirectional. The reference motion trajectory is sent to the motion controller to derive the autonomous vehicle operation maneuver (acceleration, deceleration, and steering). The limitations of the traditional motion planners are discussed in the introduction part.

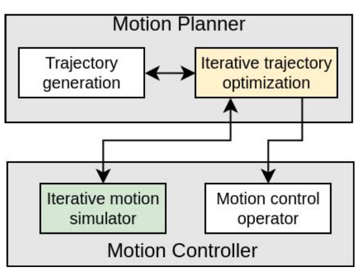

Fig. 3. The proposed structure of ITO motion planning

In this paper, an ITO based motion planning as shown in Fig. 3 is proposed to solve the issues of traditional motion planning optimization. In the proposed structure, the motion controller is divided into two parts, motion control simulator and motion control operator. The initial reference trajectory from the traditional optimization method is first sent to the motion control simulator to get the vehicle simulated trajectory. Based on the idea of iterative learning control [20], [21], this simulation repeats iteratively by updating a trajectory offset. When meeting the termination condition, the final simulated trajectory and final trajectory offset are saved. If the final simulated trajectory does not meet the requirement, e.g. collision-free, the motion planner needs to re-select a trajectory candidate and repeats the optimization process. If the requirement is met, the reference trajectory with the final offset will be sent to the motion control operator. The process detail is introduced in the Section III-B.

B. Iterative trajectory optimization

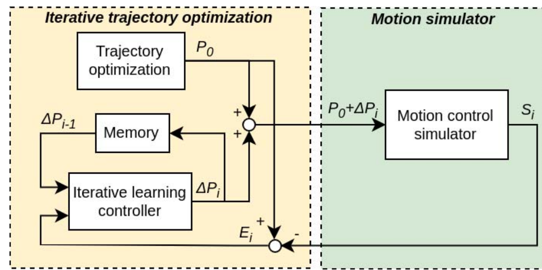

Fig. 4. The block diagram of iterative trajectory optimization and motion control simulator at the i-th iteration

Fig. 4 shows the block diagram of iterative trajectory optimization and motion simulator at the i-th iteration. First, an initial motion trajectory P_0 is generated from the traditional trajectory optimization and then sent to motion control simulator (with initial offset $\Delta P_0 = 0$). The motion trajectory is made up of n trajectory waypoints p_j ($j = 1, 2, ..., n$), which contain the desired vehicle motion status.

$$P_0 = \{p_0, p_1, ..., p_n\} \quad (11)$$

At the i-th iteration, the motion control simulator operates based on the vehicle dynamic model, vehicle operational constraints, and motion controller design in Section II. The simulated motion trajectory S_i is obtained in the form of n simulated motion points s_j ($j = 1, 2, ..., n$) corresponding to each motion trajectory waypoint $\{p_0, p_1, ..., p_n\}_i$.

$$S_i = \{s_0, s_1, ..., s_n\}_i \quad (12)$$

By comparing the difference between initial trajectory P_0 and simulated motion trajectory S_i, the tracking error at the i-th iteration is obtained as:

$$E_i = P_0 - S_i = \{e_1, e_2, ..., e_n\}_i \quad (13)$$

Applying the idea of iterative learning control, a trajectory offset ΔP_i can be derived as:

$$\Delta P_i = \Delta P_{i-1} + \Gamma E_i \quad (14)$$

where ΔP_{i-1} is the trajectory offset of previous iteration, and Γ is the iterative learning control gain matrix. Then the offset ΔP_i will be added to the initial trajectory for the simulation of the next iteration. This process is repeated until a termination condition in simulation evaluation is met.

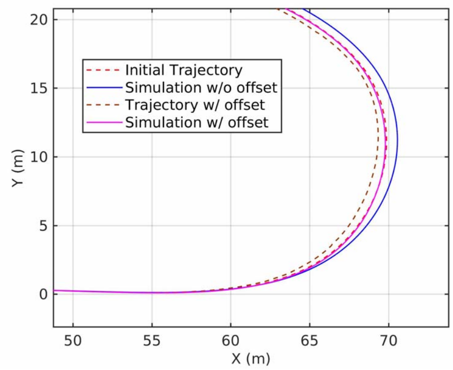

Fig. 5. Illustration of iterative motion planning

Fig. 5 shows a comparison between motion planning with ITO and traditional motion planning. In the example, the initial trajectory is regarded as the reference trajectory. For the LQR motion controller in Section II, the simulated trajectory has a certain deviation with the trajectory path. When applying the ITO, a trajectory with an offset is sent to the motion control operator instead of the initial trajectory. The simulation trajectory under the trajectory with offset is much closer to the initial trajectory comparing with the traditional plan.

C. Terminating condition and trajectory evaluation

The ITO module also needs to consider the terminating condition of the iteration simulation and the simulated trajectory evaluation.

Inside simulation evaluation, an iteration limit i_{limit} and a terminating condition

$$\sum_{k=1}^{n} \left(e_k^T W e_k \right) < \epsilon \tag{15}$$

are added for terminating the iterative simulation, where W is the weight matrix and ϵ is the terminating threshold. When iterative simulation performance is close enough to the initial trajectory P_0 (when condition (15) is met) or reaching the iteration limit i_{limit}, the iterative motion simulation terminates. The trajectory offset ΔP_i and the tracking error E_i are saved as shown in Fig. 3.

The final simulated trajectory, $P_0 + E_i$, is used to motion planning evaluation. If the simulated trajectory still meets the motion planning requirement, e.g. collision-free, it will be used as the desired motion trajectory, and an adjusted trajectory of $(P_0 + \Delta P_i)$ will be sent to motion control operator to calculate the vehicle control maneuver. If the simulated trajectory does not meet the requirement, it will trigger a motion trajectory regeneration, and then repeats the iterative motion simulation process.

D. Real-time application strategy

To apply the proposed ITO motion planning method to real-time applications, a concrete time interleaving of iterative motion planning and motion operator's execution is shown in Fig. 6. The overall planning process starts at time t_0 and finishes at time t_g when the vehicle reaches the goal. The motion planning here generates a motion trajectory for a time horizon t_p, and motion control operator executes an motion planning trajectory for a time horizon t_e ($t_e < t_p$).

For step 0, motion planning generates the initial planning for the motion control operator. At the same time, step 1 of ITO motion planning begins. The motion control simulation runs based on the initial vehicle position at time t_0, and get the ITO motion trajectory planning 1. If the simulated trajectory meets the mission and behavior planning requirement, the motion control operator will follow an offset trajectory of the ITO motion planning 1 and execute for the next time length of t_e. If it does not meet the requirement, the motion planning will regenerate another motion planning and go over the above iterative motion planning again until finding an available motion trajectory. This ITO process will repeat in the entire planning process until the motion planning time window reaches the time of t_g. For the last step, step n, the motion control operator executes the remaining motion trajectory until reaching the target.

The only requirement of this real-time application strategy is that the ITO motion planning process should finish within the time length of t_e.

IV. SIMULATION RESULTS

In this section, the proposed iterative motion planning and controller strategy are tested under several simulation scenarios. The optimization problem is solved in MATLAB on a Windows laptop with an Intel Core i5 CPU at 2.50 GHz in 43.65 milliseconds averaged over 150 tests.

A. Parameter setting

The motion planning and motion controller have different update frequency. The motion planning update frequency is 10 Hz, and the motion controller is 100 Hz in our tests. For each waypoint p_i from motion planning, it contains the desired motion profile of X and Y, heading angle ψ, trajectory curvature κ, velocity v,

$$p_i = [X, Y, \psi, \kappa, v]_i^T \tag{16}$$

For the ITO motion planning, the motion planning horizon t_p is 5s, and the motion controller execution horizon t_e is 1s. The iterative upper limit i_{limit} is set as 20. The iterative learning control gain is set as:

$$\Gamma = [0.1, 0.1, 0.05, 0, 0.05] \tag{17}$$

B. Test case: winding road

The first test is a motion trajectory of a winding road shown in Fig. 7. From the simulation comparison, the trajectory of the simulation with ITO motion planning is almost overlapping

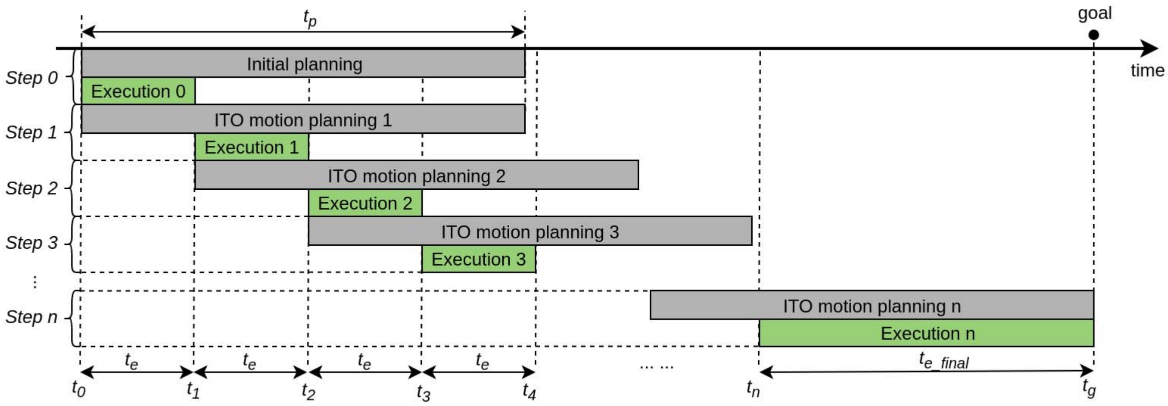

Fig. 6. The time interleaving of iterative motion planning and motion operator's execution

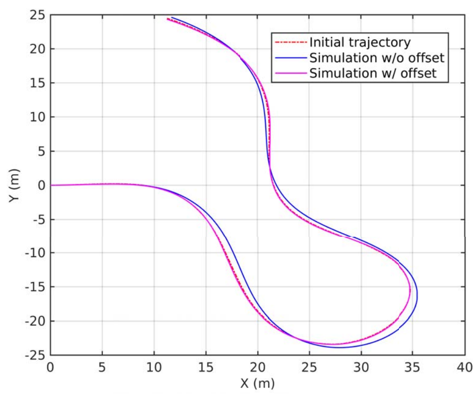

Fig. 7. Test case: motion trajectory and the simulations with and without the trajectory offset from ITO motion planning

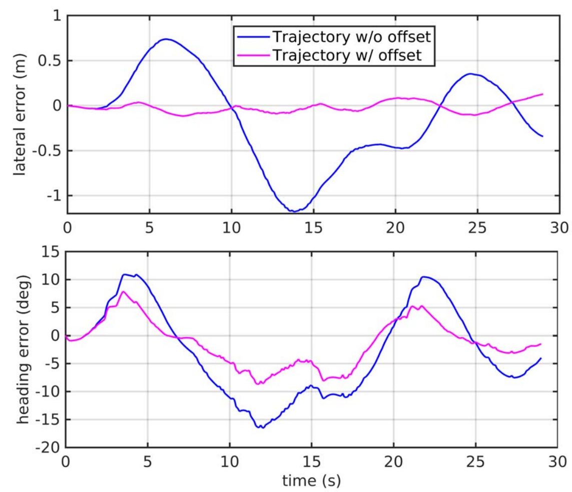

Fig. 8. Test case: lateral and heading errors of the motion simulation with and without the trajectory offset from iterative motion planning

with the initial motion planning trajectory. On the contrary, the simulation trajectory without trajectory offset has a large deviation away from the initial trajectory, especially on the curvy road part.

The comparisons in terms of lateral tracking error and heading error are shown in Fig. 8. The simulation with the ITO motion planning offset has a much smaller lateral error and heading error. The lateral tracking error is within 0.2 m along the whole trajectory, and the maximum heading error is about 8 degree. On the contrary, the simulation under traditional motion planning and motion controller structure show a large tracking error of about 1.2 m the maximum lateral tracking error and 16 degrees the maximum heading error.

When the motion control simulator gets a certain deviation away from the initial trajectory, the iterative learning controller will add a trajectory offset for the simulator until it gets a satisfied tracking performance or reaches the iteration limit. Fig. 9 shows how the motion trajectory offset affects the motion controller. From the figure, the motion control simulator detects a tracking deviation on the winding road part, and then it sends the offset back to motion planning. After doing this process repeatedly, the trajectory offset will converge to a certain value. At last, the motion planning sends the motion trajectory with offset, the green dash line in the figure, to the motion control operator. The operator executes only the first part t_e of the offset trajectory, and then repeat this process for the following trajectory.

C. Numerical result comparison

Table I compares the simulation results in term of lateral tracking error and heading error between traditional motion planning and the proposed ITO motion planning. The comparisons are for three typical driving scenarios: winding road, single-lane change, and double-lane change. The numbers listed are the root-mean-square error in meter or degree.

The comparison shows that the vehicle tracking performance of the proposed motion planning has a significant improvement over that of the traditional one. For the winding road, the proposed method has only 0.0758 m lateral error and 3.6358 degrees heading angle error, compared with 0.4627

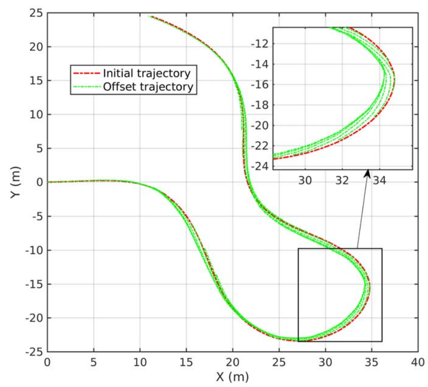

Fig. 9. Test case: initial trajectory from motion planning and trajectory with an offset to motion control operator

TABLE I
TRAJECTORY TRACKING ERROR COMPARISON

	Lateral error (m)		Heading error (deg)	
	Trad.	ITO	Trad.	ITO
Winding road	0.4627	0.0758	7.2801	3.6358
Single-lane change	0.0668	0.0375	1.6207	1.1919
Double-lane change	0.0345	0.0166	1.1175	0.8912

m and 7.2801 degree of traditional motion planning. The improvements are 83.8% and 50.1%, respectively. For the other two scenarios, the proposed method also shows 43.9% and 51.9% improvement on lateral tracking and 26.5% and 20.3% improvement in heading angle tracking.

V. CONCLUSIONS

In this paper, an iterative trajectory optimization method is proposed for real-time motion planner. The proposed method provides a feedback connection from the motion controller to the motion planner, and it brings about three benefits for the motion planner and motion controller. First, the iterative learning controller updates a motion trajectory offset by accumulating the motion control simulator tracking error. By adjusting a trajectory offset iteratively, the motion controller performance gets significant improvement on trajectory tracking. Second, taking advantage of vehicle dynamic based control algorithm, the motion planner is able to generate a physically and operationally feasible trajectory. Third, a vehicle control simulation executes before the trajectory is sent to the motion control operator. The motion planner can predict the vehicle response, have enough time to do a re-planning if the vehicle cannot follow the trajectory well enough. It is worth mentioning that iterative trajectory optimization is also a model-based method similar to model predictive control. The planning and control precision depends on model accuracy. The proposed approach has been implementing in a test vehicle to gather real-world validation results.

REFERENCES

[1] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4889–4895.

[2] J. Ziegler and C. Stiller, "Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 1879–1884.

[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[4] D. Le and E. Plaku, "Guiding sampling-based tree search for motion planning with dynamics via probabilistic roadmap abstractions," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 212–217.

[5] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[6] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive-an autonomous journey on a historic route." *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014.

[7] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 588–597.

[8] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[9] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2061–2067.

[10] P. Liu, B. Paden, and U. Ozguner, "Model predictive trajectory optimization and tracking for on-road autonomous vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3692–3697.

[11] H. Andersen, W. Schwarting, F. Naser, Y. H. Eng, M. H. Ang, D. Rus, and J. Alonso-Mora, "Trajectory optimization for autonomous overtaking with visibility maximization," in *Intelligent Transportation Systems, 2017 IEEE 20th International Conference on*. IEEE, 2017, pp. 1–8.

[12] M. Gloderer and A. Hertle, "Spline-based trajectory optimization for autonomous vehicles with ackerman drive," 2010.

[13] J. Jin, Y. Kim, S. Wee, D. Lee, and N. Gans, "A stable switched-system approach to collision-free wheeled mobile robot navigation," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3-4, pp. 599–616, 2017.

[14] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[15] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 812–818.

[16] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.

[17] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[18] Z. Liu, H. Shi, *et al.*, "Control strategy design based on fuzzy logic and lqr for 3-dof helicopter model," in *2010 International Conference on Intelligent Control and Information Processing*. IEEE, 2010, pp. 262–266.

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

[20] B. Zhang, D. Wang, and Y. Ye, "Wavelet transform-based frequency tuning ILC," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 107–114, 2005.

[21] ——, "Cutoff-frequency phase-in iterative learning control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 3, pp. 681–687, 2009.