2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Model Predictive Trajectory Optimization and Tracking for On-Road Autonomous Vehicles

Peng Liu, Brian Paden, and Umit Ozguner

Abstract— **Motion planning for autonomous vehicles requires spatio-temporal motion plans (i.e. state trajectories) to account for dynamic obstacles. This requires a trajectory tracking control process which faithfully tracks planned trajectories. In this paper, a control scheme is presented which first optimizes a planned trajectory and then tracks the optimized trajectory using a feedback-feedforward controller. The feedforward element is calculated in a model predictive manner with a cost function focusing on driving performance. Stability of the error dynamic is then guaranteed by the design of the feedback-feedforward controller. The tracking performance of the control system is tested in a realistic simulated scenario where the control system must track an evasive lateral maneuver. The proposed controller performs well in simulation and can be easily adapted to different dynamic vehicle models. The uniqueness of the solution to the control synthesis eliminates any nondeterminism that could arise with switching between numerical solvers for the underlying mathematical program.**

I. INTRODUCTION

Autonomous vehicles often decompose the selection of steering, throttle, and braking control signals into a planning process which generates a feasible motion through the perceived scene. This is followed by a control process which executes a local trajectory tracking control policy robust to process noise and load disturbances. Motion planning algorithms for autonomous driving usually simplify the planning task by first planning a geometric path followed by planning a longitudinal velocity profile along the geometric path [1], [2]. Since all motion planning algorithms with approximate completeness guarantees have exponential complexity with respect to state dimension, this decomposition affords significant reduction in computational requirements and planning latency. The decomposition is inherited by the control system which generally has separate lateral and longitudinal control policies [3]. The subtleties of designing controllers in this case have been well studied [4], [5].

The downside to this decomposition is that it implicitly discards effective options available to the planner. For example, if a dynamic obstacle suddenly crosses the vehicle's path, a geometric planner will not reason about the motion of that object and will have to assume a fixed location, or neglect the dynamic obstacle completely; the subsequent longitudinal planner must find a safe option restricted to the selected geometric path. This motivates spatio-temporal motion planning that accounts for predicted states of dynamic obstacles. However, the prohibitive complexity of motion planning for high fidelity models [6] forces motion planning modules to use simplified dynamic models with low dimensional continuous state spaces, sometimes further approximated by finite state models [3], [7]. This places a greater burden on the control process which must execute a trajectory tracking control policy that not only compensates for load disturbances and accounts for sensor noise, but has to account for model difference between the planner and the vehicle's dynamics.

To address such issue, one approach attracting increasing attention is Model Predictive Control (MPC) which incorporates a sophisticated dynamic model to recursively optimize tracking errors over the reference trajectory. MPC approaches for trajectory tracking have been widely investigated for both autonomous driving [2], [8] and semi-autonomous driving cases [9], [10]. Furthermore, if the cost function can be proved as a Lyapunov function, stability of the nominal closed-loop system is guaranteed [11]. However, the cost function for trajectory tracking of autonomous vehicles typically need to account for riding performance such as small yaw rate and smooth change of acceleration, which makes proof of the cost function as a Lyapunov function very difficult, even intractable.

This paper discusses a particular control system design for trajectory tracking which utilizes a feedforward MPC to optimize the reference trajectory with respect to a cost function that does not need to be a Lyapunov function of the closed-loop error dynamic. The proposed feedforward MPC, discussed in Section III, is formulated as a strictly convex optimization problem (SCOP), which results in unique solutions.

Then, a feedback-feedforward controller is proposed based on a time-varying Linear Quadratic Regulator (TVLQR), where the feedforward element and the nominal trajectory are obtained from the feedforward MPC. The reduced computational requirements to run a TVLQR allows the feedback loop to be run at higher frequency than the feedforward path. Moreover, stability of the error dynamic of the nominal system is ensured by carefully choosing coefficients of the cost function in the TVLQR.

Simulations results in Section IV suggest the approach produces satisfactory tracking results in realistic driving scenarios, and with computation times well within the sample rate of the controller. We conclude in Section V that this control system design is a promising approach worth further study including in our research teams repertoire of control

P. Liu and B. Paden are with the Advanced Technology Group, Samsung, San Jose, CA 95134, USA peng2.liu@samsung.com, brian.paden@samsung.com

U. Ozguner is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA ozguner.1@osu.edu

978-1-7281-0322-8/18/$31.00 ©2018 IEEE

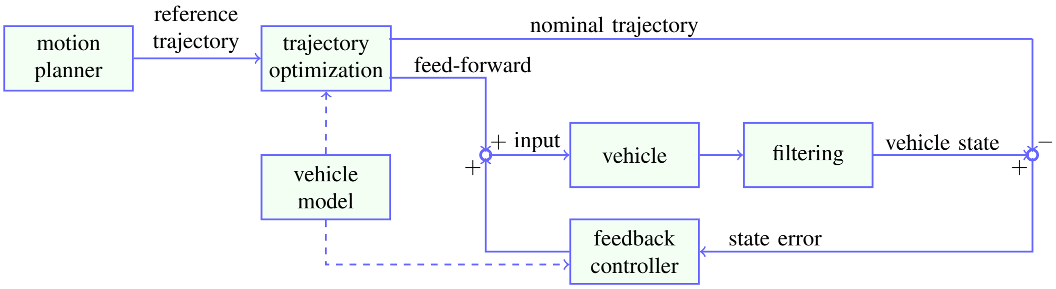

Fig. 1. Diagram of trajectory optimization and the feedback-feedforward scheme. When a reference trajectory from the motion planner is available, the trajectory optimization generates the feedforward input by solving a strictly convex optimization problem that smooths the reference trajectory. The control input to the vehicle is consist of a feedforward term from the trajectory optimization and a feedback term from the state feedback controller.

systems to be tested on or experimental autonomous driving platform.

II. Problem Formulation

A. Trajectory Optimization

A reference trajectory, denoted as $\hat{\mathcal{T}}_t := \{\hat{\mu}_0, \hat{\mu}_1, \ldots, \hat{\mu}_N\}$ with $t \in \mathbb{N}$, published by the planner at instant t is consists of $N+1$ reference states to be reached at uniformly spaced time intervals of duration Δt. The trajectory optimization process is to generate a sequence of control-state pairs satisfying the dynamic model of the vehicle. We refer to the state sequence and the control sequence as the nominal trajectory and the feedforward term if the trajectory optimization problem is feasible. Given the initial state, the vehicle is expected to track the nominal trajectory without error using the feedforward control if no disturbances and model uncertainty appear. Fig. 1 shows the connections of the trajectory optimization to the motion planner and the feedback controller. Instead of the reference trajectory, the optimized nominal trajectory is used by the feedback controller to calculate tracking state-error and to generate the feedback term. The feedforward control together with the feedback term formulates the control input that actuates the vehicle. The state of the vehicle and feedforward control are denoted x and u_{ff} respectively. When comparing states in $\hat{\mathcal{T}}$ and $\mathring{\tau}$ is required, we augment $\hat{\mu}$ to $\mathbb{R}^{\dim(x)}$ with the values of the augmented states set to the equilibrium point, and denote the augmented state as $\hat{x}$. Let $\hat{\tau} \triangleq [\hat{x}(t)^T, \ldots, \hat{x}(t+N)^T]^T$ be the augmented vector of states from the reference trajectory. The trajectory optimization problem is defined,

$$\underset{U_{\text{ff}} \in \mathcal{U} \times \cdots \times \mathcal{U}}{\arg\min} \quad J(\hat{\tau}, \mathring{\tau}, U_{\text{ff}}) \tag{1a}$$

$$\text{subject to } x(t+1) = f_{\text{ff}}(x(t), u_{\text{ff}}(t)) \tag{1b}$$

$$x(t+k) \in \mathcal{X}, \quad k = 0, \ldots, N \tag{1c}$$

$$u(t+k) \in \mathcal{U}, \quad k = 0, \ldots, N-1 \tag{1d}$$

$$U_{\text{ff}} = [u_{\text{ff}}(t)^T, \ldots, u_{\text{ff}}(t+N-1)^T]^T \tag{1e}$$

$$\mathring{\tau} = [x(t)^T, \ldots, x(t+N)^T]^T \tag{1f}$$

$$x(t) = x_0 \tag{1g}$$

where $f_{\text{ff}}(\cdot,\cdot)$ is the feedforward model of the vehicle, U_{ff} is the vector of all feedforward terms over the samples from t to $t+N$. $\mathcal{X}$ and $\mathcal{U}$ are the state constraint set and the feedforward input constraint set, respectively.

While tracking performance requirements can imposed with an inequality constraint on error $||\hat{\tau} - \mathring{\tau}||$, this can make the optimization (1) infeasible. Thus, we focus on formulating tracking performance in the cost function as a soft constraint. Detailed construction of the cost function (1a) and constraints (1b-d) to a strictly convex quadratic program will be discussed in the following two subsections.

B. Vehicle Dynamics

The trajectory optimization problem formulated in II-A relies on the equality constraint (1b) to predict future states and satisfying the differential constraints of the vehicle. The trajectory optimization estimates (1b) from the same dynamic model used in the feedback controller, which is defined as follows

$$\dot{s} = v \cos\theta \tag{2a}$$

$$\dot{y} = v \sin\theta \tag{2b}$$

$$\dot{\theta} = \frac{v}{L} \tan\delta \tag{2c}$$

$$\dot{\delta} = -\lambda_1 \delta + \lambda_1 \delta_{\text{in}} \tag{2d}$$

$$\dot{v} = \alpha \tag{2e}$$

$$\dot{\alpha} = -\lambda_2 \alpha + \lambda_2 \alpha_{\text{in}} \tag{2f}$$

where s, y, θ are the pose at the center point of the rear axle of the vehicle in an inertial coordinate system. δ, v, α, L are the steering-wheel angle, the longitudinal speed, the longitudinal acceleration, and the wheel base of the vehicle, respectively. δ_{in} and α_{in} are the control input of steering angle and the control input of acceleration/deceleration, respectively. A first order inertial response is added to both steering control and acceleration/deceleration control to formulate the system lag, where $\lambda_1 > 0$ and $\lambda_2 > 0$ are response coefficients.

Direct implementation of the nonlinear model (2) in the optimization (1) results in a non-convex problem making real-time trajectory optimization intractable. It can be easily verified that even the one-step quadratic cost $x^T Q x + u_{\text{ff}}^T R u_{\text{ff}}$ with (2) is non-convex in u_{ff} with both Q and R positive definite. In order to obtain a convex cost in U_{ff}, an LTV

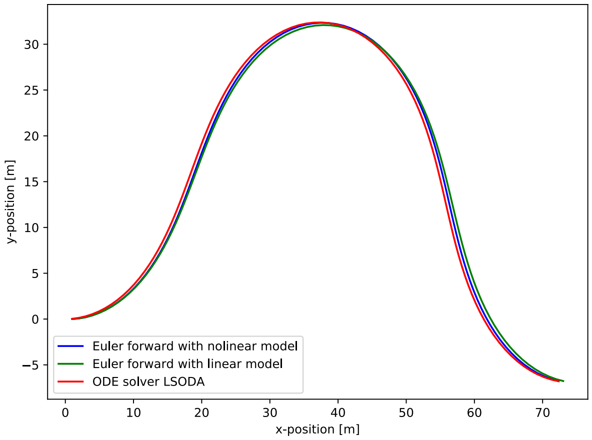

Fig. 2. Verification of the linearized model. Initial condition is set as $[0, 0, 0, 20, 0]^T$ with α_{in} and δ_{in} sampled from sinusoid $2\cos(\frac{t}{5}\pi)$ and $0.2\cos(\frac{2t}{5}\pi)$ respectively, $0 \le t \le 5$, $\Delta t = 0.1s$, $\beta = 0.5$. The red line shows the baseline generated using the ODE solver LSODA. The blue line and the green line denote the Euler forward approximation results of (2) and (3), respectively.

model is used to approximate (2). The linearized discrete-time LTV equation of (2) with $x \triangleq [s, y, \theta, \delta, v, \alpha]^T$ is given as follows.

$$x(t+1) = A(t)x(t) + B(t)u(t) \tag{3}$$

with $A(t) =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cos\theta\Delta t & 0 \\ 0 & 1 & \beta v\Delta t & 0 & (1-\beta)\sin\theta\Delta t & 0 \\ 0 & 0 & 1 & \frac{v\Delta t}{L} & 0 & 0 \\ 0 & 0 & 0 & 1-\lambda_1\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1-\lambda_2\Delta t \end{bmatrix}$$

$$B(t) = B = \begin{bmatrix} 0 & 0 & 0 & \lambda_1\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_2\Delta t \end{bmatrix}^T.$$

where $\beta \in [0, 1]$ is a weighting coefficient. The input matrix B is time-invariant. Values of states in $A(t)$ are assigned using the corresponding states in $\hat{\tau}(t)$. Fig. 2 shows numerical forward simulation of the LTV model (3) compared to (2).

C. Convexity of Quadratic Functions

In the interest of formulating (1) as a SCOP, we focus on the quadratic programming formulations with strictly convex cost functions admitting unique solutions. Uniqueness of the optimal solution ensures that the optimized nominal trajectory is reproducible and solver-invariant[1]. Quadratic programs are well understood and various numerical solvers are available [12]. While the focus is on quadratic programs, the synthesis approach proposed in this paper is applicable to SCOPs with cost functions of different categories. For instance, a SCOP with self-concordant barrier functions available on the constraints can be solved by a generic interior point method.

The following is a brief review of some useful facts and also introduces some of the notation that will be used in subsequent sections. Throughout this work, let $\mathcal{S}_+^n$ (respectively, $\mathcal{S}_{++}^n$) be the set of positive semi-definite (respectively, positive definite) symmetric matrices in $\mathbb{R}^{n\times n}$.

Definition 1 (strict convexity): Let $\mathcal{X}$ be a convex set in $\mathbb{R}^n$, the objective function $J : \mathcal{X} \to \mathbb{R}$ is called strictly convex if $\forall x_1, x_2 \in \mathcal{X}$, $\forall\lambda \in (0, 1)$, $J(\lambda x_1 + (1-\lambda)x_2) < \lambda J(x_1) + (1-\lambda)J(x_2)$.

Definition 2 (quadratic program): A quadratic program over the decision variable $x \in \mathbb{R}^n$ is of the following form:

$$\text{minimize } \frac{1}{2}x^T Hx + F^T x + Y \tag{4a}$$

$$\text{subject to } G_I x \le h \tag{4b}$$

$$G_E x = b \tag{4c}$$

where $H \in \mathcal{S}_+^n$ is the Hessian matrix. $G_I \in \mathbb{R}^{m\times n}$, $G_E \in \mathbb{R}^{p\times n}$ are the element-wise inequality matrix and the element-wise equality matrix, respectively.

If $H \in \mathcal{S}_{++}^n$ and the feasible set is convex, then the cost function is strictly convex and (4) is a strictly convex quadratic program so that the optimal solution x^* exists and is unique. The notation $||x||_R^2$ indicates the quadratic form of x where $R \in \mathcal{S}_+^n$.

III. MAIN RESULTS

A. Cost Function Construction

The dynamics of the LTV system over an N-step planning horizon is written compactly as

$$\mathring{\tau}(t) = \boldsymbol{A}(t)x(t) + \boldsymbol{B}(t)U_{\text{ff}}(t) \tag{5}$$

where

$$\boldsymbol{A}(t) = \begin{bmatrix} I \\ A(t) \\ \vdots \\ \prod_{i=0}^{N-1} A(t+i) \end{bmatrix},$$

$$\mathbf{B}(\mathbf{t}) = \begin{bmatrix} 0 & & & \\ B & & & \\ A(t)B & \ddots & & \\ \vdots & & \ddots & \\ (\prod_{i=0}^{N-2} A(t+i))B & \cdots & \cdots & B \end{bmatrix}.$$

The idea of trajectory optimization is to construct a cost function to meet objectives leveraged by the motion planner and the feedback control system (refer to the connections in Fig. 1). For instance, the motion planner could have riding comfort as an objective and the feedback controller prefers slow changes over the nominal trajectory. It is observed that derivatives of control input are required in the objectives of optimization in addition to control input itself. Therefore, we focus on constructing the cost function of (4) that extends a generic MPC cost function with penalties on differentiation

[1] within the resolution tolerance set by the numerical solver.

of control input in trajectory optimization. The difference matrix $E \in \mathbb{R}^{2N\times 2N}$ over the feedforward sequence U_{ff} is as follows

$$E = \begin{bmatrix} I & & & \\ -I & I & & \\ & \ddots & \ddots & \\ & & -I & I \end{bmatrix},$$

then the change of the control sequence is represented as

$$\Delta^{i+1} U_{\text{ff}} = E \Delta^i U_{\text{ff}} + V_i, \quad i \geq 0 \tag{6}$$

where $\Delta^i U_{\text{ff}}$ is the i-th order difference of U_{ff} with $\Delta^0 U_{\text{ff}} = U_{\text{ff}}$. V_i is the estimation of the initial state that is stored at the previous step. For example, $V_0 = [-u_{\text{ff}}(t-1)^T, 0, \ldots, 0]^T$ at sample t.

Here we focus on various differences of control input since the changes of the state are encoded in the equations for the system dynamics. Given $R_i \in \mathcal{S}^{2N}_{++}$, we have

$$||\Delta^i U_{\text{ff}}||^2_{R_i} = ||E^i U_{\text{ff}} + \sum_{j=0}^{i-1} E^j V_{i-j-1}||^2_{R_i} \tag{7a}$$

$$= ||U_{\text{ff}}||^2_{E^{iT} R_i E^i} + 2\left(\sum_{j=0}^{i-1} E^j V_{i-j-1}\right)^T R_i E^i U_{\text{ff}} \tag{7b}$$

$$+ ||\sum_{j=0}^{i-1} E^j V_{i-j-1}||^2_{R_i}. \tag{7c}$$

The cost function comprising system state and control input is constructed as follows

$$J(\mathring{\tau}, \hat{\tau}, U_{\text{ff}}) = ||\mathring{\tau} - \hat{\tau}||^2_Q + ||U_{\text{ff}}||^2_{R_0} + \sum_{i=1}^{M} ||\Delta^i U_{\text{ff}}||^2_{R_i} \tag{8}$$

where $M \geq 0$ is the difference order of interest. $Q \in \mathcal{S}^{6(N+1)}_{++}$ is the weight coefficient of trajectory tracking deviation. $R_i \in \mathcal{S}^{2N}_{++}, 0 \leq i \leq M$, is the weight coefficient of the i-th order control input. If M is set to 0, (8) is equivalent to the cost function of an MPC scheme for tracking. In particular, we have the quadratic program of trajectory optimization as minimizing (8) subject to dynamic constraints (5), (1c), and (1d). Furthermore, we assume that the constraint sets $\mathcal{U}$ and $\mathcal{X}$ are polyhedra formulated by half planes generated by linear inequalities.

With the cost function designed in the form of (8), we have the following result on uniqueness.

Proposition 1: Given the trajectory optimization problem in the form of minimizing (8) subject to constraints (5), (1c), and (1d), if the polyhedral constraint set is non-empty, then the problem admits a unique optimal solution.
Justification of this observation can be found in Appendix I.

B. Feedback and Feedforward Control

When a new reference trajectory $\hat{\mathcal{T}}$ is available from the motion planner, the trajectory optimization takes the reference trajectory to set up the SCOP with (8), (3), and (1c-d). The solution to the SCOP, U^*_{ff}, and the corresponding nominal trajectory together with a feedback controller form the feedback-feedforward control scheme. State update of the closed-loop system implementing the feedback-feedforward scheme is given as follows

$$x(t+1) = A(t)x(t) + Bu^*_{\text{ff}}(t) + A_{\text{fb}}(t)\tilde{x}(t) + Bu_{\text{fb}}(t) \tag{9}$$

where $u^*_{\text{ff}} \triangleq [U^*_{\text{ff}}(1), U^*_{\text{ff}}(2)]^T$, $\tilde{x} = x - \hat{x}$ is the state tracking error, u_{fb} is the corresponding feedback input. $A_{\text{fb}}(t) = \frac{\partial f(x,u)}{\partial x}|_{x=\hat{x}(t)}$.

The state feedback controller follows the scheme of TVLQR. First, the system is augmented to include the integral of tracking error.

$$\tilde{z}(t+1) = \begin{bmatrix} A_{\text{fb}}(t) & 0 \\ I & I \end{bmatrix} \tilde{z}(t) + \begin{bmatrix} B \\ 0 \end{bmatrix} u_{\text{fb}}(t) \tag{10}$$

where $\tilde{z} = [\tilde{x}^T, \tilde{v}^T]^T$ is the augmented state with $\tilde{v}$ be the integrator state. The TVLQR problem is defined as follows

$$\min_{U_{\text{fb}}} \sum_{k=0}^{N-1} \left(||\hat{z}(t+k)||^2_{\bar{Q}} + ||u_{\text{fb}}(t+k)||^2_{\bar{R}} \right) + ||\tilde{z}(t+N)||^2_{\bar{P}} \tag{11}$$

where $U_{\text{fb}}(t) = [u^T_{\text{fb}}(t), \ldots, u^T_{\text{fb}}(t+N-1)]^T$. $\bar{Q} \in \mathcal{S}^{12}_{+}$, $\bar{R} \in \mathcal{S}^2_{++}$, $\bar{P} \in \mathcal{S}^{12}_{++}$. Closed-loop stability and disturbance rejection properties of the feedback system can be found in [13]. The trajectory optimization is called when a new trajectory from the motion planner is available. In general, the controller updates at a higher rate than the motion planner. Multiple samples in U_{ff} are used in the feedback-feedforward scheme, which is different from solving a quadratic program per control step implemented in an MPC scheme. The TVLQR is responsible for disturbance rejection of the closed-loop system, which requires less computation resource.

IV. SIMULATION RESULTS AND DISCUSSION

A. Trajectory Optimization Results

A sample trajectory is used to test the trajectory optimization. The trajectory consists of states $[s, y, \theta, v]^T$ over a 5-second horizon. The update interval of the trajectory optimization is set to $0.1s$ with the optimization horizon set to $5s$. The cost function is designed to minimize a weighted norm of the tracking error, the feedforward input, and the rate of change of feedforward input. In particular, $Q = \text{diag}(10, 10, 10, 0, 10, 0)$, $R_0 = \text{diag}(0.1, 0.1)$, $R_1 = \text{diag}(1, 1)$. The constraint set $\mathcal{U}$ is set as $-4.0 \leq \alpha_{\text{in}} \leq 2.5$, and $-0.1 \leq \delta_{\text{in}} \leq 0.1$. Using the state model (5) with a 50 time-step horizon, the resulting quadratic program has 100 decision variables and 200 inequality constraints. The optimization problem is solved in MATLAB using **quadprog** on a Windows laptop with an Intel Core i5 CPU at 2.50GHz in 43.65 milliseconds averaged over 150 tests.

The reference trajectory from the planner is piecewise constant in steering angle and velocity consisting of 5 piecewise constant segments. The reference velocity and steering angle

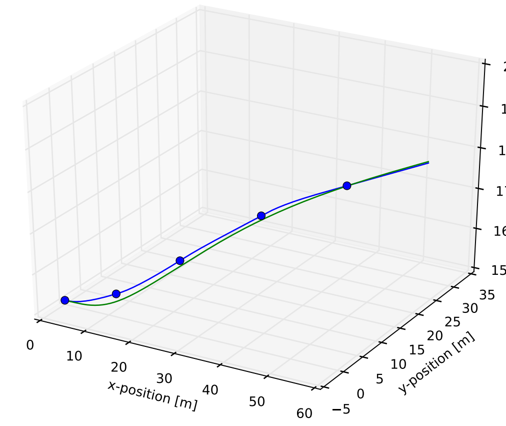

Fig. 3. Optimization of a 5-second reference trajectory. The blue curve shows the coarse trajectory from the motion planner in the pose-velocity space, the green curve shows the optimized trajectory.

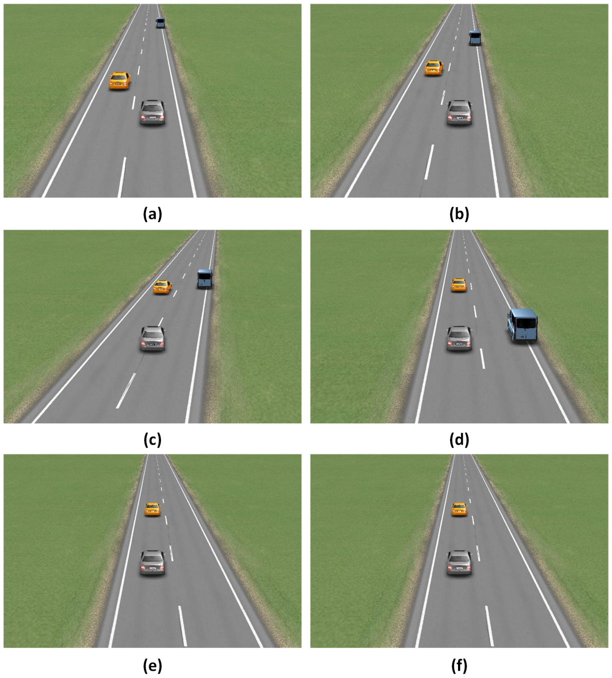

Fig. 4. Screenshots of the simulation scenario. The grey sedan, the blue van, and the yellow sedan are the host vehicle, the pulled-over vehicle, and the cruising vehicle in the target lane, respectively.

over each sub-segment are constant and only change at the starting point of each sub-segment, denoted as blue dots in Fig. 3. Fig. 3 shows the optimized trajectory in the $s-y-v$ space, depicted as the green curve. It is observed that the velocity over the optimized trajectory is adjusted in order to track the s-y position in the reference trajectory.

B. Feedback-feedforward Performance

In order to test the effectiveness of the proposed feedforward-feedback approach for motion trajectory tracking, a test scenario is designed considering an evasion maneuver. The host vehicle in the scenario needs to change lane to avoid a pulled-over vehicle while keeping safe inter-vehicle distance to the car in the target lane. The relative position of vehicles in the simulation is sequentially depicted in Fig. 4.

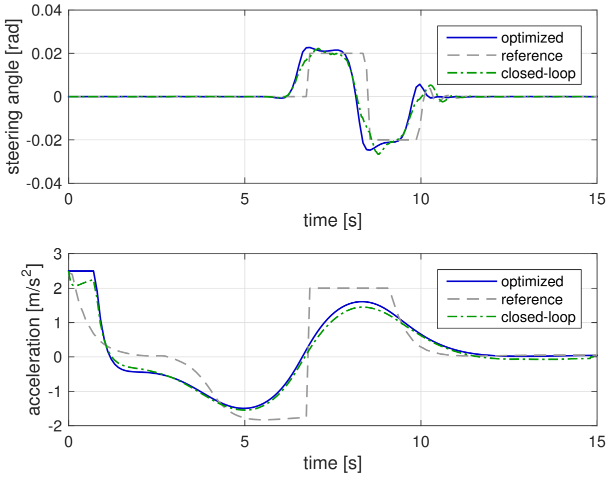

Fig. 5. Control inputs during the evasion maneuver. Grey dash line: input derived from the reference trajectory; blue line: the feedforward input from the trajectory optimization; green dash-dot line: input of the closed-loop system.

The reference trajectory is obtained from the motion planner that minimizes deviations to a desired velocity profile calculated using time-headway and speed limits with forward kinematic simulation. The first 15-second of the planned trajectory is saved and is used as the reference trajectory. The dashed grey line in Fig. 5 shows the estimated feedforward input derived from the reference trajectory using (3) without the first order lag. Steep changes of steering angle and acceleration are observed during $t = 7$ and $t = 10$ where the host vehicle needs to adjust longitudinal velocity while changing to the adjacent lane. Parameters of the trajectory optimization are set as the same to IV-A. Parameters of the TVLQR feedback controller is set as $\bar{Q} = \mathrm{diag}(10, 5, 10, 1, 10, 1\mathrm{e}{-4}, 1\mathrm{e}{-4}, 0, 0, 0, 0)$, $\bar{R} = \mathrm{diag}(1, 5)$. The sampling interval of the TVLQR is set to $0.02s$ with the horizon set equal to the trajectory optimization horizon. The feedback-feedforward scheme is then tested on a nonlinear vehicle model that takes into account acceleration saturation of the powertrain, aero dynamics, and noises on steering and acceleration measurements.

Fig. 3 shows the tracking result of the feedback-feedforward scheme in $s - y$ plane. The corresponding input generated for the closed-loop system is shown in Fig. 5 together with the feedforward term from the trajectory optimization. The feedforward term has been smoothed out during the evasion maneuver as the trajectory optimization takes into account i) change rates of the feedforward term, and ii) a vehicle model sharing the same state space as the feedback controller. Fig. 7 shows the tracking error over longitudinal position, lateral position, vehicle heading, and longitudinal velocity.

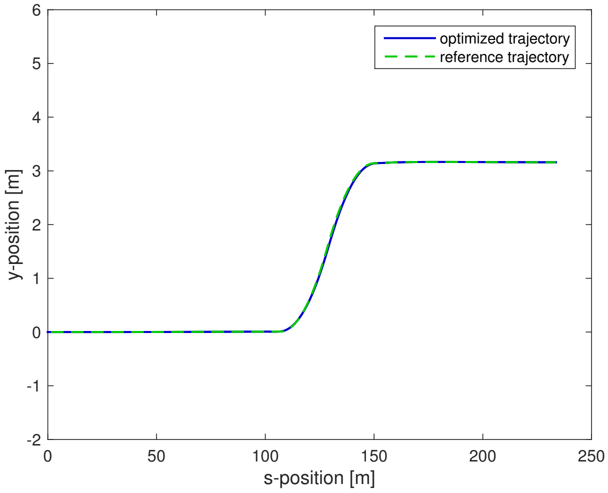

Fig. 6. The tracking profile of the feedback-feedforward scheme in $s - y$ plane.

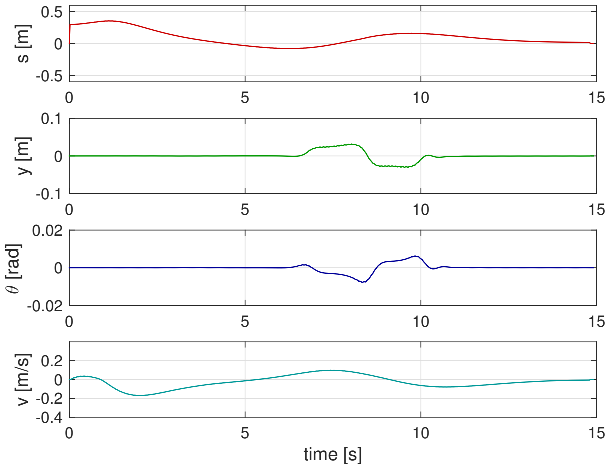

Fig. 7. Trajectory tracking error of the feedback-feedforward scheme over the evasion maneuver.

V. CONCLUSIONS

This paper proposes a trajectory tracking control approach for autonomous vehicles based on a model predictive trajectory optimization to generate a feedforward control and a time varying linear quadratic regulator for feedback. Optimization of a reference trajectory is formulated as a strictly convex quadratic program by leveraging a linearization about the reference trajectory, polyhedral constraints, and a family of strictly convex quadratic cost functions. Additionally, the quadratic cost function is developed taking into account the rate of change of the feedforward input. A feedback-feedforward control scheme is proposed to actuate the vehicle by combining the optimized feedforward input and the feedback input generated by a TVLQR. The trajectory optimization and tracking scheme has been tested in simulation with an evasive maneuver. The proposed approach shows satisfactory tracking results in realistic driving scenarios, and with computation times well within the sample rate of the controller.

APPENDIX I
PROOF OF PROPOSITION 1

R_i is a positive-definite symmetric matrix indicates that R_i can be decomposed as $R_i = L_i^T L_i$. For any vector $x \in \mathbb{R}^{2N}$, $x^T E^{iT} R_i E^i x = x^T E^{iT} L_i^T L_i E^i x = L_i E^i x \cdot L_i E^i x \geq 0$. Therefore, $E^{iT} R_i E^i \in \mathcal{S}_+^n$. Substituting $\mathring{\tau}(t)$ and $\Delta^i U_{\text{ff}}$ in (8) with (5) and (7) respectively turns the cost function (8) into the following form.

$$J = ||\boldsymbol{A}(t)x(t) + \boldsymbol{B}(t)U_{\text{ff}}(t) - \hat{\tau}||_Q^2 + ||U_{\text{ff}}||_{R_0}^2 + \prod_{i=1}^{M} ||(7)||_{R_i}^2$$
$$= U_{\text{ff}}^T H U_{\text{ff}} + 2F^T U_{\text{ff}} + Y$$

where $H = R_0 + \sum_{i=1}^{M} E^{iT} R_i E^i + \boldsymbol{B}(t)^T Q \boldsymbol{B}(t)$, $F^T = \sum_{i=1}^{M} (\sum_{j=0}^{i-1} E^j V_{i-j-1})^T R_i E^i + (\boldsymbol{A}(t)x(t) - \hat{\tau})^T Q \boldsymbol{B}(t)$.

$E^{iT} R_i E^i$, $\boldsymbol{B}(t)^T Q \boldsymbol{B}(t)$ are positive semidefinite and R_0 is positive definite, which indicates that H is positive definite in U_{ff}. For any $U_1, U_2 \in \mathbb{R}^{2N}, U_1 \neq U_2$ and $\lambda \in (0, 1)$,

$$\lambda||U_1||_H^2 + (1-\lambda)||U_2||_H^2 - ||\lambda U_1 + (1-\lambda)U_2||_H^2$$
$$=\lambda(1-\lambda)||U_1 - U_2||_H^2$$

where $\lambda(1 - \lambda) > 0$. Thus, J is strictly convex in U_{ff}. Moreover, the non-empty polyhedral constraint set is a convex set. Therefore, the feasible solution in Proposition 1 is a unique global optimum.

REFERENCES

[1] 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.

[2] 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 driven autonomous journey on a historic route," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, no. 2, pp. 8–20, 2014.

[3] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[4] A. Rupp and M. Stolz, "Survey on control schemes for automated driving on highways," in *Automated Driving*. Springer, 2017, pp. 43–69.

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

[6] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.

[7] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[8] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE transactions on intelligent transportation systems*, vol. 10, no. 1, pp. 92–102, 2009.

[9] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 4239–4244.

[10] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[11] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[12] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[13] W. J. Rugh and W. J. Rugh, *Linear system theory*. prentice hall Upper Saddle River, NJ, 1996, vol. 2.