

# High-Speed Cornering for Autonomous Off-Road Rally Racing

Changxi You     Panagiotis Tsiotras

**Abstract**—High-speed cornering can typically be performed only by an expert driver. Advanced control system design for (semi-) autonomous vehicles requires understanding of such aggressive driving maneuvers in order to be able to take advantage of the full handling capacity of the vehicle, resulting in enhanced stability and improved safety vehicle characteristics. In this paper, we first learn a primitive high-speed cornering maneuver using a series of demonstrations obtained by solving the minimum-time cornering problem subject to different initial conditions. This primitive trajectory indicates that a typical high-speed cornering maneuver can be approximated by three segments, namely, entry corner guiding, steady-state sliding, and exiting. Based on this observation, we divide a high-speed cornering maneuver into three stages. A switch-mode control strategy is designed for these three stages, using a combination of linear and nonlinear control techniques. Conditions for fast trajectory replanning, optimal speed profile generation and tracking control are analyzed. Simulation and experimental results validate the proposed approach on a fifth-scale robotic vehicle.

**Keywords:** Differential flatness, trajectory learning, trajectory planning, high-speed cornering.

## I. INTRODUCTION

Vehicle active safety is one of the most important design requirements in automotive engineering, and many vehicle control techniques have been developed to provide enhanced stability and handling performance, such as differential braking [1], [2], torque vectoring [3], [4], active steering [5], [6] and integrated chassis control [7], [8]. Most of these techniques are designed to minimize, or restrict, the tire sideslip angle within the linear operating region, so that an average driver can maintain control of the vehicle during an emergency [9]. Instead of restricting the lateral dynamics of the vehicle, a better control strategy may be to take advantage of the full handling capacity of the vehicle and perform an accident avoidance maneuver. Such an approach has been studied in [10], where the authors used aggressive, but controlled, yaw motion to mitigate the effects of a T-bone collision. Good knowledge of aggressive driving maneuvers may be required in order to achieve better collision avoidance and mitigation strategies. For this reason, recently there has been a growing interest in the investigation of aggressive driving maneuvers [11]–[15].

The authors in [11]–[13] suggested different definitions of aggressive driving, all of which, however, assumed that the driver intends to inflict psychological or physical harm to the victim. A more precise definition of aggressive driving

was provided by Tasca [14], who performed a review of aggressive driving research. He concluded that a driving behavior is aggressive if it is deliberate, likely to increase the risk of collision, and is motivated by impatience, hostility and/or attempt to save time. Some typical aggressive driving behaviors on the road may include tailgating, weaving in and out of traffic, improper lane change, driving at speeds far in excess of the norm, improper passing etc. [14], [15].

The above aggressive driving behaviors may violate traffic rules and irritate other drivers, and hence are considered to be dangerous. These behaviors do not require the driver to have good driving skills, and the vehicle may not be driving at a high speed. The present study deals instead with a form of aggressive driving that involves driving at high speed through a curve along with speeding/braking actions. Such maneuvers are performed primarily by expert drivers. These aggressive maneuvers may be utilized to improve vehicle safety of (semi-)autonomous vehicles, by duplicating expert driver behavior during emergency avoidance scenarios.

High-speed cornering is a technique used especially in rally racing, during which the vehicle is driving at high sideslip angles while cornering to shave off excess speed. Velenis in [16]–[18] modeled a form of high-speed cornering maneuver called trail braking and showed that trail braking can be generated as the solution of a minimum-time or a maximum-exit velocity cornering problem subject to appropriate boundary conditions. Tavernini [19] investigated minimum-time cornering strategies for a vehicle with different transmission layouts using various road surfaces, and showed that the minimum-time driving strategy under low-friction conditions turned out to be an aggressive high-drift cornering maneuver. Hindiyeh [20] analyzed the stability of the vehicle under high sideslip drifting conditions and revealed the existence of unstable equilibria corresponding to a steady-state cornering maneuver. The unstable equilibria during steady-state cornering were also shown by Yi [21], who proposed a hybrid tire/road model and analyzed the effect of the longitudinal slip on the lateral stability. The above papers indicate that high-speed cornering may be approximately modeled using a steady-state cornering process. Nevertheless, the evidence is not definite and limited work exists on trajectory planning and motion control to generate high-speed cornering maneuver in real time.

In this paper we first show that a high-speed cornering maneuver can be generated as a sequence of three primitive stages, namely, entry guiding, steady-state cornering, and exiting. This decomposition is based on a trajectory learning result that uses a series of demonstrations obtained by solving the minimum-time cornering problem subject to different initial conditions [17]. Specifically, based on these demonstrations, we learn a primitive trajectory using an it-

C. You is a Senior Researcher at Tencent Technology Company, Beijing 100084, China. Email: changxiyou@tencent.com

P. Tsiotras is a Professor at the School of Aerospace Engineering and the Institute for Robotics & Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332-0150, USA. Email: tsiotras@gatech.edu

erative EM algorithm, and by utilizing an unscented Kalman filter (UKF) along with a dynamic time warping (DTW) algorithm to align the time indexing of all demonstrations. This primitive high-speed cornering maneuver indicates the existence of a segment of sustained steady-state cornering and leads to a decomposition of a high-speed cornering maneuver into three stages, namely, guiding, steady-state cornering, and exiting. Subsequently, we design a switch-mode control strategy for the three stages separately, using a combination of linear and nonlinear control techniques (e.g., differential flatness). This methodology allows the on-line computation of a high-speed cornering maneuver, in contrast to the off-line computation reported in [17] that used trajectory optimization.

An early version of the work in this paper was presented in [22], where the three stage decomposition of the high-speed cornering maneuver, along with the calculation of the primitive trajectory, was first proposed. The current work extends the results in [22] along several fronts: First, we improve the design of the controller for steady-state sliding by: a) switching the controller to use an appropriate equilibrium from a set of equilibria having different cornering radii; and b) by including the control of the distance of the vehicle from a certain fixed cornering center, such that one can better track the desired trajectory during sliding. Second, we propose a path planning algorithm that utilizes cubic Bézier curves to minimize the jerk energy, which leads to smooth paths with gradually changing curvature. Third, and in order to generate minimum-time-travel speed profiles, we extend the approach proposed in [23] by using a single track vehicle model, instead of a point mass model. Fourth, we provide experimental validation of the proposed switch-mode controllers on an actual platform. These experiments demonstrate the ability of the proposed methodology for on-board, real-time high-speed cornering trajectory generation.

This paper is organized as follows: Section II formulates the optimization problem and solves a series of demonstrations of high-speed cornering trajectories. Section III obtains a primitive trajectory from these demonstrations using trajectory learning. Section IV introduces differential flatness, shows the flatness property of the vehicle model, and explains how differential flatness can be used for trajectory generation during the entry stage for this problem. Section V introduces high-speed cornering segmentation and plans the trajectory and the speed profile. Sections VI and VII design and implement the controller in simulations. Section VIII generates real-time high-speed cornering using the Auto-Rally experimental platform. Finally, Section IX summarizes the results of this study.

## II. HIGH-SPEED CORNERING TRAJECTORIES

A demonstration of the high-speed cornering trajectory can be obtained by solving either a minimum-time or a maximum-exit velocity path optimization problem [16], [24]. The minimum-time path consists of shortest distance to travel paths and paths of highest average velocity through

the corner. The vehicle decelerates until the point of minimum radius and accelerates past that point. In contrast, the maximum-exit velocity path does not penalize time of travel, but instead maximizes the available time for accelerating, and hence obtains a larger exit corner velocity. To this end, the maximum exit-velocity trajectory utilizes maximum lowest-velocity during cornering and largest available time for accelerating after the vehicle reaches this lowest-velocity. Oversteering, combined with intense braking, may be required to get the vehicle ready for accelerating again as soon as possible. A detailed analysis can be found in [24]. This paper uses the minimum-time solution as the generative demonstration for high-speed cornering, since it better shows the high-speed cornering ability of the vehicle.

It is worth mentioning that one distinctive feature of a high-speed cornering trajectory generated by an expert rally driver is the “late apex”, in which the vehicle exits the corner close to the inner edge of the road [17]. This feature can be observed from the maximum-exit velocity path, as well as the minimum-time path subject to the fixed final positioning of the vehicle after the corner. The vehicle is likely to have to enter the corner from the outer edge of the road in order to observe a trajectory that exhibits “late apex” [24].

### A. Vehicle Modeling

The single-track vehicle model used in this work is shown in Fig. 1. In this figure,  $X_I - O - Y_I$  and  $X_V - C.M. - Y_V$  denote the inertial frame and the body frame (fixed on the vehicle), respectively. Furthermore,  $V_f$ ,  $V_r$  and  $V$  denote the velocities at the front and rear wheels and the vehicle's center of mass (CM), and  $\alpha_f$ ,  $\alpha_r$  and  $\beta$  denote the sideslip angles of the front and rear wheels and the CM, respectively. The parameters  $\ell_f$  and  $\ell_r$  denote the distances of the CM to the front and rear axles,  $f_{ij}$  ( $i = F, R$  and  $j = x, y$ ) denote the longitudinal and lateral friction forces at the front and rear wheels, and  $\psi$  and  $r$  denote the yaw angle and the yaw rate of the vehicle, respectively. Finally,  $\delta$  is the steering angle of the front wheel.

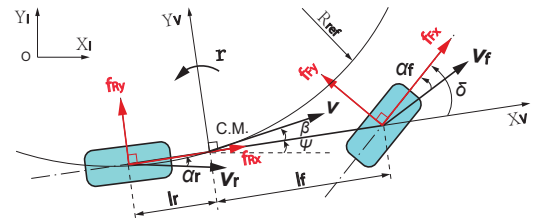


Fig. 1: Vehicle model.

The equations of motion can be expressed in a body-fixed frame with the origin at CM as follows [9]

$$\dot{V} = \frac{1}{m} (f_{Fy} \sin(\beta - \delta) + f_{Fx} \cos(\beta - \delta) + f_{Ry} \sin \beta + f_{Rx} \cos \beta), \quad (1a)$$

$$\dot{\beta} = -r + \frac{1}{mV} (f_{Fy} \cos(\beta - \delta) - f_{Fx} \sin(\beta - \delta) + f_{Ry} \cos \beta - f_{Rx} \sin \beta), \quad (1b)$$

$$\dot{r} = \frac{1}{I_z} ((f_{Fy} \cos \delta + f_{Fx} \sin \delta) \ell_f - f_{Ry} \ell_r), \quad (1c)$$

where  $m$  is the total vehicle mass,  $I_z$  is the moment of inertia of the vehicle about the vertical axis through its CM, and the control is chosen as  $u = [\delta, f_{Fx}, f_{Rx}]^T$ . The lateral tire forces  $f_{Fy}$  and  $f_{Ry}$  are calculated as follows [25], [26]

$$f_{iy} = D_i \sin(C_i \text{atan}(B_i \alpha_i)), \quad i = F, R, \quad (2)$$

where  $D_i, C_i$  and  $B_i$  are constants, and the tire sideslip angles are given by

$$\alpha_F = \delta - \text{atan}\left(\frac{V \sin \beta + \ell_f r}{V \cos \beta}\right), \quad \alpha_R = -\text{atan}\left(\frac{V \sin \beta - \ell_r r}{V \cos \beta}\right). \quad (3)$$

Finally, the kinematic equations of the vehicle CM are given by

$$\dot{X} = V \cos(\beta + \psi), \quad \dot{Y} = V \sin(\beta + \psi), \quad \dot{\psi} = r, \quad (4)$$

where  $X$  and  $Y$  are the coordinates of the vehicle CM in the inertial frame.

### B. Problem Formulation

For 90deg cornering the road geometry is plotted in Fig. 2. In this figure  $S_1$  and  $S_2$  denote the lengths of the two straight road segments before and after the corner, and  $R_{\text{ref}}$  and  $O$  denote the radius and the center of the centerline of the corner, respectively. The vehicle enters at Point A and exits at Point D with certain initial and final velocities.

We momentarily assume that the general case of a high-speed cornering maneuver does not have to show the “late apex” feature, and hence we relax the fixed position condition of the vehicle after the corner. The minimum-time cornering optimal control problem is then formulated as follows

$$\min_u t_f, \quad (5)$$

subject to (1) and (4) and the boundary conditions

$$\begin{aligned} X(t_0) &= X_A, \quad Y(t_0) = Y_A, \quad \psi(t_0) = \frac{\pi}{2}, \\ V(t_0) &= V_0, \quad \beta(t_0) = 0, \quad r(t_0) = 0, \\ X(t_f) &= X_D, \quad \psi(t_f) = \pi, \end{aligned}$$

and the constraints  $-d/2 < \Delta s < d/2$  for all  $t \in [t_0, t_f]$ , where  $t_0$  and  $t_f$  are the initial and final time, respectively;  $(X_A, Y_A)$  are the coordinates of the initial position A,  $X_D$  is the horizontal coordinate of final position D, and  $V_0$  is the initial velocity.  $\Delta s$  is the deviation of the vehicle’s mass center from the centerline of the road, which is calculated by

$$\Delta s = \begin{cases} X - R_{\text{ref}}, & \text{if } \text{atan2}(Y, X) < 0, \\ \sqrt{X^2 + Y^2} - R_{\text{ref}}, & \text{if } 0 \leq \text{atan2}(Y, X) \leq \frac{\pi}{2}, \\ Y - R_{\text{ref}}, & \text{if } \text{atan2}(Y, X) > \frac{\pi}{2}, \end{cases} \quad (6)$$

where  $\text{atan2}(\cdot, \cdot)$  is a function that returns the four quadrant arctangent of the arguments.

TABLE I: Initial conditions.

|                                |                                 |
|--------------------------------|---------------------------------|
| Initial Position A [ (m,m) ]   | (9.5, -5), (10, -5), (10.5, -5) |
| Initial Velocity $V_0$ [ m/s ] | 6, 7, 8, 9, 10                  |

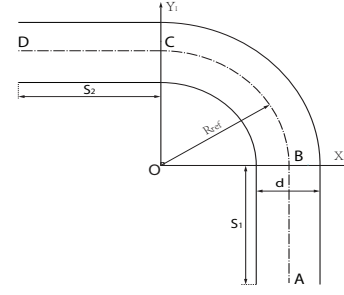


Fig. 2: Road geometry.

### C. Optimal Trajectories

In this section, we specify the geometry of the road and solve for the optimal trajectories of the vehicle subject to different initial conditions. For instance, by assigning the road geometry parameters  $S_1 = S_2 = 5$  [m],  $R_{\text{ref}} = 10$  [m],  $d = 2$  [m] (see Fig. 2), and the fifteen different initial positions and velocities showing in TABLE I, the minimization problem in (5) can be solved numerically. The vehicle model parameters we use to solve (5) are summarized in TABLE IX, which are taken from an actual fifth-scale AutoRally platform (see Section VIII-B for more details on the particular vehicle platform used).

The optimal control solver GPOPS II [27] was used to obtain numerically the solution. The results, for all fifteen trajectories are shown in Fig. 3(a).

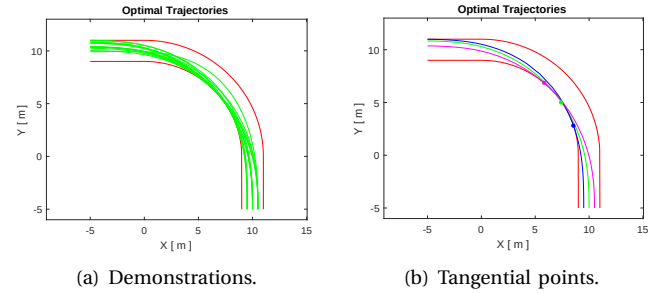


Fig. 3: Optimal trajectories corresponding to different initial positions and velocities.

Based on the results in Fig. 3(a), one sees that the minimum-time cornering solutions tend to pass through the corner close to the inner boundary of the road. We can also see how “late apex” is effected by the choice of different initial positions. Figure 3(b) depicts three of the trajectories and shows their tangential points to the inner edge of the road. All trajectories in Fig. 3(b) were generated using the same initial velocity (6 [m/s]). “Late apex” is more obvious if the initial position is closer to the outer edge of the road. This result indicates that in order to generate high-speed cornering with “late apex” one may require to steer the vehicle to the outer boundary before entering the corner, which is typically how expert human rally drivers initiate a high-speed cornering maneuver [17].

### III. TRAJECTORY LEARNING

In this section we present an algorithm to learn a primitive high-speed cornering trajectory from the series

of demonstrations shown in Fig. 3. The algorithm is based on the approach proposed in [28], which assumes that each demonstration is an independent, noisy observation of some (unknown) primitive trajectory, along with a possible time reparameterization.

#### A. Generative Model

To proceed with the analysis, we suppose that we are given  $M$  representative demonstrations of length  $N_k$  for  $k = 0, 1, \dots, M-1$ . Each trajectory is assumed to be a discrete sequence of states  $x_j^k$  and controls  $u_j^k$ , which are combined into the augmented state vector:

$$y_j^k = \begin{bmatrix} x_j^k \\ u_j^k \end{bmatrix}, \quad j = 0, 1, \dots, N_k - 1, \quad k = 0, 1, \dots, M-1. \quad (7)$$

We then define the “hidden” target trajectory  $z^*$  of length  $T$ , which is denoted by

$$z_t^* = \begin{bmatrix} x_t^* \\ u_t^* \end{bmatrix}, \quad t = 0, 1, \dots, T-1. \quad (8)$$

The hidden trajectory in (8) must satisfy the system dynamics in (1)-(4). Assuming the control in the hidden trajectory does not change fast with time, the hidden trajectory satisfies the following equations,

$$\dot{x}^* = f(x^*, u^*), \quad \dot{u}^* = \eta, \quad \dot{\eta} = 0. \quad (9)$$

After discretizing (9) and assuming some external noise, one obtains

$$z_{t+1}^* = f^*(z_t^*) + w_t^{(z)}, \quad \eta_{t+1} = \eta_t + w_t^{(\eta)}, \quad (10)$$

where  $w_t^{(z)} \sim \mathcal{N}(0, \Sigma^{(z)})$  and  $w_t^{(\eta)} \sim \mathcal{N}(0, \Sigma^{(\eta)})$  are Gaussian process noises. The value of  $\Sigma^{(\eta)}$  determines the smoothness of  $u^*$  for the hidden trajectory. The function  $f^*$  is given by

$$f^*(z_t^*) = \begin{bmatrix} f(x_t^*, u_t^*) \\ \eta_t \end{bmatrix}, \quad t = 0, 1, \dots, T-1. \quad (11)$$

The demonstrations are independently observed from (10). The observations are therefore given by

$$y_j^k = z_{\tau_j^k}^* + w_j^{(y)}, \quad (12)$$

where  $w_j^{(y)} \sim \mathcal{N}(0, \Sigma^{(y)})$  is Gaussian observation noise. Here  $\tau_j^k$  is the time index in the hidden trajectory to which the observation  $y_j^k$  is mapped. The observation model is shown graphically in Fig. 4(a). Since  $\tau_j^k$  are not observed, we assume the following distribution with parameters  $d_i^k$

$$\mathbb{P}(\tau_{j+1}^k | \tau_j^k) = \begin{cases} d_1^k & \text{if } \tau_{j+1}^k - \tau_j^k = 1, \\ d_2^k & \text{if } \tau_{j+1}^k - \tau_j^k = 2, \\ d_3^k & \text{if } \tau_{j+1}^k - \tau_j^k = 3, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

with  $\tau_0^k = 0$ , where  $\sum_{i=1}^3 d_i^k = 1$ , and  $d_i^k \geq 0$ . One can find more discussion on the generative model in [28].

For the sake of simplicity, in this work we apply inter/extrapolation (i.e., spline curves) to each demonstration

with the given time index  $\tau_j^k$ , such that the demonstrations have the same length as the hidden trajectory. This idea is shown in Fig. 4(b).

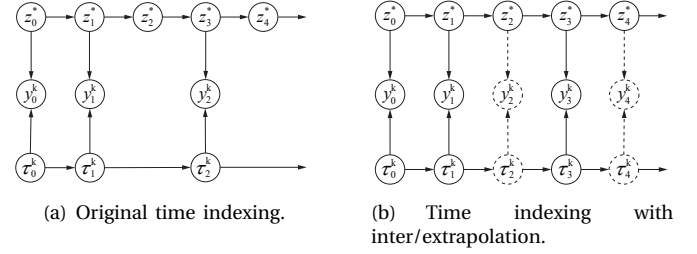


Fig. 4: Graphical observation model with time indexing  $\tau_j^k$ .

By comparing the two plots in Fig. 4, one sees that the unobserved states  $z_2^*$  and  $z_4^*$  in Fig. 4(a) are observed in Fig. 4(b). The new observation in Fig. 4(b) is considered to be reasonably accurate if the sampling interval is small and the demonstrations  $y_j^k$  in (7) are smooth. Since the demonstrations now have the same length as the hidden trajectory, we can rewrite Equation (12) as follows

$$\begin{bmatrix} y_j^0 \\ y_j^1 \\ \vdots \\ y_j^{M-1} \end{bmatrix} = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} z_j^* + \begin{bmatrix} 0w_j^{(y)} \\ 1w_j^{(y)} \\ \vdots \\ M-1w_j^{(y)} \end{bmatrix}, \quad (14)$$

where the  $M$  Gaussian observation noises  $0w_j^{(y)}, \dots, M-1w_j^{(y)}$  are assumed to be independent and identically distributed. The most likely hidden trajectory is then obtained by solving the following maximization problem

$$\max_{\tau, d, \Sigma^{(\cdot)}} \log \mathbb{P}(y; \tau, d, \Sigma^{(\cdot)}), \quad (15)$$

where  $\mathbb{P}$  is the joint likelihood of the observed trajectories  $y$  for the learned parameters  $\tau, d, \Sigma^{(\cdot)}$ .

---

#### Algorithm 1 Trajectory Learning Algorithm

---

**Input:**  $y$

**Output:**  $z^*, \tau^*, d^*, \Sigma^{(\cdot)}$

1:  $\Sigma^{(\cdot)} \leftarrow I, d_j^k \leftarrow \frac{1}{3}, \tau_j^k \leftarrow j \frac{T-1}{N^k-1}$

2: Converge  $\leftarrow$  False

3: **while not** Converge **do**

4: E-step: run UKF to find the distributions  $z_t \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$  using observations  $y$

5: M-step: update the covariances  $\Sigma^{(\cdot)}$  using the standard EM update

6: E-step: run dynamic time warping to find  $\tau$  that maximizes  $\mathbb{P}(\tilde{z}, y, \tau)$ , where  $\tilde{z} = \mu_{t|t-1}$  (or  $\mu_{t|T-1}$ )

7: M-step: estimate  $d$  from  $\tau$  directly

8: **if**  $z$  converges **then**

9: Converge  $\leftarrow$  True

10:  $z^* \leftarrow z, \tau^* \leftarrow \tau, d^* \leftarrow d, \Sigma^{(\cdot)} \leftarrow \Sigma^{(\cdot)}$

---

To optimize the function in (15) we alternatively optimize over  $\tau, d$  and  $\Sigma^{(\cdot)}$ . The procedure of optimization is described as follows. Given the initial values of  $\tau, d$  and  $\Sigma^{(\cdot)}$ , one can implement an UKF to obtain the estimates for the distribution of the “hidden” target trajectory  $z^*$ . We denote

the result as  $\bar{z} \sim \mathcal{N}(\mu_{t|t-1}, \Sigma_{t|t-1})$ . For the current estimate  $\bar{z}$ , the measurements  $y$ , and the initial values of  $\tau$  and  $d$ , one updates the covariances  $\Sigma^{(i)}$  in the M-step using the standard EM algorithm. Next, for the updated covariances  $\Sigma^{(i)}$ , we optimize the time indexing  $\tau$  that maximizes the joint probability  $\mathbb{P}(\bar{z}, y, \tau)$  using a dynamic time warping program. We optimize  $d$  in the last step. The algorithm is summarized in Algorithm 1.

In the trajectory learning algorithm, Step 4) and Step 5) are standard. We show in greater detail Step 6), in which we use a dynamic time warping algorithm to find  $\tau$  to maximize  $\mathbb{P}(\bar{z}, y, \tau)$ , where  $\bar{z}$  is the mode of the distribution of the latent state computed using UKF. Mathematically, we want to solve

$$\begin{aligned}\bar{\tau} &= \arg \max_{\tau} \log \mathbb{P}(\bar{z}, y, \tau) \\ &= \arg \max_{\tau} \log \mathbb{P}(y|\bar{z}, \tau) \mathbb{P}(\bar{z}) \mathbb{P}(\tau) \\ &= \arg \max_{\tau} \log \mathbb{P}(y|\bar{z}, \tau) \mathbb{P}(\tau),\end{aligned}\quad (16)$$

where  $\bar{z}$  is fixed to  $\mu_{t|t-1}$ . Using  $\ell(\cdot)$  to denote the log-likelihood, then  $\bar{\tau}$  in (16) is computed by

$$\bar{\tau} = \arg \max_{\tau} \sum_{k=0}^{M-1} \left( \sum_{j=0}^{N_k-1} \ell(y_j^k | \bar{z}_{\tau_j^k}, \tau_j^k) + \sum_{j=1}^{N_k-1} \ell(\tau_j^k | \tau_{j-1}^k) \right), \quad (17)$$

where  $\ell(y_j^k | \bar{z}_{\tau_j^k}, \tau_j^k)$  is given by

$$\ell(y_j^k | \bar{z}_{\tau_j^k}, \tau_j^k) = \ell(y_j^k | \mu_{\tau_j^k}, \tau_j^k) = (2\pi)^{-\frac{L}{2}} \|\Sigma_{\tau_j^k}^{-1}\|^{\frac{1}{2}} e^{-\frac{1}{2} (y_j^k - \mu_{\tau_j^k})^T \Sigma_{\tau_j^k}^{-1} (y_j^k - \mu_{\tau_j^k})}, \quad (18)$$

where  $L$  is the length of the vector  $\bar{z}_{\tau_j^k}$ ,  $\mu_{\tau_j^k} = \mu_{\tau_j^k | \tau_{j-1}^k}$  and  $\Sigma_{\tau_j^k} = \Sigma_{\tau_j^k | \tau_{j-1}^k}$  are calculated in Step 2) of the Trajectory Learning Algorithm. Since the  $M$  observations are independent, the likelihoods for the  $M$  observations in (17) can be maximized separately. The maximization problem in (17) can be solved using dynamic programming.

### B. Primitive High-Speed Cornering Trajectory

After implementing the previous Trajectory Learning Algorithm we obtain the result shown in Fig. 5. The primitive trajectory in Fig. 5 is likely to share a segment of a circle that is tangent to the inner road boundary, and the vehicle seems to keep a constant side-slip angle  $\beta$  during the cornering. This observation leads to the conjecture that there is a segment of steady-state cornering in the primitive trajectory.

Steady-state cornering is characterized by a trajectory of constant radius  $R$ , negotiated at a constant speed  $V$ , constant yaw rate, and constant side-slip angle [9], namely,  $R = R^{\text{ss}}$ ,  $V = V^{\text{ss}}$ ,  $\beta = \beta^{\text{ss}}$ ,  $r = r^{\text{ss}} = V^{\text{ss}}/R^{\text{ss}}$ , where the steady-state triplet  $(V^{\text{ss}}, \beta^{\text{ss}}, r^{\text{ss}})$  are obtained using (1) by letting  $\dot{V} = \dot{\beta} = \dot{r} = 0$  as follows

$$0 = \frac{1}{m} (f_{\text{Fy}} \sin \sigma^{\text{ss}} + f_{\text{Fx}}^{\text{ss}} \cos \sigma^{\text{ss}} + f_{\text{Ry}} \sin \beta^{\text{ss}} + f_{\text{Rx}}^{\text{ss}} \cos \beta^{\text{ss}}), \quad (19a)$$

$$0 = -r^{\text{ss}} + \frac{1}{mV^{\text{ss}}} (f_{\text{Fy}} \cos \sigma^{\text{ss}} - f_{\text{Fx}}^{\text{ss}} \sin \sigma^{\text{ss}} + f_{\text{Ry}} \cos \beta^{\text{ss}} - f_{\text{Rx}}^{\text{ss}} \sin \beta^{\text{ss}}), \quad (19b)$$

$$0 = \frac{1}{I_z} ((f_{\text{Fy}} \cos \delta^{\text{ss}} + f_{\text{Fx}}^{\text{ss}} \sin \delta^{\text{ss}}) \ell_{\text{f}} - f_{\text{Ry}} \ell_{\text{r}}), \quad (19c)$$

where  $\sigma^{\text{ss}} = \beta^{\text{ss}} - \delta^{\text{ss}}$ , and the control components  $\delta^{\text{ss}}, f_{\text{Fx}}^{\text{ss}}$  and  $f_{\text{Rx}}^{\text{ss}}$  are the steering angle and longitudinal tire forces during steady-state cornering. By denoting the constant vector  $u^{\text{ss}} = [\delta^{\text{ss}}, f_{\text{Fx}}^{\text{ss}}, f_{\text{Rx}}^{\text{ss}}]^T$ , the equilibrium  $(V^{\text{ss}}, \beta^{\text{ss}}, r^{\text{ss}})$  with the corresponding control  $u^{\text{ss}}$  are calculated following the equations in Section 3 of [9].

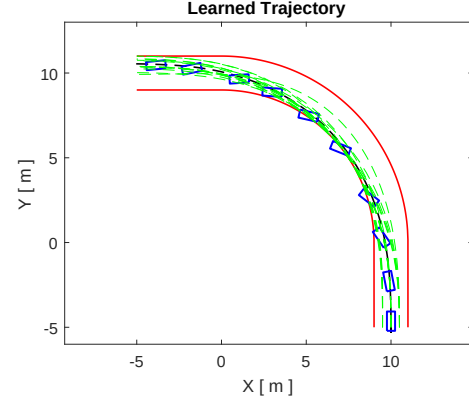


Fig. 5: Multiple demonstrations and the learned primitive trajectory.

In order to see the existence of steady-state cornering in the primitive trajectory, we plot the velocity, side-slip angle and yaw rate of the primitive maneuver in Fig. 6. From Fig. 6 one observes that  $V, \beta$  and  $r$  are quite close to constants from 1.4 second to 2.4 second (stage B), which is typical of steady-state cornering. One also notices that steady-state cornering begins after the vehicle has entered the corner for a while and terminates before the vehicle has left the corner. The large values of  $V$  and  $\beta$  indicate that the primitive maneuver indeed performs a high-speed, high-sideslip cornering maneuver.

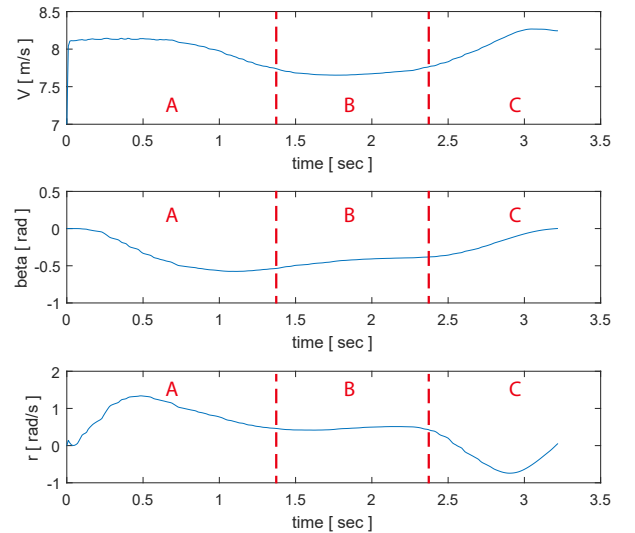


Fig. 6: The velocity, side-slip and yaw motion of the learned primitive.

Based on the result in Fig. 6, we may divide a high-speed cornering trajectory into three segments. The first segment allows the vehicle to complete the transition from

straight-line driving to steady-state cornering. Before the vehicle enters the corner, the vehicle gradually changes its velocity using appropriate steering and accelerating/braking commands, until it reaches the target steady state near the entry of the corner. The vehicle then maintains steady-state cornering during the second segment, until it gets (close) to the exit of the corner. Finally, in the last segment, the vehicle stops the steady-state cornering and is steered back to straight-line driving after the vehicle exits the corner.

#### IV. DIFFERENTIALLY FLATNESS TRAJECTORY GENERATION

In this section, we briefly introduce the concept of differential flatness and show that the vehicle model (1) is differentially flat. This property of the vehicle will allow us to plan the vehicle's trajectory and build the corresponding control for the vehicle using the desired output of the system.

##### A. Differential Flatness

Differential flatness theory was introduced in the late 1980s by Michel Fliess and his colleagues and it provides an efficient solution to several nonlinear control and state estimation problems [29], [30].

A nonlinear system  $\dot{x} = f(x, u)$  is differentially flat if there exists an output variable  $y \in \mathbb{R}^m$  in the following form

$$y = h(x, u, \dot{u}, \ddot{u}, \dots, u^{(r)}), \quad (20)$$

where  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$ , such that

$$x = \Psi(y, \dot{y}, \ddot{y}, \dots, y^{(r)}), \quad (21a)$$

$$u = \Phi(y, \dot{y}, \ddot{y}, \dots, y^{(r)}), \quad (21b)$$

where  $f$  is a smooth vector field and  $h, \Psi$  and  $\Phi$  are smooth functions. Equations (21) imply that the state  $x$  and the control  $u$  can be recovered using the  $m$  algebraic output variables  $y_i$ ,  $i = 1, 2, \dots, m$ . The output  $y$  given in (20) is the flat output of the system.

The authors in [25], [31]–[33] designed a tracking control based on the differential flatness property of the single-track vehicle model, but only [33] used this for planning trajectories. Although [33] generated reference trajectories for different cases from geometric path constraints, the assumptions were strong and the tires were required to operate within their linear regions. In this paper we plan the motion of a vehicle that involves a desired steady-state cornering, and we analyze the conditions on the trajectory in order to be recovered using differential flatness, while taking into account the road condition and the steering capacity of the vehicle.

##### B. Differential Flatness of Vehicle Model

In this section we show that the equations (1) of the vehicle model are differentially flat with respect to a particular output. This property is stated in Theorem 4.1.

*Theorem 4.1:* ([25]) The vehicle model in (1) is differentially flat with respect to the following output

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} V \cos \beta \\ V \sin \beta - \left( I_z / m \ell_f \right) r \end{bmatrix}. \quad (22)$$

*Proof:* To show Theorem 4.1, we just need to find the expressions of the state  $x$  and control  $u$  in the form of (21). To this end, recall from (2) that the lateral tire forces  $f_{iy}$  ( $i = F, R$ ) are smooth functions of the sideslip angles  $\alpha_i$  at the front and rear wheels, respectively. We can then derive the state of the system in (1) in terms of the flat output in (22) as follows

$$V = \sqrt{y_1^2 + \left( y_2 + \frac{I_z r}{m \ell_f} \right)^2}, \quad \beta = \text{atan} \left( \frac{m \ell_f y_2 + I_z r}{m \ell_f y_1} \right), \quad (23)$$

where  $r = r(y_1, y_2, \dot{y}_2)$  is given by solving the following implicit equation,

$$\begin{aligned} & D_R \sin \left( C_R \text{atan} \left( B_R \text{atan} \left( \frac{(m \ell_f \ell_r - I_z) r - m \ell_f y_2}{m \ell_f y_1} \right) \right) \right) \\ & - \frac{m \ell_f (\dot{y}_2 + y_1 r)}{\ell_f + \ell_r} \triangleq g(y_1, y_2, \dot{y}_2, r) = 0. \end{aligned} \quad (24)$$

Equations (23)–(24) give the expressions of the state  $x$  in terms of the flat output in (22). The control  $u = [\delta, f_{Fx}, f_{Rx}]^T$  can be recovered by solving the following equations

$$\begin{aligned} & f_{Fy}(\delta, y_1, y_2, \dot{y}_2) \cos \delta \ell_f + f_{Fx} \sin \delta \ell_f - I_z \dot{r}(y_1, \dot{y}_1, y_2, \dot{y}_2, \ddot{y}_2) \\ & - f_{Ry}(y_1, y_2, \dot{y}_2) \ell_r = 0, \end{aligned} \quad (25a)$$

$$\begin{aligned} & m \left( \dot{y}_1 - y_2 r(y_1, y_2, \dot{y}_2) - \frac{I_z}{m \ell_f} r^2(y_1, y_2, \dot{y}_2) \right) - f_{Fx} \cos \delta - f_{Rx} \\ & + f_{Fy}(\delta, y_1, y_2, \dot{y}_2) \sin \delta = 0, \end{aligned} \quad (25b)$$

$$\Gamma(f_{Fx}, f_{Rx}) = 0, \quad (25c)$$

where the rear wheel lateral tire force is given by

$$f_{Ry}(y_1, y_2, \dot{y}_2) = \frac{\dot{y}_2 m \ell_f + y_1 r(y_1, y_2, \dot{y}_2) m \ell_f}{\ell_f + \ell_r}, \quad (26)$$

and where  $\Gamma(f_{Fx}, f_{Rx})$  in (25c) is a force distribution function related to the specific drive type of the vehicle. Proper choices of  $\Gamma(f_{Fx}, f_{Rx})$  for different drive types are as follows: a) All-Wheel-Drive: All four wheels provide accelerating and braking forces,

$$\Gamma(f_{Fx}, f_{Rx}) = \kappa f_{Fx} - (1 - \kappa) f_{Rx}, \quad (27)$$

where  $\kappa \in [0, 1]$  is a constant that need to be specified.

b) Front-Wheel-Drive: All four wheels provide braking force but only the front wheels provide accelerating force,

$$\Gamma(f_{Fx}, f_{Rx}) = \begin{cases} f_{Rx}, & \text{if } f_{Fx} \geq 0, \\ \kappa f_{Fx} - (1 - \kappa) f_{Rx}, & \text{if } f_{Fx} < 0. \end{cases} \quad (28)$$

c) Rear-Wheel-Drive: All four wheels provide braking force but only the rear wheels provide accelerating force,

$$\Gamma(f_{Fx}, f_{Rx}) = \begin{cases} f_{Fx}, & \text{if } f_{Rx} \geq 0, \\ \kappa f_{Fx} - (1 - \kappa) f_{Rx}, & \text{if } f_{Rx} < 0. \end{cases} \quad (29)$$

Specifically, the Auto-Rally vehicle model introduced in Section VIII-B uses only the rear wheels for accelerating and braking and hence  $f_{Fx} = 0$ . We can then simplify (25) and determine the remaining control variables  $\delta$  and  $f_{Rx}$

by solving the following equations

$$\begin{aligned} f_{Fy}(\delta, y_1, y_2, \dot{y}_2) \cos \delta \ell_f - I_z \dot{r}(y_1, \dot{y}_1, y_2, \dot{y}_2) \\ - f_{Ry}(y_1, y_2, \dot{y}_2) \ell_r = 0, \end{aligned} \quad (30a)$$

$$\begin{aligned} m \left( \dot{y}_1 - y_2 r(y_1, y_2, \dot{y}_2) - \frac{I_z}{m \ell_f} r^2(y_1, y_2, \dot{y}_2) \right) - f_{Rx} \\ + f_{Fy}(\delta, y_1, y_2, \dot{y}_2) \sin \delta = 0. \end{aligned} \quad (30b)$$

Equations(22)-(30) show the differential flatness of the vehicle model in (1). In the following sections, we design a tracking controller based on this differential flatness property of the single-track vehicle model in order to generate the high-speed cornering maneuver.

## V. TRAJECTORY PLANNING

As mentioned above, we will assume that a high-speed cornering maneuver consists of three stages: 1) An entry stage before the vehicle enters the corner; 2) a sliding stage where the vehicle passes through the corner at a steady-state; and 3) an exiting stage after the vehicle leaves the corner.

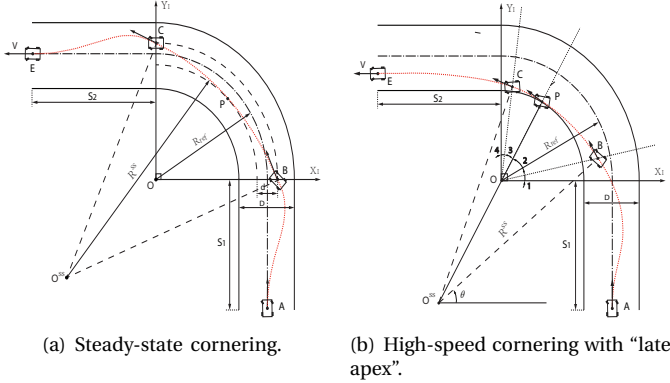


Fig. 7: Road geometry and high-speed cornering trajectory.

We consider two different scenarios of high-speed high-slip cornering maneuvers, as shown in Fig. 7. In the first scenario, we assume that the vehicle is required to pass through the corner next to the road centerline. Figure 7(a) shows the entire cornering process for this scenario. In this figure  $R_{ref}$  and  $O$  denote the radius and the center of the centerline of the corner,  $R^{ss}$  and  $O^{ss}$  denote the radius and the center of the arc  $BPC$ , and  $D$  denotes the road width.  $d$  denotes the distance between the inner and outer constraint lines on the vehicle's trajectory in the corner, such that the vehicle keeps close to the road centerline. The points  $B$  and  $C$  are located on the outer constraint line and indicate the entering and exiting position of the corner, and  $P$  is a tangent point on the inner constraint curve. The vehicle starts at position  $A$  and enters the corner from position  $B$ . The trajectory of the vehicle is divided into three parts: the arc trajectory connecting  $BC$  for the steady-state cornering; and the two pieces of the trajectories connecting  $AB$  and  $CE$  where the vehicle enters/leaves the corner.

The second scenario shows the feature of “late apex”, where the vehicle exits the corner next to the inner edge of the road. This scenario does not require the vehicle to be close to the road centerline. Figure 7(b) shows this maneuver, which postpones the tangent point  $P$  to the second half of the corner and removes the inner/outer constraint lines. Similarly with the first scenario, the steady-state cornering begins from  $B$ , passes through  $P$  and ends up at  $C$ . We introduce  $\angle 1, \dots, \angle 4$  to describe the position and length of the arc  $BPC$ . The nonnegative angles  $\angle 1$  and  $\angle 4$  denote the vehicle's late entering and early exiting positions of the steady-state cornering.

### A. Steady State Cornering

With the given road geometry shown in Fig. 7, one can first determine the radius  $R^{ss}$  of the arc  $BPC$  for steady state cornering. By assigning appropriate values of  $\angle 1, \dots, \angle 4$ , one can change the position of the tangent point  $P$  and the length of the arc  $BPC$ , and hence obtain a trajectory with the desired “late apex” feature.

We denote the steady-state of the vehicle using the triplet  $x^{ss} = [V^{ss}, \beta^{ss}, r^{ss}]^T$ . As mentioned earlier, we can determine the equilibrium  $x^{ss}$  with the desired cornering speed  $V^{ss}$ , or the desired sideslip angle  $\beta^{ss}$ , following (19) and the equations in Section 3 of [9].

In this paper we calculate the equilibrium of the steady-state cornering and design the sliding mode trajectory offline.

### B. Guiding Trajectory

The steady-state triplet  $x^{ss} = [V^{ss}, \beta^{ss}, r^{ss}]^T$  obtained in Section V-A defines the state of the vehicle at position  $B$ . We assume that the initial state of the vehicle is known at  $A$  and we use the notations in TABLE II to represent the boundary conditions.

TABLE II: Boundary conditions

|            | Speed    | Slip angle   | Yaw rate | X position | Y position | Yaw angle |
|------------|----------|--------------|----------|------------|------------|-----------|
| Position A | $V_A$    | $\beta_A$    | $r_A$    | $X_A$      | $Y_A$      | $\psi_A$  |
| Position B | $V^{ss}$ | $\beta^{ss}$ | $r^{ss}$ | $X_B$      | $Y_B$      | $\psi_B$  |

In order to determine a feasible trajectory of the vehicle from  $A$  to  $B$ , we claim that the trajectory must be designed to satisfy equation (24). To clarify the idea, let us assume that one has designed a trajectory  $x(t) = [V(t), \beta(t), r(t)]^T$ ,  $t \in [t_0, t_f]$ , and has calculated the flat outputs  $y_1(t)$  and  $y_2(t)$  using the trajectory  $x(t)$  following (22). However, it is not possible to recover  $x(t) = [V(t), \beta(t), r(t)]^T$  using the flat outputs  $y_1(t)$  and  $y_2(t)$ , unless  $x(t)$  is designed to satisfy equation (24). We summarize this condition in Proposition 5.1.

**Proposition 5.1:** A trajectory defined by  $x(t) = [V(t), \beta(t), r(t)]^T$ ,  $t \in [t_0, t_f]$ , can be recovered from the output in (22) if and only if  $y_1(t)$ ,  $y_2(t)$  and  $r(t)$  satisfy equation (24).

**Proof:** To show necessity, we need to show that for given  $x(t) = [V(t), \beta(t), r(t)]^T$  the flat outputs  $y_1(t)$  and  $y_2(t)$  satisfy the constraint equation (24). Recall from (22) that

$y_2 = V \sin \beta - (I_z / m \ell_f) r$ . Then the Lie derivative of  $y_2$  along the vector field in (1) is given by

$$\dot{y}_2 = \mathcal{L}_f y_2(x, u) = \frac{\ell_f + \ell_r}{m \ell_f} f_{Ry} - V r \cos \beta, \quad (31)$$

where, following equations (2) and (3),  $f_{Ry}$  is given by

$$f_{Ry} = -D_R \sin \left( C_{Ratan} \left( B_{Ratan} \left( \frac{V \sin \beta - \ell_r r}{V \cos \beta} \right) \right) \right). \quad (32)$$

Substituting (32) into (31), and replacing  $V \cos \beta$  and  $V \sin \beta$  with  $y_1$  and  $y_2 + I_z r / m \ell_f$ , respectively, we then obtain the equation

$$D_R \sin \left( C_{Ratan} \left( B_{Ratan} \left( \frac{(m \ell_f \ell_r - I_z) r - m \ell_f y_2}{m \ell_f y_1} \right) \right) \right) \frac{\ell_f + \ell_r}{m \ell_f} - \dot{y}_2 - y_1 r = 0. \quad (33)$$

Equation (24) is implied by (33).

To show sufficiency, we need to prove that for given flat outputs  $y_1(t)$ ,  $y_2(t)$ , one can solve for the trajectory  $x^*(t) = [V^*(t), \beta^*(t), r^*(t)]^T$  that generates  $y_1(t)$  and  $y_2(t)$ . This is, indeed, guaranteed by the property of differential flatness of the system. The implicit function theorem [34], [35] guarantees the local solvability of (24). The conditions are summarized in Theorem 5.2, from which one can recover the yaw rate  $r^*(t)$ . The states  $V^*(t)$  and  $\beta^*(t)$  are then calculated directly using the equations (23) in Section IV-B.

**Theorem 5.2:** Suppose there exists an open set  $\Omega \subseteq \mathbb{R}^4$  and a point  $P^* = (y_1^*, y_2^*, \dot{y}_2^*, r^*) \in \Omega$ , such that  $g(P^*) = 0$  and

$$\left. \frac{\partial g}{\partial r} \right|_{P^*} = D_R C_R B_R \frac{\cos \left( C_{Ratan} \left( B_{Ratan} \left( \Pi(y_1^*, y_2^*, r^*) \right) \right) \right) (m \ell_f \ell_r - I_z)}{\left( 1 + B_R^2 \tan^2 \left( \Pi(y_1^*, y_2^*, r^*) \right) \right) (1 + \Pi^2(y_1^*, y_2^*, r^*))} - \frac{m \ell_f y_1^*}{\ell_f + \ell_r} \neq 0, \quad (34)$$

where

$$\Pi(y_1^*, y_2^*, r^*) = \frac{(m \ell_f \ell_r - I_z) r^* - m \ell_f y_2^*}{m \ell_f y_1^*}, \quad (35)$$

then there exist some neighborhood  $\mathcal{N}(P^*) \subseteq \Omega$  about  $P^*$  and a continuous function  $G : \mathbb{R}^3 \mapsto \mathbb{R}$  such that  $r = G(y_1, y_2, \dot{y}_2)$  for all  $(y_1, y_2, \dot{y}_2, r) \in \mathcal{J}$ , where  $\mathcal{J} \triangleq \{(y_1, y_2, \dot{y}_2, r) \in \mathcal{N}(P^*) | g(y_1, y_2, \dot{y}_2, r) = 0\}$ .

For details about the implicit function theorem one can refer to [35]. Based on the idea in Proposition 5.1, a feasible trajectory for the vehicle from  $A$  to  $B$  that satisfies all boundary conditions in TABLE II can be designed as follows.

**1) Path design:** We generate a smooth path connecting points  $A$  and  $B$  in Fig. 7, which is denoted as  $\overline{AB}$ . The tangent lines to  $\overline{AB}$  at  $A$  and  $B$  must be parallel with the direction of the vehicle's velocity (see Fig. 8). We use a cubic Bézier curve to generate  $\overline{AB}$  [36], which is constructed using four control points, namely,  $P_0, \dots, P_3$ , and is represented by

$$\gamma(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3, \quad t \in [0, 1], \quad (36)$$

where  $P_0 = A$ ,  $P_3 = B$ , and the points  $P_1, P_2$  need to be

selected. In this paper the optimal choices for the control points  $P_1, P_2$  are determined by minimizing the jerk energy of the Bézier curve  $\overline{AB}$ , namely, we want to solve the following optimization problem,

$$P_1^*, P_2^* = \arg \min_{P_1, P_2} E(\gamma) = \int_0^1 \|\gamma'''(P_1, P_2, t)\|^2 dt, \quad (37)$$

where  $\|\cdot\|$  denotes the squared 2-norm. By substituting (36) into the jerk energy functional  $E(\gamma)$  in (37), one obtains the following equation,

$$E(\gamma) = 36\| -P_0 + 3P_1 - 3P_2 + P_3 \|^2. \quad (38)$$

We recall that a Bézier curve lies inside the convex hull

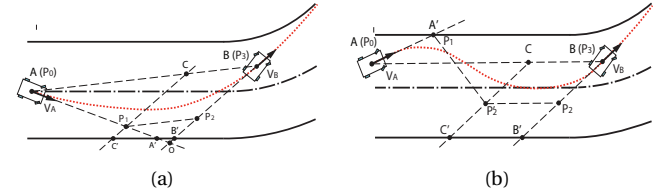


Fig. 8: Path planning for guiding control.

defined by its control points. The path  $\overline{AB}$  must satisfy the road boundary constraints if the control points  $P_1$  and  $P_2$  are inside the lane. Hence, if one can find two points  $P_1, P_2$  in the interior of the lane such that  $\overline{P_1 P_2} \parallel \overline{P_0 P_3}$  and the length is  $\|P_1 P_2\| = \|P_0 P_3\|/3$  (see Fig. 8(a)), then such  $P_1, P_2$  are the optimal solution and the optimal jerk energy is  $E^*(\gamma) = 0$ . Nevertheless, in some cases the optimal jerk energy  $E^*(\gamma)$  cannot be zero and the corresponding optimal control points  $P_1^*$  and  $P_2^*$  may not be obtained inside the lane. For instance, in Fig. 8(b) the optimal control point  $P_1^*$  is selected on the road boundary in order to minimize the jerk energy, which is given by  $E^*(\gamma) = 324\|P_1 P_2'\|^2$  in this case. We summarize the path planning algorithm in Algorithm 2 below.

#### Algorithm 2 Path Generation Using Cubic Bézier Curves

**Input:**  $A, B, V_A, V_B$

**Output:**  $P_1^*, P_2^*, \gamma(\tau)$

- 1:  $P_0 \leftarrow A, P_3 \leftarrow B$
- 2: Determine  $C$  on segment  $\overline{AB}$  such that  $\|BC\| = \|\overline{AB}\|/3$
- 3: Determine  $B'$  on the outer boundary such that  $\|BB'\| \parallel V_B$
- 4: Determine  $C'$  on the outer boundary such that  $\|CC'\| \parallel \overline{BB'}$
- 5: Determine  $A'$  on the inner/outer boundary such that  $\|AA'\| \parallel V_A$
- 6: **if**  $A'$  is on the outer boundary (case Fig. 8(a)) **then**
- 7:   **if**  $AA'$  intersects  $CC'$  (which is the ideal case:  $E^*(\gamma) = 0$ ) **then**
- 8:      $P_1^* \leftarrow$  intersection of  $AA'$  and  $CC'$
- 9:     Determine  $P_2^*$  on  $\overline{BB'}$  such that  $\overline{P_1^* P_2^*} \parallel \overline{AB}$
- 10: **else**
- 11:    $P_1^* \leftarrow A'$
- 12:   Determine  $P_2^*$  on  $\overline{BB'}$  such that  $\overline{C' P_2^*} \parallel \overline{AB}$
- 13: **else** (case Fig. 8(b))
- 14:    $P_1^*, P_2' \leftarrow \arg \min_{P_1 \in \overline{AA'}, P_2' \in \overline{CC'}} \|P_1 P_2'\|$  (see [37])
- 15:   Determine  $P_2^*$  on  $\overline{BB'}$  such that  $\overline{P_2' P_2^*} \parallel \overline{AB}$
- 16:  $\gamma(\tau) = (1-\tau)^3 P_0 + 3(1-\tau)^2 \tau P_1^* + 3(1-\tau) \tau^2 P_2^* + \tau^3 P_3, \quad \tau \in [0, 1]$

**2) Speed profile design:** In order to achieve maximum

speed, this work refers to [23] to design the speed profile along the trajectory generated using Algorithm 2, with the extension that we use a single track model instead of a single mass model as in [23]. This allows to take into account the longitudinal load transfer arising from the accelerating/braking of the vehicle.

We formulate an optimal control problem to find the optimal speed profile. Let  $\hat{f}_{ij}$  ( $i = F, R$  and  $j = x, y$ ) denote the longitudinal and lateral friction forces at the front and rear wheels represented in the frame  $X_V - C.M. - Y_V$  (fixed on the chassis, see Fig. 1). The components  $\hat{f}_{ij}$  should not be confused with the components  $f_{ij}$  in (1a)-(1c). In the case of the vehicle having rear wheel drive of a differential type, we assume that: a) the lateral tire force of the front wheel  $\hat{f}_{Fy}$  is always able to balance the moment arising from the lateral tire force of the rear wheel  $\hat{f}_{Ry}$ ; and b) the time derivative of the yaw rate along the trajectory is small, namely,  $\dot{r}(s) \approx 0$ , where  $s$  denotes the length of the path. These assumptions mean that the vehicle is able to keep lateral stability and the vehicle's yaw rate changes slowly, which should be reasonable if the planned path is smooth with respect to certain curvature constraint. Hence, one can approximate the front wheel tire forces using the rear wheel tire forces,

$$\hat{f}_{Fy} = \hat{f}_{Ry} \ell_r / \ell_f, \quad (39a)$$

$$\hat{f}_{Fx} = -\hat{f}_{Fy} \tan(\delta - \alpha_F) = -\hat{f}_{Ry} \ell_r (\ell_f + \ell_r) / \ell_f R(s), \quad (39b)$$

where  $R(s)$  denotes the radius. We use  $\bar{\mu}$  to denote the peak friction coefficient and assume that the velocity  $V(s)$  satisfies  $V^2(s) - \bar{\mu}gR(s) \leq 0$ . Then the limit of the acceleration  $a$  can be determined by solving (39a)-(39b) and the equations (40a)-(40d),

$$\hat{f}_{Ry} = mV^2(s)\ell_f / (\ell_f + \ell_r)R(s), \quad (40a)$$

$$ma = \hat{f}_{Rx} + \hat{f}_{Fx}, \quad (40b)$$

$$\bar{\mu}f_{Rz} = \sqrt{\hat{f}_{Ry}^2 + \hat{f}_{Rx}^2}, \quad (40c)$$

$$f_{Rz}(\ell_f + \ell_r) = mah + mg\ell_f, \quad (40d)$$

The result is given by

$$a_{\min} = -\frac{\mathcal{B} - \sqrt{\Delta}}{2\mathcal{A}}, \quad a_{\max} = -\frac{\mathcal{B} + \sqrt{\Delta}}{2\mathcal{A}}, \quad (41)$$

where  $\mathcal{A} = \bar{\mu}^2 h^2 / (\ell_f + \ell_r)^2 - 1$ ,  $\mathcal{B} = 2\bar{\mu}^2 gh\ell_f / (\ell_f + \ell_r)^2 - 2V^2\ell_r / R^2$ , and  $\Delta = \mathcal{B}^2 - 4\mathcal{A}\mathcal{C}$ , and where  $\mathcal{C} = \bar{\mu}^2 g^2 \ell_f^2 / (\ell_f - \ell_r)^2 - V^4 \ell_r^2 / R^4 - V^4 \ell_f^2 / (\ell_f - \ell_r)^2 R^2$ . If we define the new states  $z_1 \triangleq s$ ,  $z_2 \triangleq ds/dt = V$ , the state equations may be written as

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = -\frac{\mathcal{B}(z_1, z_2) + u\sqrt{\Delta(z_1, z_2)}}{2\mathcal{A}}, \quad u \in [-1, 1]. \quad (42)$$

Next, we derive the feasibility condition for (42). Since it is not possible to speed up the vehicle when the total tire force of the driving wheel reaches its peak value, it follows that  $m\dot{z}_2 = \hat{f}_{Rx} + \hat{f}_{Fx} = 0$ , while (40c) holds. The feasibility

region for (42) is given by

$$S \triangleq \{(z_1, z_2) : C_0(z_1, z_2) \triangleq z_2^2 - \bar{\mu}gR(z_1) / \sqrt{1 + \ell_r^2(\ell_f + \ell_r)^2 / \ell_f^2 R^2(z_1)} \leq 0\}. \quad (43)$$

We want to determine the optimal control  $u$  that drives (42) along a given trajectory in minimum time  $t_f$  subject to (43). Mathematically, we want to solve the optimization problem

$$\min_u t_f, \quad (44)$$

subject to (43) and the following boundary conditions

$$z_1(t_0) = z_{10}, \quad z_1(t_f) = z_{1f}, \quad z_2(t_0) = z_{20}, \quad z_2(t_f) = z_{2f}.$$

The solution to this problem is given in the Appendix.

Proposition A.3 (see the Appendix) indicates that for a given path from  $A$  to  $B$  with constant or monotonically decreasing/increasing curvature, one can construct the speed profile by forward integration of (42) from  $(z_1^A, z_2^A)$  with  $u = 1$  ( $a = a_{\max}$ ) and by backward integration of (42) from  $(z_1^B, z_2^B)$  with  $u = -1$  ( $a = a_{\min}$ ). 3) **Yaw motion design:** According to Proposition 5.1, the yaw motion of the vehicle  $\psi(t)$  must be carefully designed with respect to the following boundary conditions and the nonlinear constraint in (45),

$$\psi(t_0) = \psi_A, \quad \psi(t_f) = \psi_B, \quad \dot{\psi}(t_0) = r_A, \quad \dot{\psi}(t_f) = r^{ss}, \quad g(\dot{\psi}(t)) = 0, \quad (45)$$

where the constraint function  $g$  was defined in (24). A feasible solution for (45) and (24) is found numerically.

4) **Feasibility analysis:** The trajectory designed following equations (36)-(45) does not take into account the steering range of the front wheel, and hence the trajectory is required to satisfy the following condition,

$$\min_{\delta \in [\underline{\delta}, \bar{\delta}]} f_{Fy}(\delta) \cos(\delta) \leq \Lambda(t) \leq \max_{\delta \in [\underline{\delta}, \bar{\delta}]} f_{Fy}(\delta) \cos(\delta), \quad (46)$$

where  $\underline{\delta}$  and  $\bar{\delta}$  denote the allowed minimal and maximal steering angles of the front wheel. The term  $\Lambda(t)$  in (46) has the following expression,

$$\Lambda = \frac{m y_1 r \ell_r + m \dot{y}_2 \ell_r - I_z \dot{r}(1 + \ell_r / \ell_f)}{\ell_f + \ell_r}. \quad (47)$$

The inequality (46) provides the necessary condition that is satisfied by a feasible trajectory. One can design a path  $\gamma(\tau)$  following Algorithm 2, and then one can design  $V(t)$  and  $r(t)$  subject to the constraint in (46).

## VI. CONTROL DESIGN

A switching control is designed to achieve the tasks for the three different stages. In the guiding stage, a tracking controller is designed such that the vehicle is driven to reach the desired steady-state at the entry of the corner. Afterwards, a stabilizing controller is designed such that the vehicle follows the steady-state along the desired path. In the exiting stage, the vehicle leaves the corner and switches to a new control mode depending on the specific task. In this paper, we design a state feedback controller that aligns the posture of the vehicle to be parallel with the road in the exiting stage.

## A. Tracking Controller

We denote the desired flat output as  $y^d(t)$  and denote the current output as  $y(t)$ . The tracking error is  $e(t) = y(t) - y^d(t) \triangleq [e_1, e_2]^T$  and the accumulated tracking error is  $\zeta(t) = \int_0^t e(\tau) d\tau + \zeta_0 \triangleq [\zeta_1, \zeta_2]^T$ .

The following design drives  $\zeta(t)$  and  $e(t) \rightarrow 0$  as  $t \rightarrow \infty$  with proper choices of  $\lambda$  and  $\nu$ ,

$$\dot{e}_1 = -\lambda_1 e_1 - \lambda_2 \zeta_1, \quad (48a)$$

$$\dot{e}_2 = -\nu_1 \dot{e}_2 - \nu_2 e_2 - \nu_3 \zeta_2. \quad (48b)$$

To design  $\lambda$  and  $\nu$ , we further let  $\mathcal{E} = [\zeta_1, e_1, \zeta_2, e_2, \dot{e}_2]^T$ . Then the dynamics of the error  $\mathcal{E}$  is given by  $\dot{\mathcal{E}} = A\mathcal{E}$ , where  $A \in \mathbb{R}^{5 \times 5}$  is the error system matrix. If we denote the set of the eigenvalues of  $A$  as  $\text{mspec}(A) = \{s_1, s_2, s_3, s_4, s_5\}$ , then it can be shown that

$$\lambda_1 = -s_1 - s_2, \quad (49a)$$

$$\lambda_2 = s_1 s_2, \quad (49b)$$

$$\nu_1 = s_3 s_4 + s_3 s_5 + s_4 s_5, \quad (49c)$$

$$\nu_2 = -s_3 - s_4 - s_5, \quad (49d)$$

$$\nu_3 = -s_3 s_4 s_5. \quad (49e)$$

Hence  $\lambda$  and  $\nu$  can be determined by assigning the eigenvalues  $s_1, s_2, \dots, s_5$  with appropriate negative real parts. It then follows from (48) that

$$\dot{y}_1 = \dot{y}_1^d - \lambda_1 e_1 - \lambda_2 \zeta_1, \quad (50a)$$

$$\dot{y}_2 = \dot{y}_2^d - \nu_1 \dot{e}_2 - \nu_2 e_2 - \nu_3 \zeta_2, \quad (50b)$$

and the control  $u$  is obtained by solving the following equations:

$$\mathcal{L}_f y_1(x, u) = \dot{y}_1, \quad (51a)$$

$$\mathcal{L}_f^2 y_2(x, u) = \dot{y}_2, \quad (51b)$$

where  $\mathcal{L}_f y_1(x, u)$  and  $\mathcal{L}_f^2 y_2(x, u)$  are given by

$$\mathcal{L}_f y_1(x, u) = -\frac{\sin \delta}{m} f_{Fy} + \frac{\cos \delta}{m} f_{Fx} + \frac{1}{m} f_{Rx} + Vr \sin \beta, \quad (52a)$$

$$\begin{aligned} \mathcal{L}_f^2 y_2(x, u) = & \left( \frac{r}{m} \sin \delta - \frac{\ell_f V}{I_z} \cos \delta \cos \beta \right) f_{Fy} + \frac{\ell_r V}{I_z} \cos \beta f_{Rx} \\ & - \frac{r}{m} f_{Rx} - Vr^2 \sin \beta + \frac{\ell_f + \ell_r}{m \ell_f} \frac{\partial f_{Ry}}{\partial \alpha_R} \left( \frac{\partial \alpha_R}{\partial V} \dot{V} + \frac{\partial \alpha_R}{\partial \beta} \dot{\beta} + \frac{\partial \alpha_R}{\partial r} \dot{r} \right) \\ & - \left( \frac{r}{m} \cos \delta + \frac{\ell_f V}{I_z} \sin \delta \sin \beta \right) f_{Fx}, \end{aligned} \quad (52b)$$

and where

$$\frac{\partial f_{Ry}}{\partial \alpha_R} = \frac{D_R \cos(C_R \text{atan}(B_R \alpha_R)) C_R B_R}{1 + (B_R \alpha_R)^2}, \quad (53a)$$

$$\frac{\partial \alpha_R}{\partial V} = -\frac{\ell_r r \cos \beta}{V^2 \cos^2 \beta + (V \sin \beta - \ell_r r)^2}, \quad (53b)$$

$$\frac{\partial \alpha_R}{\partial \beta} = -\frac{V^2 - V \ell_r r \sin \beta}{V^2 \cos^2 \beta + (V \sin \beta - \ell_r r)^2}, \quad (53c)$$

$$\frac{\partial \alpha_R}{\partial r} = \frac{V \ell_r \cos \beta}{V^2 \cos^2 \beta + (V \sin \beta - \ell_r r)^2}. \quad (53d)$$

The (local) solvability of equations in (51) is guaranteed by the implicit function theorem if and only if the following

conditions are satisfied

$$\left( I_z V \sin \beta (\partial f_{Ry} / \partial V) + I_z \cos \beta (\partial f_{Ry} / \partial \beta) + m V \ell_f (\partial f_{Ry} / \partial r) \right) / m^2 \ell_f^2 - V^2 \cos \beta / (\ell_f + \ell_r) \neq 0, \quad (54)$$

$$f_{Fy} \sin \delta - (\partial f_{Fy} / \partial \delta) \cos \delta \neq 0. \quad (55)$$

where  $\partial f_{Ry} / \partial V$ ,  $\partial f_{Ry} / \partial \beta$  and  $\partial f_{Ry} / \partial r$  are calculated following (53) using the chain rule, and  $\partial f_{Fy} / \partial \delta$  is given by

$$\frac{\partial f_{Fy}}{\partial \delta} = \frac{D_F \cos(C_F \text{atan}(B_F \alpha_F)) C_F B_F}{1 + (B_F \alpha_F)^2}. \quad (56)$$

One can refer to [25] for more analysis on the solvability of (51). The scheme of the controller design can be shown in Fig. 9.

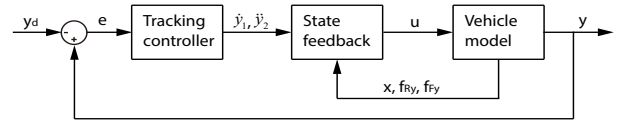


Fig. 9: Scheme of flatness-based vehicle dynamics control.

## B. Cornering Controller

After the vehicle reaches position  $B$  in Fig. 1, the controller switches from the tracking control mode to the stabilizing control mode. To this end, we linearize the vehicle model about the steady state  $x^{ss} = [V^{ss}, \beta^{ss}, r^{ss}]^T$  [9] and design the state feedback control using a linear-quadratic regulator (LQR).

The state feedback control law only tracks the desired state  $x^{ss} = [V^{ss}, \beta^{ss}, r^{ss}]^T$ , which does not take into account the deviation of the vehicle from the desired path. We therefore consider the following two methods to improve tracking performance: We calculate a series of steady states having different cornering radii  $R^{ss}$ . When the vehicle is driving too close to the inner (outer) boundary of the road, the controller will use a different steady state that has a larger (smaller) cornering radius. Alternatively, one can redesign the state feedback control using the augmented state vector that takes into account of the vehicle's position.

## C. Exiting Controller

After the vehicle exits steady-state cornering, we want to steer the vehicle along a straight line while exiting the corner. We consider the steady state for straight-line driving as  $\hat{x}^{ss} = [\hat{V}^{ss}, \hat{\beta}^{ss}, \hat{r}^{ss}]^T = [V^{ss}, 0, 0]^T$ , where  $V^{ss}$  takes the value of the exit velocity of the vehicle after the corner. In order to align the vehicle to be parallel with the road, we include a target yaw angle  $\psi^{ss}$  into the state vector, that is,  $x^{ss} = [V^{ss}, 0, 0, \psi^{ss}]^T$ . The corresponding control for steady-state straight-line driving is  $\hat{u}^{ss} = [\hat{\delta}^{ss}, \hat{f}_{Fx}^{ss}, \hat{f}_{Rx}^{ss}]^T = [0, 0, 0]^T$ . We then linearize the vehicle model about the new equilibrium  $\hat{x}^{ss}$  and  $\hat{u}^{ss}$ . In order to achieve  $\hat{x} \rightarrow 0$  as  $t \rightarrow \infty$ , we may use a state feedback control law  $\hat{u} = \hat{K} \hat{x}$ , where  $\hat{K}$  is taken such that the matrix  $\hat{A}^{ss} + \hat{B}^{ss} \hat{K}$  is Hurwitz (i.e., using LQR).

## VII. NUMERICAL SIMULATIONS

In this section we implement and validate the proposed control architecture in simulations and analyze the results.

### A. Trajectory Design

We first calculate the equilibria for the steady-state cornering using the vehicle model introduced in Section II. TABLE III shows the equilibria for steady state-cornering at different speeds.

TABLE III: Equilibrium for steady-state cornering.

|                    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9     |
|--------------------|------|------|------|------|------|------|------|------|-------|
| $V^{ss}$ [m/s]     | 5    | 8.5  | 9.4  | 10.8 | 11.9 | 12.1 | 13.9 | 15   | 6     |
| $\beta^{ss}$ [deg] | -2.8 | -30  | -20  | -20  | -28  | -20  | -23  | -32  | -14.3 |
| $r^{ss}$ [deg/s]   | 11.3 | 48.7 | 35.9 | 30.9 | 34.1 | 27.7 | 26.5 | 28   | 57.3  |
| $R^{ss}$ [m]       | 25.4 | 10   | 15   | 20   | 20   | 25   | 30   | 30.7 | 6     |

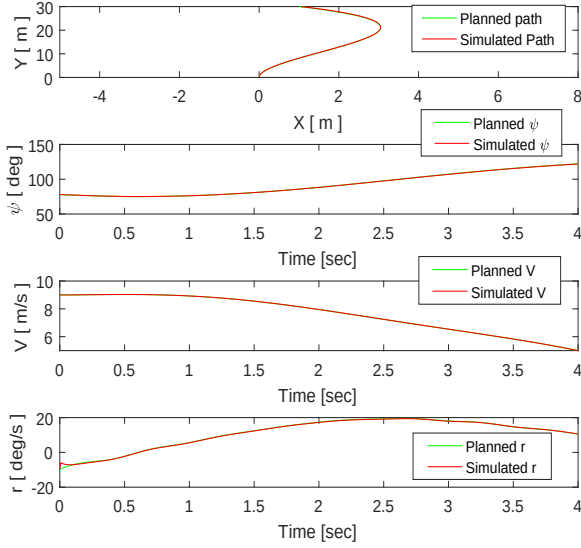


Fig. 10: The desired and simulated trajectories.

A feasible trajectory that includes the target steady-state cornering process is designed following (36)-(46). In order to demonstrate the proposed trajectory planning and controller design, we take the first equilibrium in TABLE III and use the segmentation in Fig. 3(a). For the boundary conditions and the designed time  $t$  given in TABLE IV, we plan the trajectory of the vehicle and plot it using green color in Fig. 10. The designed trajectory in Fig. 10 must satisfy the feasibility condition in (46).

TABLE IV: Boundary conditions

|            | $t$ [s] | $V$ [m/s] | $\beta$ [deg] | $r$ [deg/s] | $X$ [m] | $Y$ [m] | $\psi$ [deg] |
|------------|---------|-----------|---------------|-------------|---------|---------|--------------|
| Position A | 0       | 9         | 0             | -9.6        | 0       | 0       | 78           |
| Position B | 4       | 5         | -2.8          | 11.3        | 1       | 20      | 122          |

### B. Tracking Control

We assign the eigenvalues of the error system matrix  $A$  to be  $s_1 = s_2 = s_3 = -10$  and  $s_4 = s_5 = -5$  such that the matrix  $A$  is Hurwitz. It follows from (49) that the corresponding parameters for the tracking controller are

$$\lambda_1 = 20, \quad \lambda_2 = 100, \quad \nu_1 = 125, \quad \nu_2 = 20, \quad \nu_3 = 250. \quad (57)$$

We first tested the tracking controller using numerical simulations. The results in Fig. 10 show that the flatness-based tracking controller is able to track the trajectory designed following the procedure in (36)-(46).

We then implemented the LQR controller after the vehicle reaches position  $B$  and implemented the state feedback controller after the vehicle exits the corner. The complete trajectory is plotted in Fig. 11. The desired high-speed cornering maneuver of the vehicle can be achieved using the hybrid-mode control that switches between the flatness-based controller, the LQR controller and the state feedback controller for the exiting stage. It is worth-mentioning that the sideslip angle during cornering in Fig. 11 is not visible since in this example we used the equilibrium having  $\beta^{ss} = 2.8$  [deg]. Equilibria having larger sideslip angles are demonstrated next.

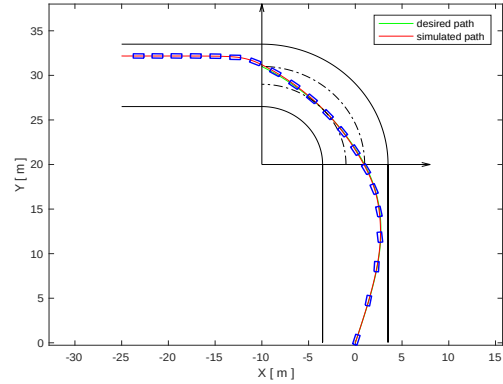


Fig. 11: Switching-mode control for steady-state cornering.

### C. Late-Apex High-Speed Cornering

In this section we generate high-speed cornering trajectories having “late apex” as shown in Fig. 7(b), following the similar control design procedures in Section VI. We used the second and the third equilibria in TABLE III for steady-state cornering to achieve a high-speed, high-slip sliding maneuver.

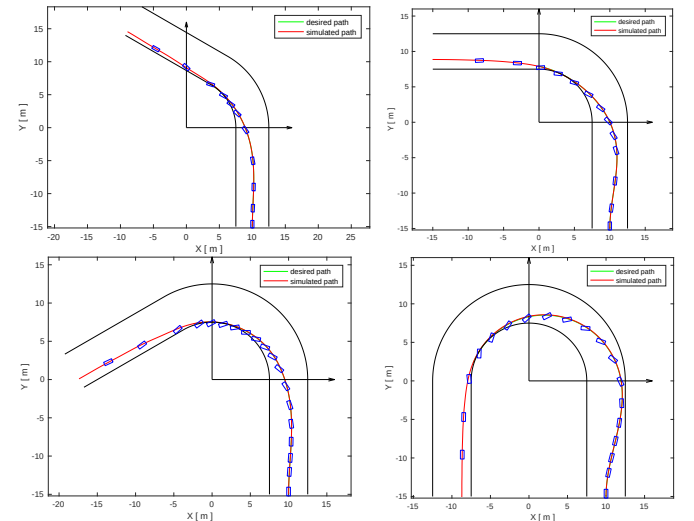


Fig. 12: High-speed cornering maneuver generation for different road geometries.

The road geometry was defined using the parameters in TABLE V. It is worth-mentioning that the other equilibria in TABLE III have steady-state cornering radii  $R^{ss} > 20$  [m], which seems too large and is not convenient to use for the road geometry in TABLE V, especially when the corner angle is larger than 90 deg. For high-speed cornering with different corner angles, TABLE V provides appropriate choices of the equilibria and the angles  $\angle 1, \dots, \angle 4$  that determine the geometry for different high-speed cornering trajectories.

Using these values we designed high-speed cornering trajectories for different corner angles and implement the proposed switch-mode controller accordingly. The planned and simulated trajectories are plotted in Fig. 12.

TABLE V: Road geometry and high-speed cornering trajectory setup.

| Road geometry     | $S_1$ [m]      | $S_2$ [m]          | $R_{ref}$ [m]    | $D$ [m]           |                  | Corner angle [deg] |                  |
|-------------------|----------------|--------------------|------------------|-------------------|------------------|--------------------|------------------|
|                   | 15             | 15                 | 10               | 5                 |                  | 60/90/120/180      |                  |
| Initial condition | $V$ [m/s]      | $\beta$ [deg]      | $r$ [deg/s]      | $(X, Y)$ [(m,m)]  |                  | $\psi$ [deg]       |                  |
|                   | 15             | 0                  | 0                | $(R_{ref}, -S_1)$ |                  | 90                 |                  |
|                   | $V^{ss}$ [m/s] | $\beta^{ss}$ [deg] | $r^{ss}$ [deg/s] | $\angle 1$ [deg]  | $\angle 2$ [deg] | $\angle 3$ [deg]   | $\angle 4$ [deg] |
| 60 deg            | 9.4            | -20                | 35.9             | 6                 | 30               | 18                 | 6                |
| 90 deg            | 9.4            | -20                | 35.9             | 9                 | 54               | 18                 | 9                |
| 120 deg           | 8.5            | -30                | 48.7             | 12                | 84               | 12                 | 12               |
| 180 deg           | 8.5            | -30                | 48.7             | 18                | 126              | 18                 | 18               |

Figure 12 shows the computed high-speed cornering trajectories for 60 deg, 90 deg, 120 deg and 180 deg cornering, respectively. In each figure, the simulated trajectory agrees well with the designed trajectory. By designing the angles  $\angle 1, \dots, \angle 4$ , one achieves high-speed cornering with different “late apex” features.

The design of the angles  $\angle 1, \dots, \angle 4$  must take into account the target equilibrium of the steady-state cornering. If the radius of the trajectory is very large (i.e., more than two times the corner radius), high-speed cornering with “late apex” will be hard to generate since the target path may cut the outer boundary of the road. One can use another equilibrium with a smaller steady-state cornering radius or change the angles  $\angle 1, \dots, \angle 4$  properly (i.e., reduce  $\angle 2$  to move the tangent point P farther from the exit of the corner). Recommended values of  $\angle 1, \dots, \angle 4$  are given in TABLE VI.

TABLE VI: “Late apex” specification.

| Corner angle | $\angle 1$  | $\angle 2$             | $\angle 3$             | $\angle 4$  |
|--------------|-------------|------------------------|------------------------|-------------|
| $\Omega$     | $0.1\Omega$ | $(0.5 \sim 0.7)\Omega$ | $(0.1 \sim 0.3)\Omega$ | $0.1\Omega$ |

## VIII. EXPERIMENTAL VALIDATION

In this section we generate high-speed cornering maneuvers by implementing the switch-mode controller on a closed track, using both the high-fidelity CarSim vehicle model and the Auto-Rally platform [38].

### A. CarSim Simulation

We first show the simulated high-speed cornering trajectories using the CarSim/Simulink software [39]. The vehicle model parameters are given by TABLE VII.

TABLE VII: CarSim vehicle model parameters.

|                           |       |                    |                           |       |                              |
|---------------------------|-------|--------------------|---------------------------|-------|------------------------------|
| $m$ [kg]                  | 850   | total mass         | $m^s$ [kg]                | 767.4 | sprung mass                  |
| $m_f$ [kg]                | 20.65 | front wheel mass   | $m_r$ [kg]                | 20.65 | rear wheel mass              |
| $w_f$ [m]                 | 1.481 | front track        | $w_r$ [m]                 | 1.481 | rear track                   |
| $L$ [m]                   | 2.4   | wheel base         | $R_w$ [m]                 | 0.311 | wheel radius                 |
| $I_z$ [kgm <sup>2</sup> ] | 1401  | rotational inertia | $I_f$ [m]                 | 1.5   | distance of CM to front axle |
| $h$ [m]                   | 0.5   | height of CM       | $I_w$ [kgm <sup>2</sup> ] | 0.6   | wheel rotational inertia     |

The geometry of the track is shown in Fig. 13, where the width of the track is 16 [m], and the inner and the outer corner radii are 5 [m] and 21 [m], respectively. The distance between the two corner centers is 60 [m]. The friction coefficient of the road is set to be 0.6.

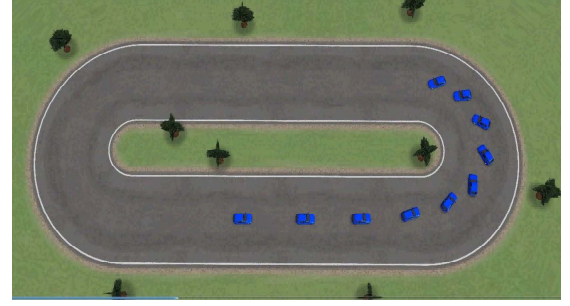


Fig. 13: Closed track in CarSim.

We calculated the steady state of the vehicle model using the parameters in TABLE VII. In this work, we generate high-speed cornering using the initial conditions and the steady state in TABLE VIII.

TABLE VIII: Boundary conditions

|                   | Speed [m/s] | Sideslip [deg] | Yaw rate [rad/s] |
|-------------------|-------------|----------------|------------------|
| Initial condition | 18          | 0              | 0                |
| Target SS         | 8.18        | -50            | 0.68             |

The switch-mode controller was implemented as follows: 1) Firstly, given the current state of the vehicle, the tracking controller calculates the entry point B of the corner based on the given map of the track. 2) Next, the tracking controller plans a smooth path connecting the vehicle's current position and point B following Algorithm 2. We implemented the tracking control for the next 0.2 seconds. The control commands (i.e., steering angle, throttle and braking wheel cylinder pressures) are computed at 100 Hz. 3) We repeat step 2 until the vehicle reaches the desired steady-state at the target position B. 4) We switch to the LQR controller after the vehicle enters the corner for SS sliding (100 Hz). 5) We switch to the exiting controller to align the vehicle with the straight road after the vehicle is about to leave the corner (100 Hz). 6) The vehicle speeds up after it exits the corner to the maximum allowable velocity. 7) We repeat steps 1-6 to finish a couple of circles along the track.

Figure 14 shows the velocity, sideslip angle and yaw rate of the vehicle for a complete circle on the track. The vehicle slides through the corners at the two ends in the desired steady state from  $t = 5.5$  [sec] to  $t = 7.5$  [sec] and from  $t = 17$  [sec] to  $t = 19$  [sec], respectively. The animation is available on the DCSL Youtube channel <sup>1</sup>.

<sup>1</sup><https://www.youtube.com/watch?v=VN9Tpr8D1tk>

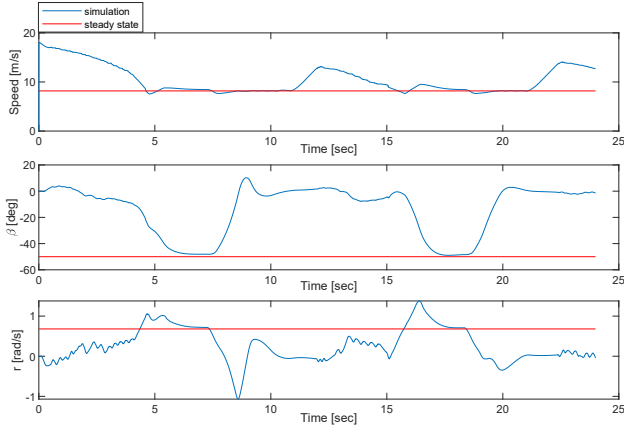


Fig. 14: Simulated trajectories.

### B. Auto-Rally Experiments

The simulation results using CarSim validates the control design by applying it on a high-fidelity vehicle model. Nevertheless, these simulations do not consider hardware performance and assume there is no computational delay, which is not realistic for real-time maneuvers. We therefore validated the control design on a fifth-scale Auto-Rally platform developed at Georgia Tech. The Auto-Rally platform and the test track are shown in Fig. 15.



Fig. 15: The test track and the auto-rally vehicle platform.

The Auto-Rally platform is driven by two rear wheels, and its top speed can reach up to 27 [m/s]. The size of the vehicle measures about 1[m]×0.6[m]×0.4[m]. The perimeter of the centerline of the test track is about 63 [m], and the width is about 3.4 [m]. A more detailed description of the vehicle model and the test track can be found in [38].

The vehicle and tire model parameters in TABLE IX are estimated using the approach in [40], [41] with the data collected from the field test.

TABLE IX: Vehicle/tire model parameters.

|                           |       |                    |              |       |                              |           |
|---------------------------|-------|--------------------|--------------|-------|------------------------------|-----------|
| $m$ [kg]                  | 21.5  | total mass         | $m^s$ [kg]   | 18.03 | sprung mass                  |           |
| $m_f$ [kg]                | 0.84  | front wheel mass   | $m_r$ [kg]   | 0.89  | rear wheel mass              |           |
| $w_f$ [m]                 | 0.44  | front track        | $w_r$ [m]    | 0.46  | rear track                   |           |
| $L$ [m]                   | 0.57  | wheel base         | $R$ [m]      | 0.095 | wheel radius                 |           |
| $I_z$ [kgm <sup>2</sup> ] | 1.02  | rotational inertia | $\ell_f$ [m] | 0.46  | distance of CM to front axle |           |
|                           | $B_F$ | $C_F$              | $D_F$ [N]    | $B_R$ | $C_R$                        | $D_R$ [N] |
|                           | 4.03  | 0.02               | 1192.16      | 2.45  | 0.02                         | 5414.88   |

Figure 16 shows the estimated tire force calculated using the expression in (2) with the identified tire parameters from TABLE IX. The lateral tire force  $f_{iy}$  ( $i = F, R$ ) can be approximated using (2) with satisfactory accuracy.

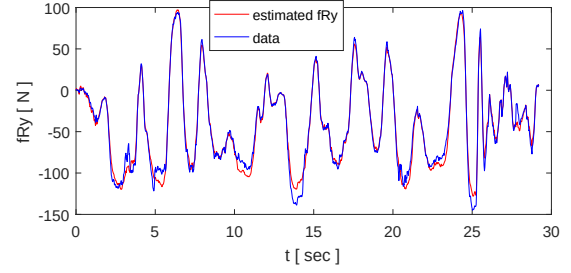


Fig. 16: The estimated lateral tire force.

We developed the controller using the ROS software framework. The controller serves as a ROS node and publishes a throttle command and a steering command at 50 Hz. In order to validate the control design before running a test on the real track, we implemented the controller on the Gazebo simulator (see [38]). The trajectory is shown in Fig. 17. The target steady state is given by the last column of TABLE III.

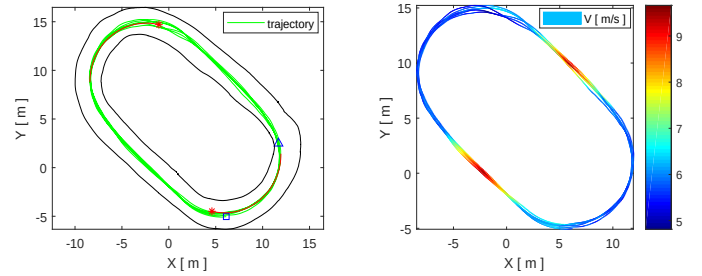


Fig. 17: The trajectories of Auto-Rally (counter clockwise).

The start and end points of the trajectory are marked with a triangle and a square signs, respectively. The red stars indicate the target points where the vehicle starts steady state cornering (point B in Fig. 7), and the red arcs indicate the desired steady state cornering trajectories. We measure the boundaries of the track and show them in black curves. The result shows that the flatness-based tracking controller is able to steer the vehicle into the corner with the desired speed, and the fixed-radius stabilizing controller is able to maintain the high sideslip sliding process of the vehicle following the desired trajectories. The speed of the vehicle is shown in Fig. 17 using a color map. The minimum speed is about 6 [m/s] in the corner and the maximum speed is about 9.6 [m/s] near the midpoint of the straight road segments.

Next, we implemented the controller using the Auto-Rally platform on the real track. Figure 18 shows the speed profile of the vehicle for a couple of rounds in the tests. The maximum speed is 9.7 [m/s], which is similar to the highest speed achieved in previous work [42]. The online path re-planning process during the guiding stage is demonstrated by the green curves in Fig. 19, where a new Bézier curve is generated every 0.1 seconds. We also compared the speed profile generated using the single track model and the speed

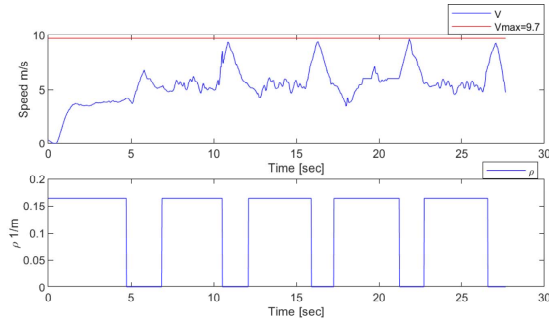


Fig. 18: The speed profile of Auto-Rally.

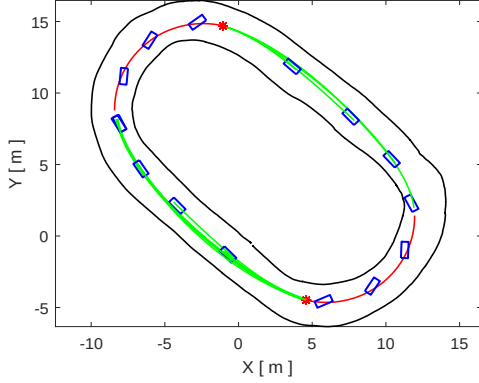


Fig. 19: Online path replanning.

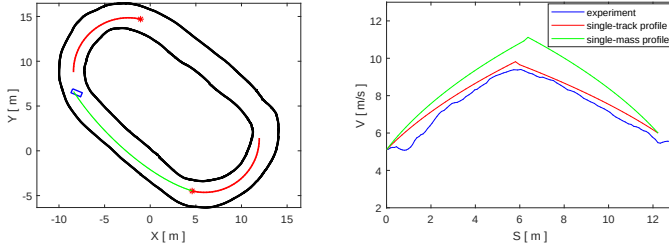


Fig. 20: Online speed profile generating.

profile generated using the single mass model [23]. We show the result in Fig. 20 (right), which corresponds to the green trajectory in Fig. 20 (left). The maximum tire friction coefficient in the experiments is about 0.8. One sees from Fig. 20 that the speed profile generated using the single track model agrees better with the experimental result than the speed profile generated using the single mass model. One also notices that the speed profile generated using the single mass model is symmetric about the axis through the peak. The reason is that the single mass model does not take into account the longitudinal load transfer arising from the accelerating/braking operations, and assumes that the maximum longitudinal tire forces for accelerating and braking are equal in magnitude. Moreover, the single mass model uses all wheels for the longitudinal control of the vehicle, which is not true for the Auto-Rally that has a rear wheel drive differential type, and consequentially, overestimates the maximum speed the vehicle is able to achieve.

Figure 21 shows a typical trajectory of the Auto-Rally

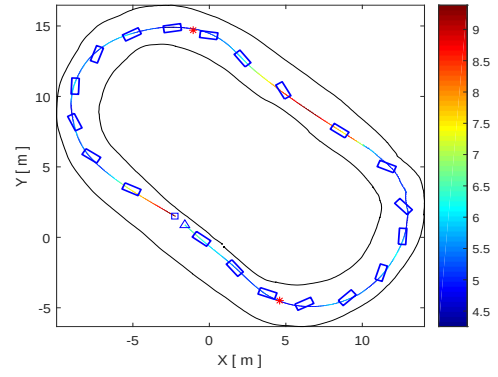


Fig. 21: The posture of Auto-Rally in a typical round.

vehicle in a complete round of experiment, where the vehicle enters the corner from the desired position (red stars) and slides through the corner at high speed (5.5-6.5 [m/s]) and high sideslip (10-20 deg). The video demonstrating this high-speed cornering maneuver is available on the DCSL Youtube channel<sup>2</sup>.

## IX. CONCLUSION

This paper provides a methodology to generate high-speed cornering maneuvers similarly to those used by expert rally racing drivers. We first use a trajectory learning technique to find a primitive trajectory that captures the salient features of a high-speed cornering maneuver, using a series of high-speed cornering demonstrations. Based on this primitive trajectory, we then approximately divide the high-speed cornering trajectory into three stages, namely, the entry, sliding and exiting stages, and we show that the middle stage includes a segment of steady-state cornering. We then develop a switching-mode control for each stage using different control techniques to generate high-speed cornering.

Based on the differential flatness property of the vehicle equations, we provide a method to plan the trajectory and design a tracking control. By changing the position of the tangential point of the vehicle's trajectory and the inner boundary of the road, we can successfully generate 60 deg, 90 deg, 120 deg and 180 deg high-speed cornering maneuvers at will with different “late apex” features. We demonstrated high-speed cornering maneuvers by implementing the switch-mode controller both in high-fidelity simulations and also using a scaled experimental platform. Since our approach plans the nominal trajectory based on geometry, the solution is given in a semi-analytic form, so there is no need to solve an optimal control problem on-the-fly. As a result, the computation effort is sufficiently low to allow for real-time high-speed cornering maneuver generation.

**Acknowledgement:** Support for this work has been provided from NSF award CPS-1544814 and from Ford Motor Company.

<sup>2</sup><https://youtu.be/3vJDdW6BEiY>

# REFERENCES

- [1] K. Yi, T. Chung, J. Kim, and S. Yi, "An investigation into differential braking strategies for vehicle stability control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 217, no. 12, pp. 1081–1093, 2003.
- [2] S. Di Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1236–1248, 2013.
- [3] L. De Novellis, A. Sorniotti, P. Gruber, and A. Pennycott, "Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3612–3623, 2014.
- [4] L. De Novellis, A. Sorniotti, P. Gruber, L. Shead, V. Ivanov, and K. Hoepfing, "Torque vectoring for electric vehicles with individually controlled motors: state-of-the-art and future developments," in *26th Electric Vehicle Symposium*, Los Angeles, CA, May 6–9 2012.
- [5] J. Ackermann, T. Bunte, and D. Odenthal, "Advantages of active steering for vehicle dynamics control," in *Proceedings of the 32nd International Symposium on Automotive Technology and Automation*, Vienna, Austria, June 14–18 1999, pp. 263–270.
- [6] 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.
- [7] Y. A. Ghoneim, W. C. Lin, D. M. Sidlosky, H. H. Chen, and Y.-K. Chin, "Integrated chassis control system to enhance vehicle stability," *International Journal of Vehicle Design*, vol. 23, no. 1–2, pp. 124–144, 2000.
- [8] Y. Kou, "Development and evaluation of integrated chassis control systems," Ph.D. dissertation, The University of Michigan, 2010.
- [9] E. Velenis, E. Frazzoli, and P. Tsiotras, "Steady-state cornering equilibria and stabilisation for a vehicle during extreme operating conditions," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2–4, pp. 217–241, 2010.
- [10] I. Chakraborty, P. Tsiotras, and J. Lu, "Vehicle posture control through aggressive maneuvering for mitigation of t-bone collisions," in *50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, December 12–15 2011, pp. 3264–3269.
- [11] A. R. Hauber, "The social psychology of driving behaviour and the traffic environment: Research on aggressive behaviour in traffic," *Applied Psychology*, vol. 29, no. 4, pp. 461–474, 1980.
- [12] L. Mizell, M. Joint, and D. Connell, "Aggressive driving: Three studies," *AAA Foundation for Traffic Safety*, pp. 1–13, 1997.
- [13] D. Shinar, "Aggressive driving: the contribution of the drivers and the situation," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 1, no. 2, pp. 137–160, 1998.
- [14] L. Tasca, *A review of the literature on aggressive driving research*. Ontario, Canada: Ontario Advisory Group on Safe Driving Secretariat, Road User Safety Branch, Ontario Ministry of Transportation, 2000.
- [15] M. Abou-Zeid, I. Kaysi, and H. Al-Naghi, "Measuring aggressive driving behavior using a driving simulator: An exploratory study," in *Third International Conference on Road Safety and Simulation*, Indianapolis, September 14–16 2011.
- [16] E. Velenis, P. Tsiotras, and J. Lu, "Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn," in *European Control Conference*, Kos, Greece, July 2–5 2007, pp. 1233–1240.
- [17] —, "Trail-braking driver input parameterization for general corner geometry," SAE Technical Paper, Tech. Rep., 2008.
- [18] —, "Optimality properties and driver input parameterization for trail-braking cornering," *European Journal of Control*, vol. 14, no. 4, pp. 308–320, 2008.
- [19] D. Tavernini, M. Massaro, E. Velenis, D. I. Katzourakis, and R. Lot, "Minimum time cornering: the effect of road surface and car transmission layout," *Vehicle System Dynamics*, vol. 51, no. 10, pp. 1533–1547, 2013.
- [20] R. Y. Hindiyeh and J. C. Gerdes, "Equilibrium analysis of drifting vehicles for control design," in *ASME 2009 Dynamic Systems and Control Conference*. Hollywood, CA: American Society of Mechanical Engineers, October 12–14 2009, pp. 181–188.
- [21] J. Yi and E. H. Tseng, "Nonlinear stability analysis of vehicle lateral motion with a hybrid physical/dynamic tire/road friction model," in *ASME Dynamic Systems and Control Conference*. Hollywood, CA: American Society of Mechanical Engineers, October 12–14 2009, pp. 509–516.
- [22] C. You and P. Tsiotras, "Real-time trail-braking maneuver generation for off-road vehicle racing," in *American Control Conference*, 2018, pp. 4751–4756.
- [23] E. Velenis and P. Tsiotras, "Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding-horizon implementation," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 275–296, 2008.
- [24] E. Velenis, "Analysis and control of high-speed wheeled vehicles," Ph.D. dissertation, Georgia Institute of Technology, 2006.
- [25] S. Fuchshumer, K. Schlacher, and T. Rittenschober, "Nonlinear vehicle dynamics control—a flatness based approach," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 15 2005, pp. 6492–6497.
- [26] M. S. Burhaumudin, P. M. Samin, H. Jamaluddin, R. A. Rahman, S. Sulaiman, et al., "Integration of magic formula tire model with vehicle handling model," *International Journal of Research in Engineering and Technology*, vol. 1, no. 3, pp. 139–145, 2012.
- [27] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software*, vol. 41, no. 1, p. 1, 2014.
- [28] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, July 5–9 2008, pp. 144–151.
- [29] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Differential flatness and defect: an overview," in *Workshop on Geometry in Nonlinear Control*, Banach Center Publications, Warsaw, 1993.
- [30] —, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [31] Y. Wang, S. Shi, and L. Li, "Flatness-based vehicle coupled control for steering stability and path tracking," in *Proceedings of SAE-China Congress 2015: Selected Papers*. Springer, 2016, pp. 49–60.
- [32] H. Souilem and N. Derbel, "Vehicle control by flatness," *Proceedings Engineering & Technology-Vol*, vol. 4, pp. 37–42, 2013.
- [33] J. Villagra, B. d'Andrea Novel, H. Mounier, and M. Pengov, "Flatness-based vehicle steering control strategy with sdre feedback gains tuned via a sensitivity approach," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 554–565, 2007.
- [34] G. M. Scarpello and D. Ritelli, "A historical outline of the theorem of implicit functions," *Divulgaciones Matemáticas*, vol. 10, no. 2, pp. 171–180, 2002.
- [35] S. G. Krantz and H. R. Parks, *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2012.
- [36] J.-w. Choi, R. Curry, and G. Elkaim, "Path planning based on Bézier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science*. San Francisco, CA: IEEE, October 22–24 2008, pp. 158–166.
- [37] P. Bourke. (1988) Points, lines, and planes. [Online]. Available: <http://paulbourke.net/geometry/pointlineplane/>
- [38] B. Goldfain, P. Drews, C. You, M. Barulic, O. Velev, P. Tsiotras, and J. M. Rehg, "Autorially: An open platform for aggressive autonomous driving," *arXiv preprint arXiv:1806.00678*, 2018.
- [39] Mechanical Simulation Corporation, 709 W. Huron, Ann Arbor, MI 48103, Jan. 2000, this manual describes the CarSim Educational software.
- [40] C. You, J. Lu, and P. Tsiotras, "Driver parameter estimation using joint E-/UKF and dual E-/UKF under nonlinear state inequality constraints," in *IEEE International Conference on Systems, Man, and Cybernetics*, Budapest, Hungary, August 24–29, 2016.
- [41] C. You and P. Tsiotras, "Vehicle modeling and parameter estimation using adaptive limited memory joint-state UKF," in *American Control Conference*, Seattle, May 24–26 2017, pp. 322–327.
- [42] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *International Conference on Robotics and Automation*, Stockholm, Sweden, 16–21 May 2016, pp. 1433–1440.

APPENDIX A  
OPTIMAL SPEED PROFILE

We recall that the optimization problem (44) is given by

$$\min_u t_f,$$

subject to (43) and the following boundary conditions

$$z_1(t_0) = z_{10}, \quad z_1(t_f) = z_{1f}, \quad z_2(t_0) = z_{20}, \quad z_2(t_f) = z_{2f}.$$

The Hamiltonian for this problem is given by

$$H(z, \kappa, u) = 1 + \kappa_1 z_2 - \kappa_2 \frac{\mathcal{B}(z_1, z_2) + u\sqrt{\Delta(z_1, z_2)}}{2\mathcal{A}} + \kappa_3 C_0(z_1, z_2). \quad (\text{A.1})$$

The dynamics equations for the adjoint states are  $\dot{\kappa}_1 = -\partial H/\partial z_1$  and  $\dot{\kappa}_2 = -\partial H/\partial z_2$ , which are given by

$$\begin{aligned} \dot{\kappa}_1 = & \frac{\kappa_2}{2\mathcal{A}} \frac{\partial B(z_1, z_2)}{\partial R(z_1)} R'(z_1) + \frac{\kappa_2 u}{4\mathcal{A}\sqrt{\Delta(z_1, z_2)}} \frac{\partial \Delta(z_1, z_2)}{\partial R(z_1)} R'(z_1) \\ & + \kappa_3 \bar{\mu} g \frac{1 + 2\ell_r^2(\ell_f + \ell_r)^2/\ell_f^2 R^2(z_1)}{(1 + \ell_r^2(\ell_f + \ell_r)^2/\ell_f^2 R^2(z_1))^{3/2}} R'(z_1), \end{aligned} \quad (\text{A.2a})$$

$$\begin{aligned} \dot{\kappa}_2 = & -\kappa_1 + \frac{\kappa_2}{2\mathcal{A}} \frac{\partial B(z_1, z_2)}{\partial z_2} + \frac{\kappa_2 u}{4\mathcal{A}\sqrt{\Delta(z_1, z_2)}} \frac{\partial \Delta(z_1, z_2)}{\partial z_2} - 2\kappa_3 z_2 \\ & + \bar{\mu} g R'(z_1), \end{aligned} \quad (\text{A.2b})$$

**Lemma A.1:** Assume  $R'(z_1) \neq 0$  and  $R'(z_1) \neq -\Gamma(R(z_1))/\Pi(R(z_1))$  for any  $z_1 \in (z_1(t_0), z_1(t_f))$ , where

$$\Gamma(R(z_1)) = \quad (\text{A.3})$$

$$\left( 2\bar{\mu} g \ell_f \ell_r / \sqrt{\ell_f^2 R^2(z_1) + \ell_r^2(\ell_f + \ell_r)^2 - \bar{\mu}^2 g h \ell_f / (\ell_f + \ell_r)} \right) / (1 + \bar{\mu}^2 h^2),$$

$$\Pi(R(z_1)) = \quad (\text{A.4})$$

$$\bar{\mu} g \left( 1 + 2\ell_r^2(\ell_f + \ell_r)^2 / \ell_f^2 R^2(z_1) \right) / 2 \left( 1 + \ell_r^2(\ell_f + \ell_r)^2 / \ell_f^2 R^2(z_1) \right)^{3/2}.$$

Then the manifold  $\partial S \triangleq \{(z_1, z_2) : C_0(z_1, z_2) = 0\}$  is not invariant for the system (42) for any control  $u$ .

*Proof:* Invariance of  $\partial S$  implies the following equation,

$$z_2 \dot{z}_2 - \Pi(R(z_1)) R'(z_1) \dot{z}_1 = 0, \quad (\text{A.5})$$

Since  $\dot{z}_1 = z_2 > 0$ , (A.5) is equivalent to

$$\dot{z}_2 - \Pi(R(z_1)) R'(z_1) = 0. \quad (\text{A.6})$$

Given  $C_0(z_1, z_2) = 0$ , one is able to derive the following condition using (39) and (40),

$$\dot{z}_2^2 + \dot{z}_2 \Gamma(R(z_1)) = 0. \quad (\text{A.7})$$

Equations (A.6) and (A.7) imply either  $\dot{z}_2 = R'(z_1) = 0$  or  $R'(z_1) = -\Gamma(R(z_1))/\Pi(R(z_1))$ , which contradicts the given conditions in Lemma A.1. ■

**Proposition A.2:** Assuming that  $C_0(z_1, z_2) < 0$  holds throughout the optimal trajectory, then there can be no singular subarcs.

*Proof:* The Kuhn-Tucker condition implies that  $\kappa_3 C_0(z_1, z_2) = 0$  and the transversality condition implies  $H(t_f) = 0$ . Since the Hamiltonian does not depend explicitly

on time, it follows that

$$H(t) = 0, \quad \forall t \in [t_0, t_f]. \quad (\text{A.8})$$

Assume that the constraint is inactive, namely,  $C_0(z_1, z_2) < 0$ , hence one obtains  $\kappa_3 = 0$  (Kuhn-Tucker condition). It follows from Pontryagin's Maximum Principle that the optimal control is given by

$$u^* = \arg \min_{u \in [-1, 1]} H(z, \kappa, u) = \begin{cases} -1, & \text{for } \kappa_2 > 0, \\ 1, & \text{for } \kappa_2 < 0, \end{cases} \quad (\text{A.9})$$

which further implies that  $u^* = -\text{sgn } \kappa_2(t)$  where  $\kappa_2$  is the switching function. Suppose there exists a singular control interval  $(t_1, t_2) \subset [t_0, t_f]$  such that  $\kappa_2(t) = 0$  for  $\forall t \in (t_1, t_2)$ .  $\kappa_2(t) = 0$  also implies that  $\dot{\kappa}_2(t) = 0$ . Then it follows from (A.2b) that  $\kappa_1 = 0$  and hence  $H(t) = 1$  for  $t \in (t_1, t_2)$ , which contradicts condition (A.8). ■

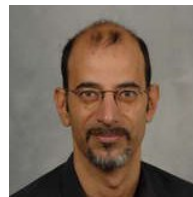
**Proposition A.3:** Assuming that  $C_0(z_1, z_2) < 0$  holds throughout the optimal trajectory and the curvature of the trajectory is constant or monotonically decreasing/increasing, then there can be at most one switching in the control from  $u = 1$  to  $u = -1$ .

*Proof:* Suppose there exists a switching time  $t_1 \in (t_0, t_f)$  such that  $\kappa_2(t_1) = 0$ . Then the transversality condition (A.8) implies  $\kappa_1(t_1) = -1/z_2(t_1)$ . Based on the fact that the velocity  $z_2 > 0$ , one obtains  $\kappa_1(t_1) < 0$  and hence  $\dot{\kappa}_2(t_1) > 0$ . Therefore  $\kappa_2$  changes sign from negative to positive at  $t_1$ , and the control switches from  $u = 1$  to  $u = -1$ .

Similarly, one can see that any other switching point in  $(t_0, t_f)$  has to be from  $u = 1$  to  $u = -1$ . Since there can be no two consecutive switching points from  $u = 1$  to  $u = -1$  without a switching point from  $u = -1$  to  $u = 1$  inbetween,  $t_1$  can be the only switching point in the interval  $(t_0, t_f)$ . ■



**Changxi You** received his B.S. and M.S. degrees from the Department of Automotive Engineering, Tsinghua University of China, and M.S. degree from the Department of Automotive Engineering, RWTH-Aachen University of Germany. He is currently a Ph.D. student under the supervision of Prof. Panagiotis Tsiotras at the School of Aerospace Engineering, Georgia Institute of Technology. His current research interests are in system identification, aggressive driving, path planning and control of (semi)autonomous vehicle.



**Panagiotis Tsiotras** is the David and Andrew Lewis Chair Professor in the School of Aerospace Engineering at the Georgia Institute of Technology. He has held visiting research appointments at MIT, JPL, INRIA Rocquencourt, and Mines ParisTech. His research interests include optimal control of nonlinear systems and ground, aerial and space vehicle autonomy. He has served in the Editorial Boards of the Transactions on Automatic Control, the IEEE Control Systems Magazine, the AIAA Journal of Guidance, Control and Dynamics and the journal Dynamics and Control. He is the recipient of the NSF CAREER award, the Outstanding Aerospace Engineer award from Purdue, and the IEEE Award for Technical Excellence in Aerospace Control. He is a Fellow of AIAA, IEEE, and AAS.