Control Engineering Practice 61 (2017) 307–316

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Safe driving envelopes for path tracking in autonomous vehicles

Matthew Brown *, Joseph Funke, Stephen Erlien, J. Christian Gerdes

Department of Mechanical Engineering, Stanford University, Stanford CA 94305, United States

ARTICLE INFO

Article history:
Received 1 June 2015
Received in revised form
20 April 2016
Accepted 21 April 2016
Available online 4 May 2016

Keywords:
Autonomous vehicles
Path tracking
Model predictive control
Obstacle avoidance
Vehicle stability
First-order hold

ABSTRACT

One approach to motion control of autonomous vehicles is to divide control between path planning and path tracking. This paper introduces an alternative control framework that integrates local path planning and path tracking using model predictive control (MPC). The controller plans trajectories, consisting of position and velocity states, that best follow a desired path while remaining within two safe envelopes. One envelope corresponds to conditions for stability and the other to obstacle avoidance. This enables the controller to safely and minimally deviate from a nominal path if necessary to avoid spinning out or colliding with an obstacle. A long prediction horizon allows action in the present to avoid a dangerous situation in the future. This motivates the use of a first-order hold discretization method that maintains model fidelity and computational feasibility. The controller is implemented in real-time on an experimental vehicle for several driving scenarios.

1. Introduction

In an autonomous vehicle, it can be useful to divide motion control into path planning and path tracking (Leonard et al., 2008; McBride et al., 2008; Montemerlo et al., 2008). In this hierarchical structure, the path planner uses data from perception systems to generate a desired path (desired positions and orientations) the vehicle should follow. The path tracker then calculates and applies steering action to guide the vehicle along the path. As long as the vehicle can feasibly follow the path, this paradigm works well (Campbell, 2007; Thrun et al., 2006). This is especially true in normal driving scenarios characterized by low accelerations in which the kinematic vehicle model often used in path planners approximates vehicle behavior well (Dolgov, Thrun, Montemerlo, & Diebel, 2010; Likhachev & Ferguson, 2009). However, a path planner may not have sufficient or accurate information, especially in cases of changing road conditions or nonlinear dynamic limits of the vehicle. Even if a path planner has access to this data, it is challenging to encode all information needed to safely navigate the environment as a single path. If the path cannot be perfectly tracked, a path tracking controller should possess information about where it is safe to deviate from the path, such as within a lane, and where it is not, such as next to another vehicle.

Path tracking controllers that consider the non-linear dynamics of the vehicle have been shown to be effective in extreme conditions such as racing, fast emergency maneuvers, and icy roads. Kritayakirana et al. used feedforward and lanekeeping-based feedback control to track a racing line even as tires approached saturation (Kritayakirana & Gerdes, 2012). Borrelli, Falcone, Keviczky, and Asgari (2005) presented the path tracking problem as a nonlinear model predictive control (MPC) problem. This controller explicitly considered tire nonlinearities to perform a double lane change under icy conditions. As further explored by Falcone, Borrelli, Tseng, Asgari, and Hrovat (2008), a carefully chosen bound on the states and inputs ensures stability, preventing the vehicle from spinning out by deviating from the path. While this approach does stabilize the vehicle, leaving the path in reaction to severe road conditions could result in hitting an obstacle. The need to go off the path to stabilize the vehicle illustrates the potential dangers of a path not suited for the vehicle and situation.

The main contribution of this paper is a path tracking controller that uses knowledge of the environment and the vehicle's dynamics to safely deviate from the path when necessary to avoid spinning out or hitting an obstacle. In order to accomplish this, the controller constrains predicted states to a region in the state space in which the dynamic vehicle model is stable. This region is known as the stable handling envelope, originally presented by Beal and Gerdes (2013). The predicted states are also constrained to a region in the environment between the road edges and free of obstacles, known as the environmental envelope. These two envelopes were first used in conjunction by Erlien, Fujita, and Gerdes (2013) as an advanced driver assistance system that shared control with a driver. The controller would give full control to the driver unless the driver's commands would cause a future violation of either envelope, in which case the controller would intervene. This allowed the driver to have control of the vehicle in most situations,

* Corresponding author.
E-mail address: mjbrown@stanford.edu (M. Brown).

http://dx.doi.org/10.1016/j.conengprac.2016.04.013
0967-0661/

with the controller only intervening to prevent the vehicle from spinning out or colliding with an obstacle.

This paper discusses the considerations necessary to adapt the envelope approach to path tracking. The presented controller uses MPC to plan a trajectory (a sequence of states) that best follows the desired path while staying inside the stable handling and environmental envelopes. There is a fundamental difference between an envelope controller designed for automated vehicles and one designed for driver assistance: the controller for the automated vehicle has precise knowledge of the desired path instead of simply a projection of driver intent from the steering angle history. With the uncertainty in the driver action removed, error in the vehicle model becomes the limiting factor for performance. Since the controller should ideally reproduce the planned trajectory exactly in the absence of disturbances, the vehicle modeling must improve to take advantage of the increased information about the upcoming path and its curvature. The solution here is to employ a long prediction horizon divided into two parts. In the first few points of the horizon, the vehicle model discretized with short time steps provides a short term plan that accurately captures vehicle dynamics. At later steps in the horizon, a vehicle model discretized with a first-order hold reduces modeling error and more accurately incorporates future curvature information. This discretization method provides the controller with an accurate model over the prediction horizon using few enough points to make the problem computationally feasible. The controller tracks the path in the absence of obstacles or stability concerns; however, it is not restricted to follow the path if the path is not safe.

The following sections of the paper describe the formulation of nominal paths and the derivation of a continuous time dynamic bicycle model. The multi-timestep prediction horizon leads to a discussion of the first-order hold discretization method. Then the optimization problem is presented, including explanations of the vehicle handling envelope and environmental envelope. Experimental results demonstrate the efficacy of the controller in situations in which the nominal path is unsafe and require a deviation. The first situation is a path that has relatively high curvature but is safe to follow, demonstrating the path tracking ability of the controller. The second situation is a hairpin turn too tight for the vehicle's given speed, which shows the controller's ability to take action early to navigate the corner. The third situation is a path with an obstacle in the middle of a turn taken near the limits of handling, showing the simultaneous consideration of stability and obstacle avoidance.

2. Nominal paths

The nominal vehicle path can be as simple as a road centerline or be a very precise sequence of desired positions from a higher level path planner using its understanding of the vehicle's behavior and environment. This path is then passed to the controller as a reference to track. Nominal paths here are defined in terms of the commonly used values curvature K and arc length s (Fraichard & Ahuactzin, 2001; Theodosis & Gerdes, 2011). $E(s)$ and $N(s)$ are defined to be the East and North measures of the position of the path at s from a local datum. The path's heading $\psi_r(s)$ is defined to be the angle from North to a vector parallel to the path at s. Curvature K is related to heading ψ_r and position (E,N) through the following differential equations:

$$\frac{d\psi_r}{ds} = K(s) \tag{1}$$

$$\frac{dE}{ds} = \cos\ \psi_r \tag{2}$$

$$\frac{dN}{ds} = \sin\ \psi_r \tag{3}$$

An example of a nominal path used in experiments is shown in Fig. 1(a): K against s specifies the path. Fig. 1(b) and (c), integrated from the curvature profile with given initial conditions, show the heading and position of the path implied by the chosen curvature. Important environmental information, such as road edges and obstacle locations and sizes, are provided in terms of s. Parsing environmental information is discussed in Section 5.

3. Continuous-time vehicle model

The controller requires a vehicle model that can adequately capture handling limits and loss of stability under some operating conditions. This can be accomplished using a planar bicycle model with two velocity states and three position states as illustrated in Fig. 2. In this controller, front steering angle δ is the only method of actuation. Allowing longitudinal velocity U_x to be variable in the same optimization as δ makes the problem non-convex; while it is possible to track a desired speed profile with an external controller, the expected speed over the prediction horizon is fixed when forming the model. For the experiments presented here, the desired speed is constant.

The velocity states sideslip (β) and yaw rate (r) are described by the following equations of motion (for constant speed):

$$\dot{\beta} = \frac{F_{yf}\cos(\delta) + F_{yr}}{mU_x} - r \approx \frac{F_{yf} + F_{yr}}{mU_x} - r \tag{4}$$

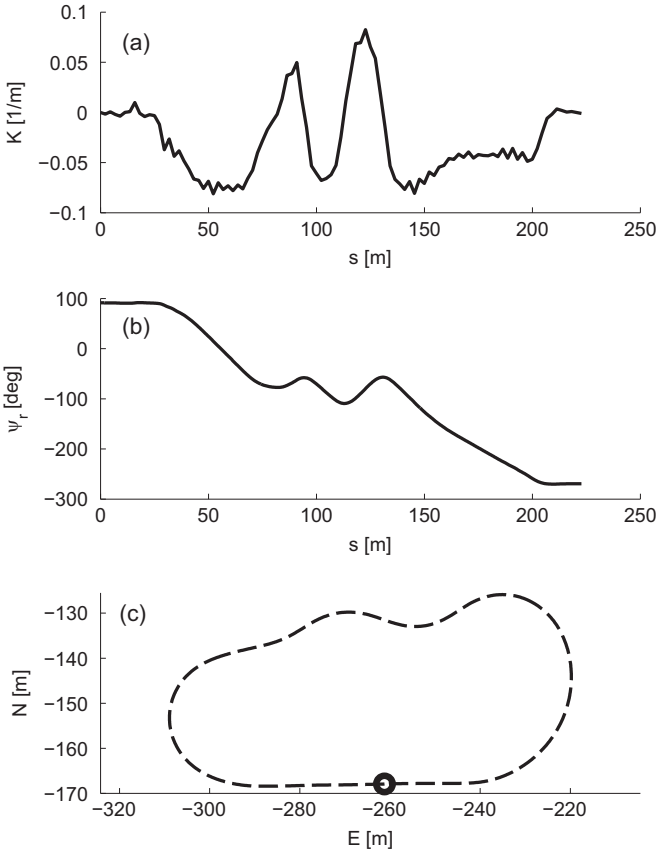

Fig. 1. Curvature (a), heading (b), and position (c) of a nominal path.

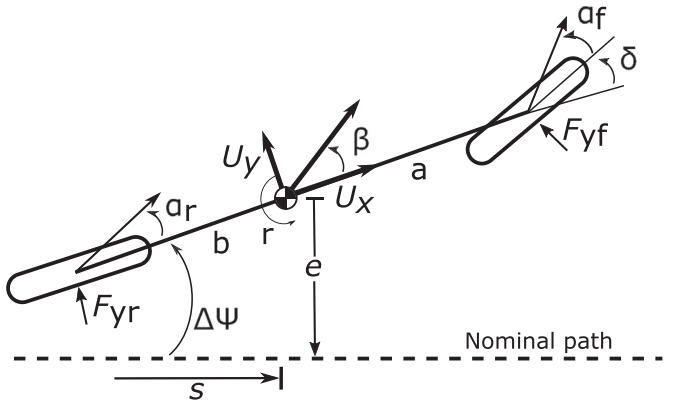

Fig. 2. Bicycle model schematic.

$$\dot{r} = \frac{aF_{yf}\cos(\delta) - bF_{yr}}{I_{zz}} \approx \frac{aF_{yf} - bF_{yr}}{I_{zz}} \tag{5}$$

where $F_{y[f,r]}$ is the lateral tire force of the [front, rear] axle. The vehicle mass, m, and the distances from the center of mass to the front and rear axles, a and b respectively, can be measured by weighing each wheel of the vehicle. Yaw inertia, I_{zz}, can be estimated from ramp steer data. Real-time measurements of β, r, and U_x are available through a GPS/INS system, but estimates of these parameters could also be provided by an estimator (Hsu, Laws, & Gerdes, 2010).

The lateral tire force, $F_{y[f,r]}$, in (4), (5) is modeled as a function of slip angle, $\alpha_{[f,r]}$. The controller uses a brush tire model developed by Fiala (1954) and presented by Pacejka (2002):

$$F_{y[f,r]} = \begin{cases} -C_\alpha \tan\alpha + \dfrac{C_\alpha^2}{3\mu F_z}|\tan\alpha|\tan\alpha \ldots \\ -\dfrac{C_\alpha^3}{27\mu^2 F_z^2}\tan^3\alpha, \quad |\alpha| < \tan^{-1}\left(\dfrac{3\mu F_z}{C_\alpha}\right) = f_{\text{tire}}(\alpha, F_z) \\ -\mu F_z \operatorname{sgn}\alpha, \quad \text{otherwise} \end{cases} \tag{6}$$

where μ is the surface coefficient of friction, F_z is the normal force, and C_α is the tire cornering stiffness.

The parameters μ and C_α can be estimated from experimental ramp steer data, and F_z can be estimated by using the mass of the vehicle. If the speed profile is known to have non-zero longitudinal acceleration, the estimate of F_z can be improved by incorporating static weight transfer or even by modeling suspension dynamics. The experiments presented here use constant speed and therefore calculate the constant F_z from the geometrical static weight distribution, so $f_{\text{tire}}(\alpha, F_z) = f_{\text{tire}}(\alpha)$.

The tire slip angles α_f and α_r, using small angle approximations, can be expressed as:

$$\alpha_f = \tan^{-1}\left(\beta + \frac{ar}{U_x}\right) - \delta \approx \beta + \frac{ar}{U_x} - \delta \tag{7}$$

$$\alpha_r = \tan^{-1}\left(\beta - \frac{br}{U_x}\right) \approx \beta - \frac{br}{U_x} \tag{8}$$

where δ is the front steer angle.

In order to properly account for the nonlinear relationship between front slip angle and front lateral force, the model considers F_{yf} as an input. The MPC optimization calculates a desired F_{yf} which is mapped to δ using (6) and (7):

$$\delta = \beta + \frac{ar}{U_x} - f_{\text{tire}}^{-1}\left(F_{yf}\right) \tag{9}$$

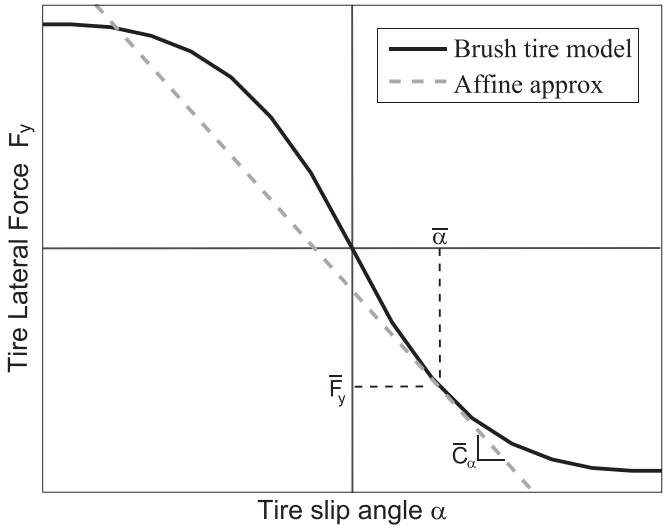

Fig. 3. The brush tire model (for given C_α, μ, and F_z) linearized around predicted $\bar{\alpha}$, allowing for direct consideration of rear tire saturation.

where f_{tire}^{-1} is computed numerically using a lookup table, which provides fast calculation of α_f given F_{yf}. Using F_{yf} as the input instead of δ allows the controller to explicitly consider saturation of the front tire.

The rear tire force is determined by a sequence of linearizations of (6) at predicted rear tire slip angles $\bar{\alpha}_r^k$, for $k = 1, \ldots, T$ over the prediction horizon. A single linearization of the tire model is shown in Fig. 3. The resulting affine expression for ${F_{yr}}^k$ is:

$$F_{yr}^k = -\bar{C}_\alpha^k(\alpha_r - \bar{\alpha}_r^k) + f_{\text{tire}}(\bar{\alpha}_r^k) \tag{10}$$

where $\bar{C}_\alpha^k$ is the local cornering stiffness at a slip angle of $\bar{\alpha}_r^k$. Although the exact sequence of ${\alpha_r}^k$ throughout the prediction horizon is not known *a priori*, it can be approximated by using the sequence of ${\alpha_r}^k$ from the last step of the controller. At each execution step, the controller plans a sequence of states along the prediction horizon. These planned states can be mapped to planned slip angles using (8) for use in linearizing the rear tire model in the next execution of the controller. In practice, this technique of successive linearization converges within a few executions of the controller. The benefits of incorporating rear tire nonlinearities in this way are further discussed by Erlien, Funke, and Gerdes (2014).

The position states in the vehicle model are specified relative to a nominal path: heading deviation $\Delta\psi$, lateral deviation e, and distance s along the path. The equations of motion of the position states can be written as:

$$\Delta\dot{\psi} = r - U_x K(s) \tag{11}$$

$$\dot{e} = U_x \sin(\Delta\psi) + U_y \cos(\Delta\psi) \tag{12}$$

$$\dot{s} = U_x \cos(\Delta\psi) - U_y \sin(\Delta\psi) \tag{13}$$

where $K(s)$ is the curvature of the nominal path at a given s. Making small angle approximations for β and $\Delta\psi$ yields:

$$\dot{e} \approx U_x\Delta\psi + U_x\beta \tag{14}$$

$$\dot{s} \approx U_x - U_x\beta\Delta\psi \approx U_x \tag{15}$$

Thus, a continuous state-space representation of the vehicle

model is expressed for each point k in the prediction horizon $k = 1, \ldots, T$:

$$\dot{\mathbf{x}} = \mathbf{A}_c^k \mathbf{x} + \mathbf{B}_{F_{yf}}^k F_{yf} + \mathbf{B}_K^k K^k + \mathbf{d}_{\bar{\alpha}_r}^k \tag{16}$$

where $\mathbf{x} = [\beta \; r \; \Delta\psi \; s \; e]^T$. The speed profile and road curvature throughout the prediction horizon, $U_x{}^k$ and K^k, are assumed to be known when forming the continuous model. The matrices $\mathbf{A}_c^k$ and $\mathbf{d}_{\bar{\alpha}_r}^k$ contain Eqs. (4), (5), (11), (14), and (15) evaluated at $U_x{}^k$, K^k, and the rear tire linearization parameters $\bar{C}_\alpha^k$ and $f_{\text{tire}}(\bar{\alpha}_r^k)$.

4. Discrete-time vehicle model

The continuous vehicle model described in Section 3 must be discretized to form a discrete time model that can be implemented in a MPC controller. Proper discretization enables the controller to effectively predict vehicle behavior along the prediction horizon.

4.1. Varying time steps

In this implementation, the sampling time (t_s) is chosen to be small enough to accurately capture the propagation of the velocity states β and r at a high frequency. The prediction horizon needs to be long enough to give the vehicle enough time to react to potentially dangerous situations, and 4 s is long enough to allow this. If problem size and computation time were not concerns, the controller would use a constant, short, time step t_s throughout the prediction horizon. However, using short time steps over a relatively long horizon can result in many discretization points that make the problem hard to solve in real-time. For example, for a prediction horizon length of 4 s and $t_s = 0.01$ s would require 400 discretization points.

To address this problem, short time steps t_{short} are used initially to accurately model vehicle stability and longer time steps t_{long} are used later in the prediction horizon to efficiently extend the preview length. The time step at the kth point into the prediction horizon is:

$$t_s^k = \begin{cases} t_{\text{short}} & 1 \le k < T_{\text{corr}} \\ t_{\text{corr}} & k = T_{\text{corr}} \\ t_{\text{long}} & T_{\text{corr}} < k \le T \end{cases} \tag{17}$$

where T is the total number of steps in the prediction horizon and T_{corr} is the number of initial time steps taken at t_{short}. For this implementation, $t_{\text{short}} = 0.01$ s, chosen to be small relative to the dynamics of (4) and (5), and $T_{\text{corr}} = 10$, which are similar to values used by Beal and Gerdes (2013). The computation power of the experimental platform limits $T = 30$; therefore $t_{\text{long}} = 0.20$ s. The correction step t_{corr} is a variable time step whose value ranges from t_{short} to t_{long}. This ensures that points in the prediction horizon correspond to constant points in space as described in Section 4.2.

4.2. Correction time step

One issue that can arise using long time steps in the prediction horizon is the tendency for obstacle locations to appear to change between execution steps of the controller, even if the obstacles are not moving. Fig. 4(a) shows the prediction horizon using long time steps at one point in time. The obstacle is expanded from its true size to meet the nearest discretization points (which are indicated by the thin vertical lines). A short time later, the vehicle moves into the position shown in Fig. 4(b). Because the discretization of the environment has shifted, the representation of the obstacle has also shifted. From the controller's perspective, the obstacle

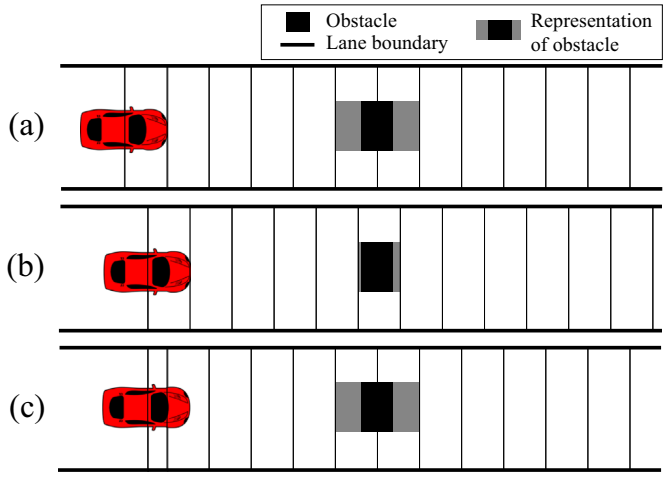

Fig. 4. The change in environmental discretization from one time step (a) to the next (b) can cause changes in the obstacle representation. A correction time step (c) ensures the later points in the environmental discretization correspond to constant points.

appears to move further away. This "moving" obstacle is an artifact of the shifting prediction horizon and is exacerbated by the long time steps.

To fix this, a correction time step after the short time steps ensures the later points correspond to the same points in space, as shown in Fig. 4(c). This means that stationary obstacles will not appear to move; moving obstacles may still "jump" time steps, but that is to be expected as they are moving. Only stationary obstacles are considered in this implementation. The correction time step was first introduced by Erlien et al. (2013).

4.3. Discretization

Every discretization method uses an assumption about the behavior of the input between time steps. A common discretization method for MPC is the zero-order hold (or Euler discretization), which assumes the input, in this case F_{yf}, is held constant over the time step. This matches typical commands to lower-level controllers to hold a constant value until the next control input is calculated. Fig. 5 shows one such input sequence in gray. For $k = 0, \ldots, 9$ the continuous model is discretized with the same time step as that of controller execution, t_{short}, so a zero-order hold provides an accurate model; lower level controllers will attempt to

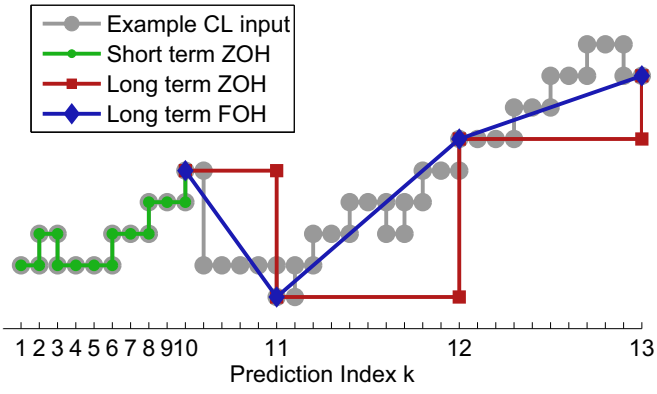

Fig. 5. Planned inputs at the first thirteen points in the prediction horizon compared to a typical closed loop input. A first-order hold discretization is a better approximation of typical closed loop inputs than a zero-order hold discretization. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

hold F_{yf} constant over one t_{short} period. The short term model (the model for $k = 0, \ldots, 9$) can be discretized by considering the augmented state vector $\mathbf{z} = [\mathbf{x}\ F_{yf}\ 1]^{\mathrm{T}}$. The augmented state evolves according the autonomous system:

$$\begin{pmatrix} \dot{\mathbf{x}}(k) \\ \dot{F}_{yf}(k) \\ \dot{1} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_c^k & \mathbf{B}_{F_{yf}}^k & \mathbf{d}_{\alpha_r}^k + \mathbf{B}_K^k K(k) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}(k) \\ F_{yf}(k) \\ 1 \end{pmatrix} = \mathbf{G}^k \mathbf{z}(k) \tag{18}$$

This system can be discretized by taking the matrix exponential of $\mathbf{G}^k t_s^k$

$$\begin{pmatrix} \mathbf{x}(k+1) \\ F_{yf}(k+1) \\ 1 \end{pmatrix} = e^{\mathbf{G}^k t_s^k} \mathbf{z}(k) = \begin{pmatrix} \mathbf{A}^k & \mathbf{B}_1^k & \mathbf{B}_3^k \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}(k) \\ F_{yf}(k) \\ 1 \end{pmatrix} \tag{19}$$

From the second row of (19), the zero-order hold assumption $F_{yf}(k+1) = F_{yf}(k)$ is verified. The first row of (19) gives the discrete model:

$$\mathbf{x}(k+1) = \mathbf{A}^k \mathbf{x}(k) + \mathbf{B}_1^k F_{yf}(k) + \mathbf{B}_3^k \tag{20}$$

The continuous input assumed by this discrete model is shown in Fig. 5 in green, matching a typical closed loop input, in gray.

However, this match between the discretization assumption and closed loop inputs does not hold for $k = 10, \ldots, 29$. A zero-order hold discretization for these indices would assume that the input F_{yf} is constant over one t_{long} period, shown in Fig. 5 in red. This is a poor assumption because in closed-loop, the input is only assumed to be constant over t_{short} seconds; over one t_{long} period, F_{yf} can have significant variation. A zero-order hold discretization in the long time steps effectively assumes the steer angle will be held constant over t_{long}.

A better approach to the long time steps is to discretize with a first-order hold. This assumes that inputs to the system vary linearly between time steps, as indicated in blue in Fig. 5. The results better approximate the system's closed-loop behavior by avoiding the restrictive assumption that front lateral force will be piecewise-constant over relatively long time periods. The assumption that F_{yf} is piecewise-linear is not an exact representation of the discrete closed-loop system, but for typical inputs is a better fit than the zero-order hold assumption.

The first-order hold also enables accurate consideration of road curvature in the vehicle model. Treating the curvature as constant is a good approximation over short distances, but for the longer distances of the long term prediction, it becomes less accurate. In the long term, a better approximation is to assume the path curvature varies linearly between time steps, which can be accomplished by treating curvature as a known input and using a first-order hold. The long term model $(k = 10, \ldots, 29)$ can be discretized by considering the augmented state vector $\mathbf{z} = [\mathbf{x}\ F_{yf}\ K\ 1\ t_s\dot{F}_{yf}\ t_s\dot{K}\ t_s\dot{1}]^{\mathrm{T}}$. The augmented state evolves according to the autonomous system $\dot{\mathbf{z}} = \mathbf{G}^k \mathbf{z}$:

$$\begin{pmatrix} \dot{\mathbf{x}}(k) \\ \begin{pmatrix} \dot{F}_{yf}(k) \\ \dot{K}(k) \\ \dot{1} \end{pmatrix} \\ \begin{pmatrix} t_s^k \ddot{F}_{yf}(k) \\ t_s^k \ddot{K}(k) \\ t_s^k \ddot{1} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \mathbf{A_c^k} & [\mathbf{B}_{F_{yf}}^k v\ \mathbf{B}_K^k\ \mathbf{d}_{\alpha_r}^k] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1/t_s^k \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x}(k) \\ \begin{pmatrix} F_{yf}(k) \\ K(k) \\ 1 \end{pmatrix} \\ \begin{pmatrix} t_s^k \dot{F}_{yf}(k) \\ t_s^k \dot{K}(k) \\ t_s^k \dot{1} \end{pmatrix} \end{pmatrix} \tag{21}$$

This system can be discretized by taking the matrix exponential of $\mathbf{G}^k t_s^k$.

$$\begin{pmatrix} \mathbf{x}(k+1) \\ \begin{pmatrix} F_{yf}(k+1) \\ K(k+1) \\ 1 \end{pmatrix} \\ \begin{pmatrix} t_s^k \dot{F}_{yf}(k+1) \\ t_s^k \dot{K}(k+1) \\ t_s^k \dot{1} \end{pmatrix} \end{pmatrix} = e^{\mathbf{G}^k t_s^k} \mathbf{z}(k) = \begin{pmatrix} \mathbf{A}^k & \Gamma_1^k & \Gamma_2^k \\ \mathbf{0} & \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}(k) \\ \begin{pmatrix} F_{yf}(k) \\ K(k) \\ 1 \end{pmatrix} \\ \begin{pmatrix} t_s^k \dot{F}_{yf}(k) \\ t_s^k \dot{K}(k) \\ t_s^k \dot{1} \end{pmatrix} \end{pmatrix} \tag{22}$$

From the second row of (22), $F_{yf}(k+1) = F_{yf}(k) + t_s \dot{F}_{yf}(k)$, which is true if $\dot{F}_{yf}(k)$ is constant over $t_s{}^k$, verifying the first-order hold assumption. Similarly from row three, $K(k+1) = K(k) + t_s^k \dot{K}(k)$, which is true if $\dot{K}(k)$ is constant over $t_s{}^k$. The first row of (22) gives the discrete model, after making the substitutions $t_s^k \dot{F}_{yf}(k) = F_{yf}(k+1) - F_{yf}(k)$ and $t_s^k \dot{K}(k) = K(k+1) - K(k)$:

$$\mathbf{x}(k+1) = \mathbf{A}^k \mathbf{x}(k) + (\Gamma_1^k - \Gamma_2^k) \begin{pmatrix} F_{yf}(k) \\ K(k) \\ 1 \end{pmatrix} + \Gamma_2^k \begin{pmatrix} F_{yf}(k+1) \\ K(k+1) \\ 1 \end{pmatrix} \tag{23}$$

Defining $\mathbf{B}_1^k$ to be the first column of $(\Gamma_1^k - \Gamma_2^k)$, $\mathbf{B}_2^k$ to be the first column of Γ_2, and lumping the remaining (constant) terms into $\mathbf{B}_3^k$, the long term discrete model can be written

$$\mathbf{x}(k+1) = \mathbf{A}^k \mathbf{x}(k) + \mathbf{B}_1^k F_{yf}(k) + \mathbf{B}_2^k F_{yf}(k+1) + \mathbf{B}_3^k \tag{24}$$

for $k = 11, \ldots, 29$. More background on the first-order hold, or triangle hold, can be found in Franklin, Workman, and Powell (1997).

Note that while the zero-order hold model is strictly causal, the first-order hold model is not. The next state $\mathbf{x}(k+1)$ depends on the next input $F_{yf}(k+1)$. This property could present problems if future inputs are unknown, but because model predictive control calculates all inputs in the prediction horizon simultaneously, the model used does not need to be causal.

Fig. 6 compares planned and closed loop front lateral force computed while tracking a nominal path in simulation. The same path is tracked with a controller using a zero-order hold (top) and first-order hold (bottom) in the long term prediction. The first-order hold model is a better predictor of closed loop behavior, and is therefore more useful in making decisions in the present.

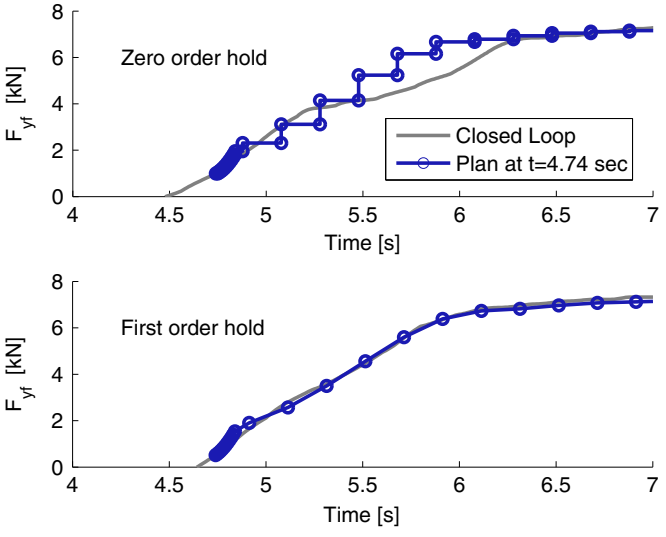

Fig. 6. Comparison of long-term plan assuming a zero-order hold on front lateral force (top) against long-term plan assuming a first-order hold with the same path (bottom). The first-order hold more accurately represents the closed-loop behavior of the controller.

5. Safe driving envelopes

5.1. Stable handling envelope

The stable handling envelope is defined by the limits on vehicle states β and r, shown in Fig. 7. These limits reflect the maximum capabilities of the tires, under steady state assumptions and the given tire model. As shown by Beal and Gerdes (2013), the nominal vehicle dynamics (4) and (5) can be stabilized within the envelope. The stable handling envelope is a control invariant set: for every state within the envelope, there exists a steering input to keep the vehicle within the envelope.

A steady-state analysis yields bounds on the yaw rate and sideslip that do not exceed the steady-state limits of the tires on the road. Considering (5) in steady-state yields:

$$r_{ss} = \frac{F_{yf} + F_{yr}}{mU_x} \tag{25}$$

Assuming no longitudinal tire force results in the bounds ② and ④:

$$|r_{ss}| \leq \frac{g\mu}{U_x} \tag{26}$$

Another important limit when considering vehicle stability is the saturation of the rear tires. The rear slip angle α_r that produces maximum lateral force for the given tire model is:

$$\alpha_{r,\text{peak}} = \tan^{-1}\left(3\frac{mg\mu}{C_{\alpha_r}}\frac{a}{L}\right) \tag{27}$$

Using this value of maximum α_r and (8) results in the bounds ① and ③:

$$|\beta_{ss}| = \alpha_{r,\text{peak}} + \frac{br}{U_x} \tag{28}$$

By restricting the planned vehicle states to remain in this "safe" polyhedron, the vehicle never enters regions of the state space that would result in a spin. This restriction can be compactly represented by the linear inequality:

$$\mathbf{H}_{\text{veh}}\mathbf{x}(k) \leq \mathbf{G}_{\text{veh}}, \quad k = 1, \ldots, 30 \tag{29}$$

While the model is stable within the envelope (a steering input always exists to push the state back within the envelope), it is not true that the vehicle becomes unstable as soon as the state leaves the envelope. The bounds in (29) are based on steady-state assumptions; from empirical tests, the vehicle state can be outside the stable handling envelope and still return to the envelope after a short excursion. It may be possible to find a larger "stable" region, and further investigation is an area of active research. Nevertheless, the stable handling envelope is still useful in planning trajectories that are guaranteed to be stable.

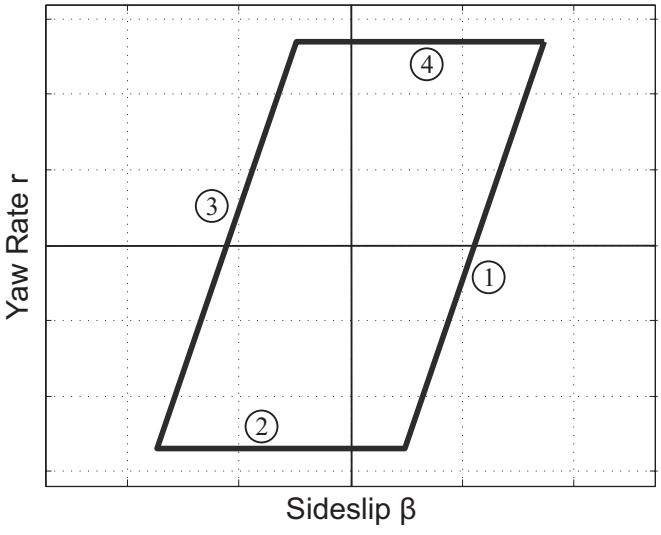

Fig. 7. The stable handling envelope represents a safe region of sideslip and yaw rate. For every state in the envelope, there exists an input to keep the state within the envelope.

5.2. Environmental envelope

In addition to planning stable trajectories, the controller also ensures that planned trajectories do not collide with obstacles or other environmental bounds like road edges by using the concept of the environmental envelope. Originally presented by Erlien et al. (2013), the environmental envelope is a convex set of points in (s,e), called tubes, that are free of obstacles and are within road boundaries. For each tube, a set of bounds on e defines the safe region:

$$e^k < e^k_{\max} - \tfrac{1}{2}w - e_{\text{buffer}} \tag{30}$$

$$e^k > e^k_{\min} + \tfrac{1}{2}w + e_{\text{buffer}} \tag{31}$$

where w is the width of the car and e_{buffer} is the desired minimum distance to an obstacle or road boundary. These inequalities can be compactly expressed by a matrix inequality:

$$\mathbf{H}_{\text{env}}\mathbf{x}(k) \leq \mathbf{G}_{\text{env}}, \quad k = 11, \ldots, 30 \tag{32}$$

Fig. 8 shows the methodology to generate the environmental envelope from an arbitrary environment like the one pictured in Fig. 8a. The vehicle's predicted $s(k)$ along the path can be determined from $\dot{s}$ (15), the known U_x, and the sampling times $t_s{}^k$. The environment can be sampled at these discrete points along the path to determine safe regions in e for each k that are between road bounds and free of obstacles, shown in Fig. 8b.

Constraining the vehicle's planned $e(k)$ to be within these safe

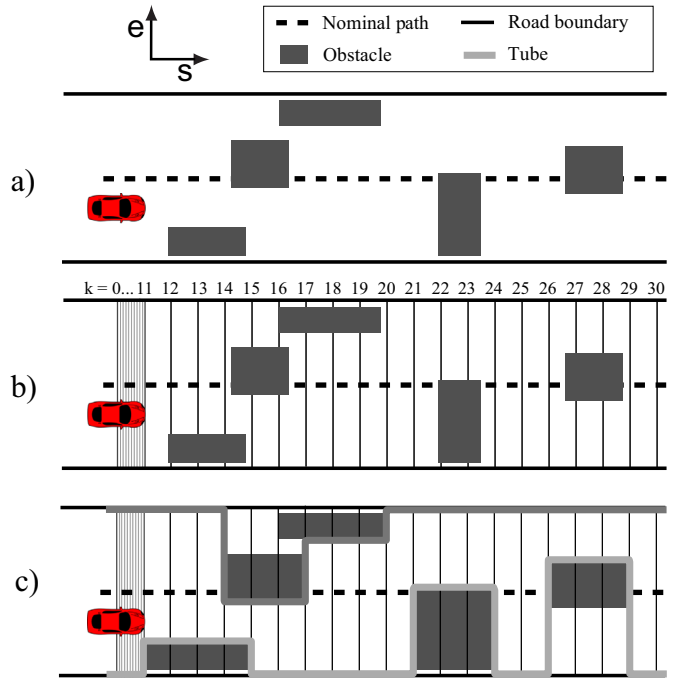

Fig. 8. The structured environment (a) can be discretized (b) to create continuous collision-free regions in (s,e) known as tubes. One such tube is shown in (c).

regions could result in a non-convex constraint on e. One such example is the feasible e at $k=27$ in Fig. 8b, which consists of two disjoint sets (corresponding to being left or right of the obstacle). Instead, the vehicle's trajectory needs to be fully contained within a *tube*, defined to be a sequence of continuous feasible e along the prediction horizon. One such tube is illustrated in Fig. 8c.

Each tube is considered in a separate convex optimization problem and the globally optimal tube can be found by comparing objective function values of the problems corresponding to each tube. Even though n obstacles can mean as many as 2^n tubes, normal driving scenarios have a relatively small number of obstacles in the prediction horizon so this is not a practical concern. Heuristics could further be used to eliminate suboptimal tubes. The tests presented in this paper have at most one obstacle (two tubes).

While a higher level path planner would presumably attempt to plan safe paths that avoid obstacles, it is still necessary to build obstacle avoidance into the controller to account for two situations. First, if it is necessary to deviate from the path for any reason, the controller needs to know where it can leave the path safely. Secondly, if an obstacle is detected after the path has been generated, the nominal path may pass through an obstacle. In either case, considering the environmental envelope and restricting states to lie within it ensures that the vehicle remains collision-free.

6. Optimization

Central to the MPC controller is an optimization problem. At each time step, the controller solves a convex optimization problem to find an optimal sequence of front lateral force inputs. It maps the first of this sequence into a steer angle through (9) and commands this steer angle to lower level controllers. The optimization problem solved is:

$$\min_{F_{yf}} \sum_{k=1}^{30} (\mathbf{x}^k)^{\mathrm{T}}\mathbf{Q}\mathbf{x}^k + (v^k)^{\mathrm{T}}Rv^k + \mathbf{W}_{\mathrm{veh}}\boldsymbol{\sigma}_{\mathrm{veh}} + \mathbf{W}_{\mathrm{env}}\boldsymbol{\sigma}_{\mathrm{env}} \tag{33}$$

subject to the vehicle model (20) and (24) and:

$$\mathbf{H}_{\mathrm{veh}}\mathbf{x}^k \le \mathbf{G}_{\mathrm{veh}} + \boldsymbol{\sigma}_{\mathrm{veh}} \quad \forall\, k \tag{29}$$

$$\mathbf{H}_{\mathrm{env}}\mathbf{x}^k \le \mathbf{G}_{\mathrm{env}} + \boldsymbol{\sigma}_{\mathrm{env}} \quad k = 11, \ldots, 30 \tag{32}$$

$$F_{yf}^k \le F_{max} \quad \forall\, k \tag{34}$$

$$|v^k| \le v_{max}^k \quad \forall\, k \tag{35}$$

where $F_{\max}$ and $v_{\max}^k$ are the maximum lateral force and slew rate capabilities of the vehicle, respectively. The objective function does not penalize the magnitude of the front lateral force directly, but rather it weighs the change in front lateral force through the variable $v^k = F_{yf}^k - F_{yf}^{k-1}$. This essentially only assigns cost to changes in steer angle, allowing the vehicle to hold a constant steer angle during a turn without penalty.

The safe driving envelope constraints (29) and (32) use the nonnegative slack variables $\boldsymbol{\sigma}_{\mathrm{veh}}$ and $\boldsymbol{\sigma}_{\mathrm{env}}$ to ensure that the optimization problem is always feasible. The weights $\mathbf{W}_{\mathrm{veh}}$ and $\mathbf{W}_{\mathrm{env}}$ are set to be relatively large, such that the slack variable terms dominate the cost function even if the slack variables are relatively small.

For the following experiments,

$$\mathbf{Q} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix},$$

$$R = 10,$$

$$\mathbf{W}_{\mathrm{veh}} = (5 \;\; 5 \;\; 5 \;\; 5),$$

$$\mathbf{W}_{\mathrm{env}} = (3 \times 10^4 \;\; 3 \times 10^4) \tag{36}$$

These weights were chosen through iteratively tuning in simulation and experiment. Empirically, the optimization problem seems to be insensitive to small changes in the weights. The relative order of magnitude of the weights sets the preference for satisfying the envelope constraints first, then path tracking, then reducing changes in steering.

6.1. Implementation

The convexity of the optimization problem allows for efficient real-time implementation. CVXGEN, developed by Mattingley and Boyd (2012), generates a solver that is implemented on a single core of an i7 processor. The resulting controller is capable of linearizing and discretizing the vehicle model, forming the two envelopes, and solving the optimization problem for two tubes in less than 10 ms allowing for a controller execution rate of 100 Hz.

7. Experimental results

7.1. Experimental vehicle

This controller was implemented on X1 (Fig. 9), a student-built research vehicle. X1 is a fully steer-by-wire, drive-by-wire, and brake-by-wire vehicle that uses differential GPS and INS for precise position and inertial measurements. Model parameters for X1 are given in Table 1. Experiments were performed on an asphalt surface with empirically estimated friction coefficient $\mu = 0.75$. In this implementation, the nominal path was assumed fixed and known *a priori*, but in the future the nominal path could be the output of a path planner acting on sensor information.

7.2. Tracking

If tracking a nominal path does not require forces beyond the capabilities of the tires, the controller should track the path as closely as possible. Fig. 10 (top) shows a nominal path with high but feasible curvature, free of obstacles. The controller's planned trajectory is shown in blue, each circle representing a discrete point in the prediction horizon. At the illustrated point in time, the

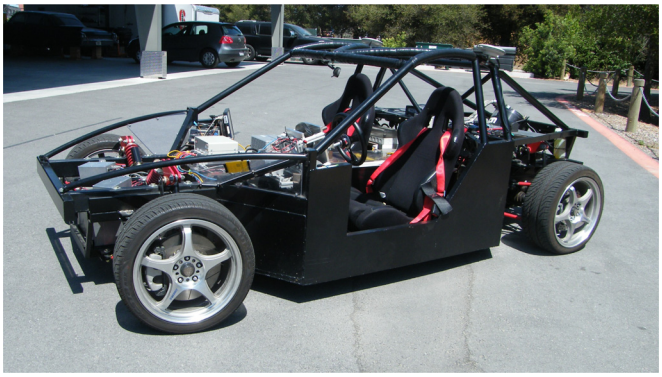

Fig. 9. X1: a steer-by-wire experimental platform.

controller plans to track the path very closely, as indicated by the blue circles on the nominal path. In closed loop, the controller tracks the path with less than 0.2 m of lateral error, see Fig. 10 (bottom), while traveling at 9 m/s and experiencing a peak lateral acceleration of 0.6g's. This approaches, but does not exceed, the limits of the tires.

Fig. 10 shows the (close) match between the planned and closed loop trajectories. This indicates that the planned trajectory is an accurate prediction of the vehicle's closed loop behavior. This consistency along the prediction horizon ensures accuracy and usefulness of the long term prediction.

Table 1
Parameters of X1.

Parameter	Value	Units
m	2009	kg
I_{zz}	2000	kg m^2
a	1.53	m
b	1.23	m
$C_{\alpha f}$	1.1441×10^5	N/rad
$C_{\alpha r}$	1.3388×10^5	N/rad
μ	0.75	n/a

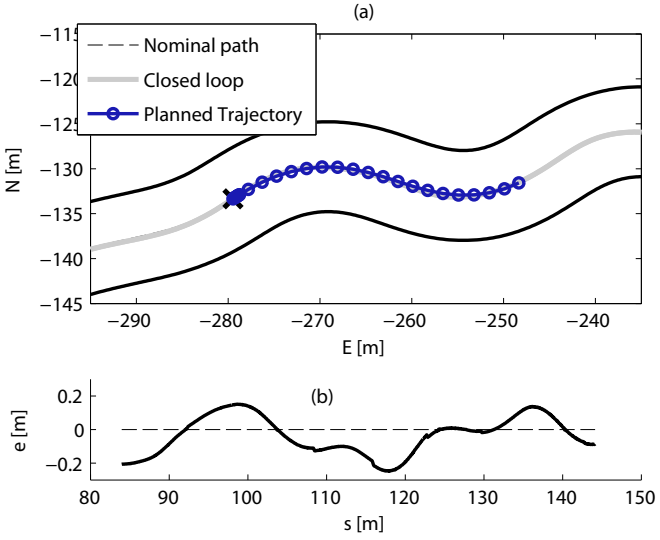

Fig. 10. The spatial trajectory of the vehicle tracking a nominal path (a) and the resulting lateral error (b). The vehicle tracks the nominal path in the absence of obstacles or stability envelope violations.

7.3. Hairpin turn

While in many situations a higher-level path planner will produce a dynamically feasible path, there may be some instances where the nominal path cannot be followed at the given speed. Such situations may occur due to changing environmental conditions or an imperfect understanding of the vehicle capabilities during path planning. The controller developed in this paper offers considerable flexibility and robustness in these situations as a

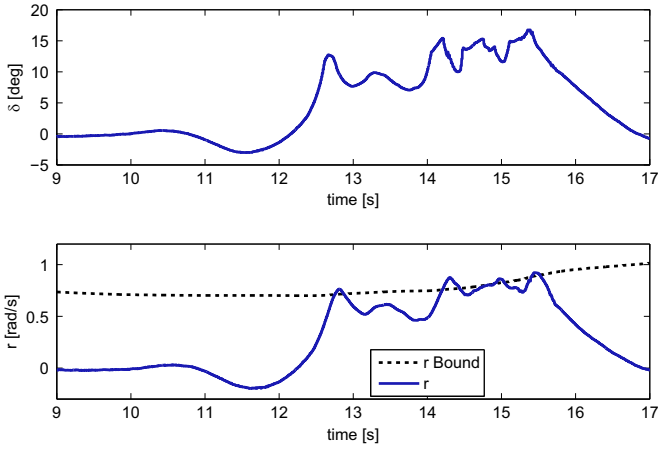

Fig. 12. The steer angle (top) during the hairpin turn shown in Fig. 11 and the yaw rate and its bound (bottom). The controller counter-steers as the yaw rate approaches and reaches the bound.

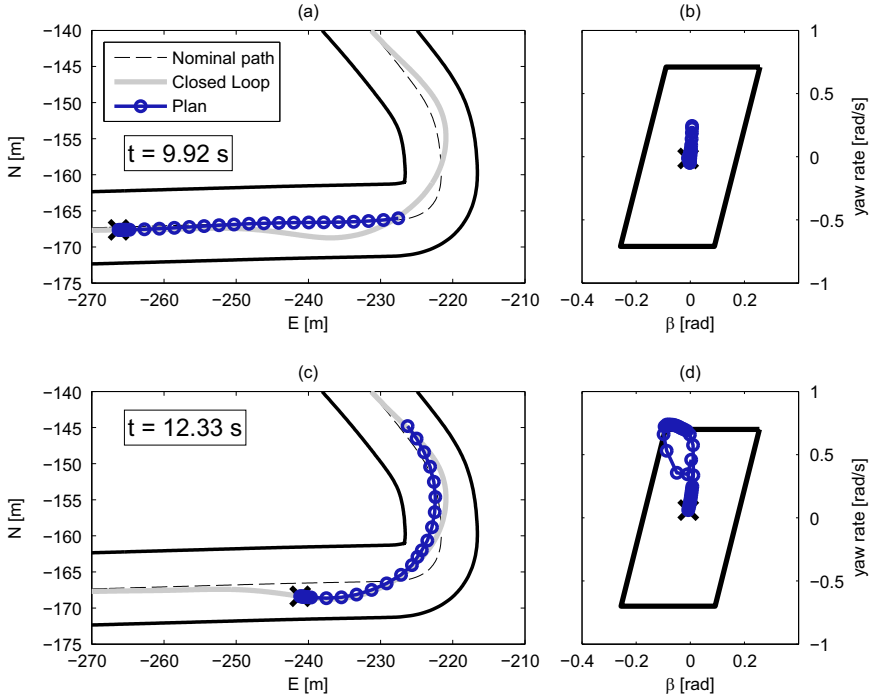

Fig. 11. The desired path at the given speed would violate dynamic constraints; the controller must deviate from the path to remain within the limits of the handling.

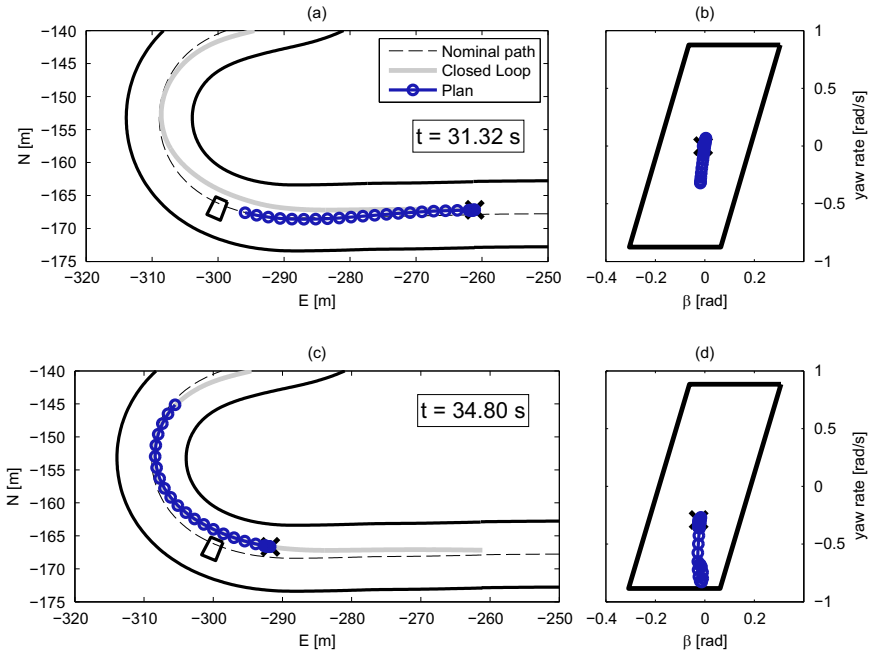

Fig. 13. When an obstacle obstructs the path, the controller deviates from the path and returns once past. Even when avoiding an obstacle, the stable handling constraint is enforced.

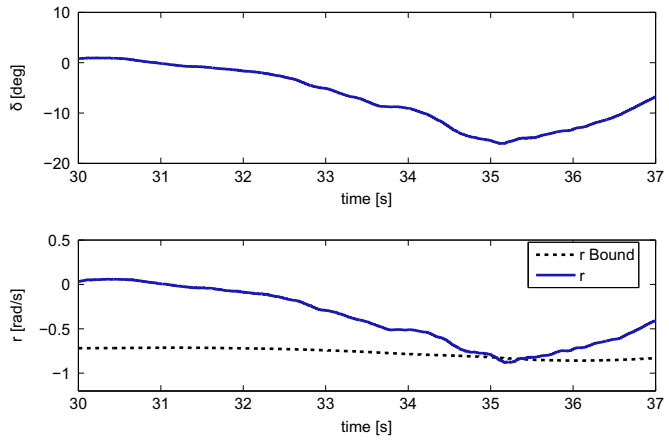

Fig. 14. The steer angle (top) during the obstacle avoidance turn shown in Fig. 13 and the yaw rate and its bound (bottom). The controller smoothly steers past the obstacle and reduces steer angle as the yaw rate reaches its bound.

result of its ability to rapidly modify the path in real time. The following two sections demonstrate the controller flexibility and the mechanisms behind it in a couple of different scenarios.

The first scenario is a path with high curvature relative to the vehicle speed, as represented by the hairpin turn shown in Fig. 11. While manageable at low speeds, the finite amount of frictional force available means that tracking is infeasible at higher speeds and attempting to follow the path may result in the vehicle spinning out. In this experiment, the vehicle is initially traveling at 10 m/s, making perfect tracking of this path impossible under the existing friction conditions.

Along the straight leading up the hairpin, the controller plans to track the path. As the end of the prediction horizon reaches the corner (Fig. 11a), the planned trajectory plans both to track the nominal path and to remain within the stable handling envelope (Fig. 11b). As the vehicle travels closer to the corner (Fig. 11c), it is not possible to find a trajectory that tracks the path and stays within the stable handling envelope. A deviation from the nominal path is necessary to keep the planned trajectory safely inside the stable handling envelope (Fig. 11d). The vehicle "swings wide" around the tight corner to stay within the limits of the tires and not spin out. The controller cannot arbitrarily deviate from the path; the environmental envelope provides information about where deviations are safe. In general, the road edge at the inside corner bounds how much the vehicle can "cut" the corner, although in this case the environmental constraint is not active.

In Fig. 11c, there is some discrepancy between the planned and closed loop trajectories late in the prediction horizon, as the vehicle pushes further out in the corner than planned. One contributing factor is the immediate response of the controller to vehicle states leaving the stable handling envelope. Fig. 12 (bottom) shows the measured yaw rate during the hairpin maneuver and the corresponding bound from (26). The steering angle in Fig. 12 (top) shows the slight turn to the left around 11 s and the hard turn into the corner beginning around 12 s. As the yaw rate approaches the stable handling envelope bound (around 12.75 s), however, the controller must counter steer, reducing the steer angle to push the yaw rate back under the upper bound and within the envelope. The re-planning nature of the controller enables it to react quickly if the vehicle behaves differently than initially planned.

A long prediction horizon is critical for allowing early action in anticipation of the upcoming hairpin corner. An accurate discrete vehicle model in the long term plan allows the controller to discern that staying on the path through the corner is not feasible. This enables the controller to begin taking action about four seconds before the reaching the corner.

7.4. Obstacle in turn

A second scenario where flexibility is desirable is when the nominal path is blocked by an obstacle that suddenly appears or comes into the sensor range of the vehicle. Fig. 13 depicts a scenario in which the nominal path is blocked, while the vehicle is traveling at 10 m/s. As shown in Fig. 13a, the controller plans a trajectory that is closely aligned with the path and stays within the

stable handling envelope (Fig. 13b). When the end of the prediction horizon reaches the obstacle (corresponding to the obstacle coming into the vehicle's sight), the planned trajectory shifts to avoid it. As the vehicle progresses closer to the corner (Fig. 13c), the competing objectives of returning to the path after the obstacle by veering left and taking the corner by turning right threaten to destabilize the vehicle (as shown by the active stable handling constraint in Fig. 13). The stable handling envelope ensures that the vehicle safely returns to the path post-obstacle.

The steering angle and yaw rate through this maneuver are shown in Fig. 14. The controller smoothly turns to the right to avoid the obstacle, backing off as the vehicle reaches the apex of the turn at 35 s and the yaw rate reaches the stable handling bound.

A key aspect of the controller is the simultaneous consideration of obstacle avoidance and stabilization. The controller constrains the planned trajectory to the stable handling envelope even as it avoids obstacles. This can result in deviations from the nominal path that are larger than strictly necessary to avoid contact, but ensure that the vehicle is stable while making potentially aggressive maneuvers to avoid obstacles.

8. Conclusion

Situations which require deviation from a desired path present a challenge for traditional path tracking controllers because they are not given information about where it is safe to deviate from the path. The controller presented in this paper uses environmental information and a vehicle model to ensure that, should deviations from the path be necessary to stabilize the vehicle or avoid obstacles, the vehicle will deviate in a way that is safe. Experimental results of the controller tracking a path with very high curvature and a path through an obstacle demonstrate the ability of the controller to simultaneously achieve stabilization and obstacle avoidance.

Acknowledgments

The authors would like to thank OmniSTAR for providing GPS corrections and Thunderhill Raceway Park for accommodating testing. Brown and Funke are supported by the Graduate Research Fellowship from the National Science Foundation.

References

Beal, C. E., & Gerdes, J. C. (2013). Model predictive control for vehicle stabilization at the limits of handling. *IEEE Transactions on Control Systems Technology*, *21*(4), 1258–1269.

Borrelli, F., Falcone, P., Keviczky, T., & Asgari, J. (2005). Mpc-based approach to active steering for autonomous vehicle systems. *International Journal of Vehicle Autonomous Systems*, *3*(2), 265–291.

Campbell, S. F., (2007). *Steering control of an autonomous ground vehicle with application to the darpa urban challenge* (Ph.D. dissertation). Massachusetts Institute of Technology.

Dolgov, D., Thrun, S., Montemerlo, M., & Diebel, J. (2010). Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, *29*(5), 485–501.

Erlien, S., Fujita, S., & Gerdes, J. C. (2013). Safe driving envelopes for shared control of ground vehicles. *Advances in Automotive Control*, *7*(1), 831–836.

Erlien, S.M., Funke, J., & Gerdes, J.C., (2014). Incorporating non-linear tire dynamics into a convex approach to shared steering control. In *American control conference (ACC)* (pp. 3468–3473). IEEE, Portland, OR.

Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., & Hrovat, D. (2008). Linear time-varying model predictive control and its application to active steering systems: *Stability analysis and experimental validation*. *International Journal of Robust and Nonlinear Control*, *18*(8), 862–875.

Fiala, E. (1954). Lateral forces on rolling pneumatic tires. *Zeitschrift VDI*, *96*(29), 973–979.

Fraichard, T., Ahuactzin, J. M. (2001). Smooth path planning for cars. In *ICRA* (pp. 3722–3727).

Franklin, G. F., Workman, M. L., & Powell, D. (1997). *Digital control of dynamic systems*. Half Moon Bay, CA: Addison-Wesley Longman Publishing Co., Inc.

Hsu, Y.-H., Laws, S. M., & Gerdes, J. C. (2010). Estimation of tire slip angle and friction limits using steering torque. *IEEE Transactions on Control Systems Technology*, *18*(4), 896–907.

Kritayakirana, K., & Gerdes, J. C. (2012). Autonomous vehicle control at the limits of handling. *International Journal of Vehicle Autonomous Systems*, *10*(4), 271–296.

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., et al. (2008). A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, *25*(10), 727–774.

Likhachev, M., & Ferguson, D. (2009). Planning long dynamically feasible maneuvers for autonomous vehicles. *The International Journal of Robotics Research*, *28*(8), 933–945.

Mattingley, J., & Boyd, S. (2012). Cvxgen: *A code generator for embedded convex optimization*. *Optimization and Engineering*, *13*(1), 1–27.

McBride, J. R., Ivan, J. C., Rhode, D. S., Rupp, J. D., Rupp, M. Y., Higgins, J. D., et al. (2008). A perspective on emerging automotive safety applications, derived from lessons learned through participation in the darpa grand challenges. *Journal of Field Robotics*, *25*(10), 808–840.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., et al. (2008). Junior: *The Stanford entry in the urban challenge*. *Journal of Field Robotics*, *25*(9), 569–597.

Pacejka, H. B. (2002). *Tire and vehicle dynamics*. Oxford, UK: Butterworth-Heinemann (Elsevier).

Theodosis, P.A., & Gerdes, J. C., (2011). Generating a racing line for an autonomous race car using professional driving techniques. In *ASME 2011 dynamic systems and control conference and Bath/ASME symposium on fluid power and motion control* (pp. 853–860). American Society of Mechanical Engineers, Arlington, VA.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., et al. (2006). Stanley: *The robot that won the darpa grand challenge*. *Journal of Field Robotics*, *23*(9), 661–692.