Predictive Control for Agile Semi-Autonomous Ground Vehicles using Motion Primitives

Andrew Gray*, Yiqi Gao*, Theresa Lin*, J. Karl Hedrick*, H. Eric Tseng†, Francesco Borrelli*
*Department of Mechanical Engineering
University of California
Berkeley, CA, USA
†Ford Research Laboratories
Dearborn, MI, USA

***Abstract*— This paper presents a hierarchical control framework for the obstacle avoidance of autonomous and semi-autonomous ground vehicles. The high-level planner is based on motion primitives created from a four-wheel nonlinear dynamic model. Parametrized clothoids and drifting maneuvers are used to improve vehicle agility. The low-level tracks the planned trajectory with a nonlinear Model Predictive Controller. The first part of the paper describes the proposed control architecture and methodology. The second part presents simulative and experimental results with an autonomous and semi-autonomous ground vehicle traveling at high speed on an icy surface.**

I. INTRODUCTION

Real-time generation and tracking of feasible trajectories is a major issue in autonomous guidance systems when the vehicle is traveling at the limits of its handling capability. Trajectories generated by using oversimplified models violate system constraints, while the computation of trajectories by using high-fidelity vehicle models and nonlinear optimization is computationally demanding. Moreover, the presence of external disturbances and model uncertainties might still prevent the vehicle from following the desired path.

Because of its capability to systematically handle system nonlinearities and constraints, work in a wide operating region and close to the set of admissible states and inputs, Model Predictive Control (MPC) is an attractive method to generate feasible trajectories and robustly track them [1]. Parallel advances in theory and computing systems have enlarged the range of applications where real-time MPC can be applied [2], [3], [4], [5], [6]. Yet, the computational burden of Nonlinear MPC (NMPC) is still a barrier for agile autonomous drive. As an example, in [7] an NMPC has been implemented on a passenger vehicle. It was shown that the real-time execution (20 Hz) is limited to low vehicle speeds on icy roads, because of its computational complexity. The complexity of the planning problem is further increased when obstacles are considered. Previous work by the authors [8] have accounted for pop-up obstacles by decomposing the problem into the two-level NMPC problem, depicted in Figure 1. The high-level "Path Planner" uses a simplified point-mass vehicle model to generate an obstacle avoiding trajectory by using a NMPC controller. The trajectory is fed to the low-level "Path Follower" designed by using a NMPC based on a higher fidelity vehicle model [1]. In [8] the proposed hierarchical framework has been implemented on an autonomous ground vehicle driving at high speeds on an icy road.

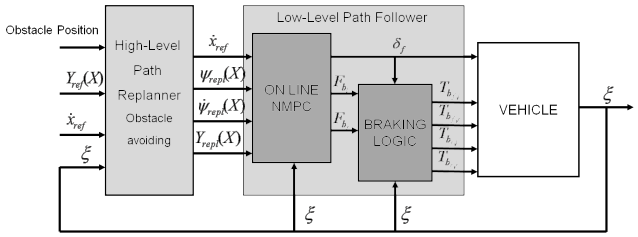

Fig. 1. Architecture of the two-level MPC. The point-mass vehicle model is used for high-level path planning. The four-wheel vehicle model is used for low-level path tracking.

Although the decomposition in Figure 1 allows for real-time implementation, the trajectories generated by the point-mass path planner are not always feasible. The lower level tracking performance deteriorates and obstacle collisions can be observed in conditions where the obstacle could have been avoided.

In order to overcome this issue and still maintain real-time feasibility, this paper *studies the use of a motion primitive path planner*. Path planners based on motion primitives for dynamical systems have been first introduced in [9]. There, the author considered two types of primitives: trims and maneuvers. Trims are steady-state equilibrium trajectories and maneuvers are pre-specified trajectories connecting trims, designed offline, and stored in a library. The method shifts the complexity of nonlinear dynamical optimization to the sequencing of "useful" trims and maneuvers together in order to generate a feasible path. If the number of motion primitives is small, we expect the combinatorial nature of the resulting problem to be easily handled by state-of-the-art mixed-integer solvers. Moreover, if the environment is highly structured (as in urban driving), one can expect to generate feasible trajectories for a variety of scenarios with only a few motion primitives.

*Email: {ajgray,yiqigao,tlin051,khedrick,fborrelli}@berkeley.edu
Email: htseng@ford.com

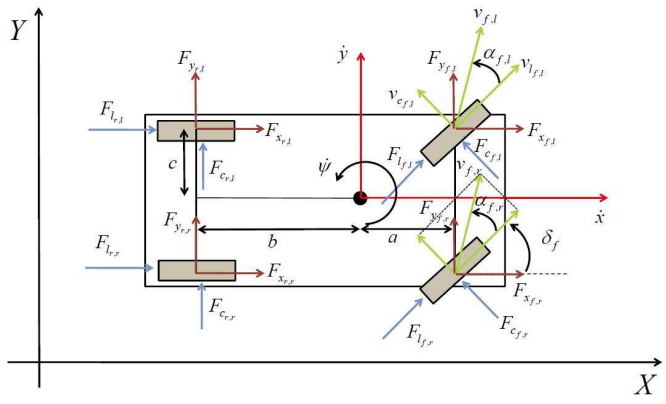

Fig. 2. Simplified four-wheel dynamic model.

The motion primitive approach has been used to successfully plan time-optimal paths for small robotic helicopters [9], [10] and other simple mechanical systems [11]. However, to the best of the authors' knowledge, this method has not yet been implemented on autonomous ground vehicles. The paper is structured in two parts. In the first part of the paper we describe in detail the proposed hierarchical control architecture and the approach to generate a trajectory by using motion primitives. We will provide details on specific lane-change and drifting motion primitives which are very useful for designing a semi-autonomous vehicle with adjustable degree of autonomy. In the second part, simulative and experimental results with an autonomous and semi-autonomous ground vehicle traveling at high speed on an icy surface are presented.

II. Four Wheel Vehicle Model

We use the vehicle model presented in [1] for the control design. This section provides the basic equations, more details can be found in [1]. The model captures the main vehicle lateral and longitudinal dynamics and uses a nonlinear Pacejka tire model to compute tire forces as a function of wheel longitudinal slips, lateral slips and road friction coefficient.

The vehicle dynamics can be compactly written as

$$\dot{\xi}(t) = f^{4w}(\xi(t), u(t)) \tag{1}$$

where $\xi(t) \in \mathbb{R}^6$ is the system state and $u(t) \in \mathbb{R}^3$ is the system input. The six state components are grouped into two parts $\xi = [\xi_g,\ \xi_b]$. The vector $\xi_g = [X, Y, \psi]$ collects longitudinal and lateral coordinates in the inertial frame and the yaw angle. The vector $\xi_b = [\dot{x}, \dot{y}, \dot{\psi}]$ collects the longitudinal and lateral velocities in the body frame and the yaw rate. The three inputs are $u = [\delta_f, F_{b_l}, F_{b_r}]'$ where δ_f is the front steering angle and F_{b_l}, F_{b_r} are the left and right braking forces.

The dynamics in Equation (1) are derived by using the equations of motion about the vehicles Center of Gravity (CoG) and coordinate transformations between the inertial frame and the vehicle body frame:

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \tag{2a}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}} \tag{2b}$$

$$\begin{aligned} I\ddot{\psi} = {} & a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + \\ & + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}) \end{aligned} \tag{2c}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi \tag{2d}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi \tag{2e}$$

where the constant m is the vehicle's mass, I is the rotational inertia about the yaw axis, a and b are the distances from the CoG to the front and rear axles and c is the distance from the CoG to the left/right side at the wheels (see Figure 2). The tire forces in Equation (3) are computed using a Pacejka tire model [12].

In this paper a sliding mode controller is used to control the braking forces F_{b_l} and F_{b_r} in order to track a desired rear slip ratio. Model (2) in closed loop with the slip ratio controller can be compactly written as

$$\dot{\xi}(t) = f_{\sigma}^{4w}(\xi(t), u(t)) \tag{3}$$

where the input is $u = [\delta_f, \sigma_r]$ and σ_r is the desired rear slip ratio.

TABLE I

TABLE OF TRIMS

trim	description	$\xi_b = [\dot{x}(m/s), \dot{y}(m/s), \dot{\psi}(r/s)]$	$[\delta_f(rad), \sigma_r]$
q_1	straight	$[11, 0, 0]$	$[0, 0.0015]$
q_2	left turn	$[11, 0.02, 0.27]$	$[0.09, 0.0015]$
q_3	right turn	$[11, -0.02, -0.27]$	$[-0.09, 0.0015]$
q_4	drift left	$[8, -1.09, 0.50]$	$[0, 0.1]$

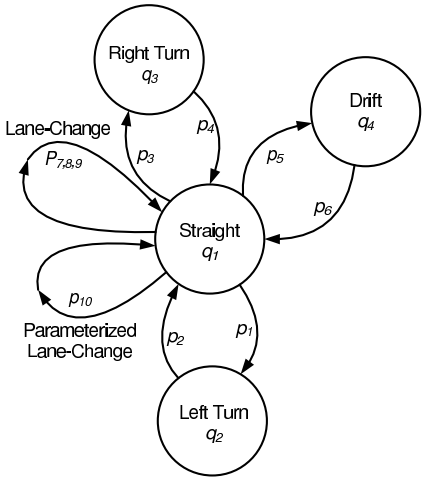

Fig. 3. An example hybrid system with steady-state trims as discrete states and maneuvers execute transitions.

III. High-Level Motion Primitive Framework

In this section we introduce the motion primitive framework. We use the ideas presented in [9] and adapt the

TABLE II

TABLE OF MANEUVERS

maneuver	description	$q_{\rm from}$	$q_{\rm to}$	$\Delta\xi_g = [\Delta X(m), \Delta Y(m), \Delta\psi(rad)]$	$\tau_p(sec.)$
p_1	straight to left turn	1	2	$[16.58, 1.61, 0.28]$	1.5
p_2	left turn to straight	2	1	$[5.53, 0.08, 0.02]$	0.5
p_3	straight to right turn	1	3	$[16.58, -1.61, -0.28]$	1.5
p_4	right turn to straight	3	1	$[5.53, -0.08, -0.02]$	0.5
p_5	straight to drift	1	4	$[17.19, 3.59, 0.77]$	2.0
p_6	drift to straight	4	1	$[17.19, 3.59, 0.77]$	2.0
p_7	right lane-change 1	1	1	$[39.67, -6.16, 0]$	3.7
p_8	right lane-change 2	1	1	$[55.14, -6.09, 0]$	5.2
p_9	left lane-change	1	1	$[87.93, 6.00, 0]$	8.1
p_{10}	parameterized lane-change	1	1	$(a, L) \leftrightarrow \Delta\xi_g$	$\tau_{(a,L)}$

formalism to fit our application. We will often sacrifice mathematical rigourousness for the sake of simplicity. Two types of motion primitives are used: trims and maneuvers.

A *trim* is a trajectory for system (3) with constant body-fixed frame states $\bar{\xi}_b^q = [\bar{\dot{x}}, \bar{\dot{y}}, \bar{\dot{\psi}}]$ and constant inputs $\bar{u}^q = [\bar{\delta_f}, \bar{\sigma_r}]$, i.e., $(\bar{\xi}_b^q, \bar{u}^q)$ is an equilibrium of the dynamical system (2a)-(2c) in closed-loop with the slip controller. By changing initial conditions of the vehicle states in the global frame $\xi_g(t_0)$, a set of trim trajectories is generated. We denote this set of trim trajectories as Ξ_q where q is used to highlight the dependence of the trim trajectories to the single equilibrium point $(\bar{\xi}_b^q, \bar{u}^q)$. In summary, a trim $\xi^q(t)$ (or simply "a trim q") is the solution to:

$$\dot{\xi^q}(t) = f_\sigma^{4w}([\xi_g(t)', \xi_b(t)']', \bar{u}^q), \ \forall t \geq t_0 \tag{4a}$$
$$\xi_b(t) = \bar{\xi}_b^q, \ \forall t \geq t_0 \tag{4b}$$
$$\xi_g(t_0) = \xi_{g,0} \tag{4c}$$

where the initial condition belongs to a compact set $\xi_{g,0} \in \mathcal{X} \times \mathcal{Y} \times \Psi$. A trim trajectory in Ξ_q requires the specification of the initial conditions $\xi_g(t_0) = [X_0, Y_0, \psi_0]$ and does not have any constraint on its duration. The set of all trims is $\bigcup_{q \in Q} \Xi_q$. Table I shows a subset of the trims used later in the examples in Section VII.

The time spent in trim q will be denoted as τ_q and will be called the *coasting time*. Since the body-fixed velocity and inputs are constant in a trim motion, the nonlinear four-wheel model (4) can be used to easily predict the state of the vehicle at time $t_0 + \tau_q$ when a trim q is initiated at time t_0:

$$\xi_g(t^+) = \xi_g(t_0) + \mathbb{R}\ \bar{\xi}_b^q \tau_q \tag{5a}$$
$$\xi_b(t^+) = \bar{\xi}_b^q \tag{5b}$$
$$q(t^+) = q(t_0) \tag{5c}$$
$$t^+ = t_0 + \tau_q \tag{5d}$$

where $\mathbb{R}$ is the rotation matrix for coordinate transformation:

$$\mathbb{R} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

A *maneuver* is a finite time trajectory which is used to transition between two trims. A maneuver p is defined by its starting and ending trim, $q_{\rm from}$ and $q_{\rm to}$, respectively, a fixed duration τ_p, a fixed displacement in the global coordinates $\Delta\xi_g^p = [\Delta X, \Delta Y, \Delta\psi]$ and the corresponding input profile $\tilde{u}^p : [t_0 \ \ t_0 + \tau_p] \to U$ where U is the set of feasible inputs.

A maneuver is executed by applying to the vehicle the input $\tilde{u}^p(t)$ for t in the interval $t_0 \leq t \leq t_0 + \tau_p$. If the vehicle is in trim $q_{\rm from}$ at time t_0, then by executing the maneuver p, the end state at time $t_0 + \tau_p$ is

$$\xi_g(t^+) = \xi_g(t_0) + \Delta\xi_g^p \tag{7a}$$
$$\xi_b(t^+) = \bar{\xi}_b^{q_{\rm to}} \tag{7b}$$
$$q(t^+) = q_{\rm to} \tag{7c}$$
$$t^+ = t_0 + \tau_p \tag{7d}$$

Transitions between trims are only permitted through the execution of a maneuver. We will use the set P to collect the index p of all stored maneuvers. Table II shows the maneuvers designed to transition between the trims in Table I.

The resulting system can be reformulated as a hybrid system with the steady-state trims as the discrete states and a maneuver used as a control input to transition between two discrete states. Figure 3 shows the hybrid system obtained by using the trims and maneuvers described in Tables I and II. With this model at hand, the motion planning problem is then formulated as a hybrid optimal control problem. In particular, finding an optimal path is reduced to finding a sequence of coasting times in trims and corresponding maneuvers that optimize a given cost function. More details can be found in [13] and [10].

IV. Motion Primitives for Collision Avoidance

The trims and maneuvers described in Tables I and II have been derived by using their definitions and forward simulations of the vehicle model (3). The intent was to generate motion primitives useful for collision avoidance of a ground vehicle. Clearly, the library of trims and maneuvers can be easily expanded.

The objective of this section is to provide details on two specific maneuvers in Table I. First, we discuss maneuver p_{10} which is a parameterized lane change obtained by using clothoids. Later in this paper we will show that this type

of maneuver is very useful for obtaining a semi-autonomous vehicle with an adjustable degree of autonomy.

Then, we discuss the drift-straight-drift maneuvers p_5 and p_6. Drifting maneuvers are used to improve the vehicle agility. Moreover, at high speed on slippery surfaces it may be the only feasible maneuver to avoid an obstacle.

1) Parameterized Clothoids: Because of the structured environment of on-road driving, a common motion primitive to avoid an oncoming collision is a lane-change [14]. A lane change motion primitive is the sequence of a straight trim, a lane-change maneuver and a straight trim. Maneuvers p_7, p_8, p_9 in Table I are classical lane change maneuvers.

Maneuver p_{10} is a parameterized lane change obtained by using four clothoids. A clothoid is a curve with a curvature that changes linearly with the curve length. It is widely used in railway and high way engineering for curvature transition. Also, its use in road design and robot path planning has a long history. Relevant references include [15], [16], [17]. In a clothoid, at a given curve length s the curvature $K(s)$ is determined by the linear function $K(s) = 2a^2 \cdot s$, where a is the curvature change rate. The clothoid is expressed by the Fresnel integral [15] as:

$$\begin{aligned} X &= \tfrac{1}{a}\int_0^{aL} \cos(\mathrm{s}^2)\mathrm{ds} \\ Y &= \tfrac{1}{a}\int_0^{aL} \sin(\mathrm{s}^2)\mathrm{ds} \end{aligned} \tag{8}$$

where X and Y are the global coordinates and L is the maximum curve length. In numerical computation schemes, the integrals in Equation (8) are approximated by their Taylor expansion:

$$\begin{aligned} \int_0^{aL} \cos(\mathrm{s}^2)\mathrm{ds} &= aL - \tfrac{(aL)^5}{5\times 2!} + \tfrac{(aL)^9}{9\times 4!} - \tfrac{(aL)^{13}}{13\times 6!}\cdots \\ \int_0^{aL} \sin(\mathrm{s}^2)\mathrm{ds} &= \tfrac{(aL)^3}{3} - \tfrac{(aL)^7}{7\times 3!} + \tfrac{(aL)^{11}}{11\times 5!} - \tfrac{(aL)^{15}}{15\times 7!}\cdots \end{aligned} \tag{9}$$

By varying the parameters a and L, different clothoids are generated. A lane-change maneuver is built by connecting 4 pieces of clothoids together, as shown in Figure 4.

In our approach, in order to avoid storing lane change maneuvers for different a and L, we do not pre-compute the maneuver p_{10}. Instead, we find conditions on a and L guaranteeing that the maneuver is feasible and let the path planner find the optimal and feasible a and L. The maximum curvature of the clothoid defined by Equation (8) is $2a^2L$. Thus, the maximum lateral force needed to track the clothoid is $m\dot{x}^2 \cdot 2a^2L$. Thus, by constraining a and L we can guarantee that the required lateral force is feasible, and therefore, that the clothoid can be followed without tracking error.

For a point-mass model, the upper bound on lateral force is μmg. Thus, the constraint on curvature for a point-mass model is $2a^2L \leq \frac{\mu g}{\dot{x}^2}$. For the four-wheel model described in section II, the maximum achievable lateral force depends on the current vehicle states. At the beginning of a turn where

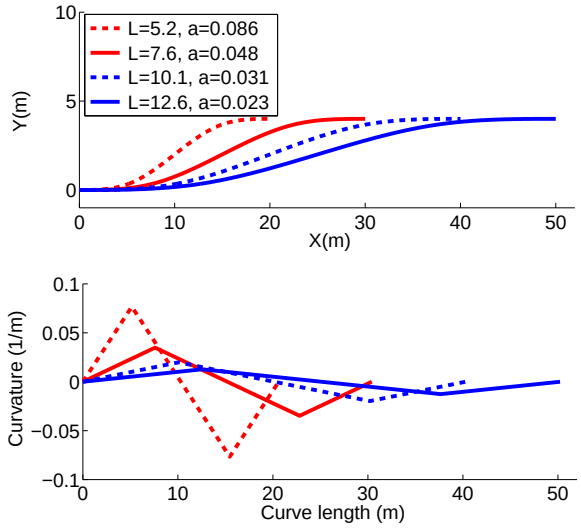

Fig. 4. Lane change paths with different aggressiveness. Each is generated by connecting four pieces of clothoids. The upper figure shows the shapes of the paths in global frame. The lower figure shows the piecewise affine relation between curvature and curve length for each path.

lateral velocity v_y and yaw rate $\dot{\psi}$ are both zero, the lateral force at the rear tires are zero, therefore the only source of lateral force at that instant are the front tires. We use sampling and extensive simulations of our vehicle model to determine the maximum available lateral force at the beginning of a turn ($0.46\mu mg$). After the turn begins, v_y and $\dot{\psi}$ increase and more lateral force will become available. For a conservative approximation, the value $0.46\mu mg$ is taken as the upper bound of lateral force along the turn. The constraint on a and L for the clothoid becomes:

$$2a^2L \leq \frac{0.46\mu g}{\dot{x}^2} \tag{10}$$

where μ is the friction coefficient and $\dot{x}$ is the body-fixed longitudinal velocity. Specifying a and L determines $\Delta\xi_g^{p_{10}}$ and the time to track the clothoid $\tau_{a,L}$ depends on the low-level MPC.

2) Drift Trim and Maneuver: Maneuvers p_5 and p_6 are designed to bring the vehicle into a steady-state drift. During a drifting trim the tires are operating outside their linear region (see Figure 6) and the constant side slip angle of the vehicle is high. This is commonly achieved by saturating the rear tire forces.

Bringing the system to a steady-state drift is a maneuver often difficult to achieve since the drifting trim is an unstable equilibrium point of the system (2). We compute the equilibrium points as in [18] by solving a set of nonlinear equations, $\dot{\xi} = f_\sigma^{4w}(\xi_e, u_e) = 0$ for constant ξ_e and u_e. Once a drift trim has been computed, we compute the drifting maneuvers off-line by solving a sequence of nonlinear optimization problems to find a trajectory in the body-frame that lands in the desired steady-state point.

Computing offline drifting maneuvers allows for the path

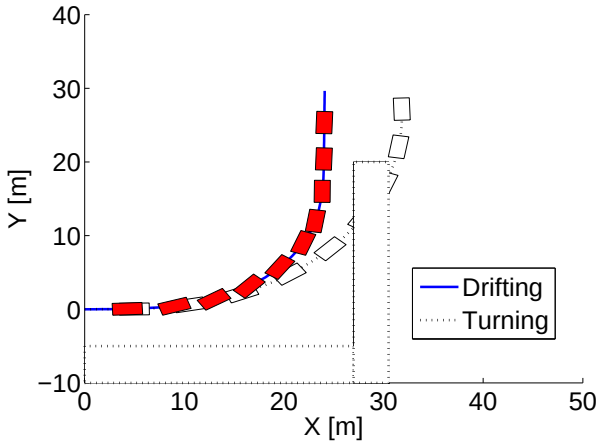

Fig. 5. A drifting maneuver compared to an aggressive turn. The turn is the sharpest allowable maneuver within the constraint of the slip angles of the tires. At this initial vehicle state ($\xi_g = [0, 0, 0],\ \dot{x} = 11\text{m/s},\ q_1$) the drifting maneuver is the only safe maneuver to avoid the obstacle. $\mu = 0.55$.

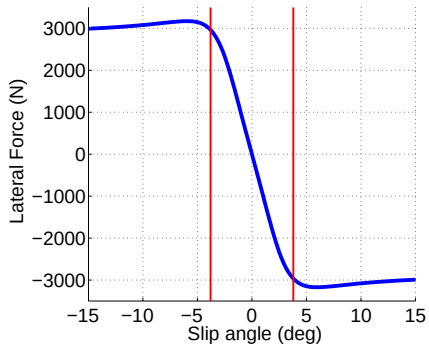

Fig. 6. The linear region of the tire model is considered the operational envelope where the vehicle maneuvers are stable.

planner to quickly plan a feasible path which can be difficult to find in real-time with online nonlinear optimization.

In Figure 5, the vehicle approaches an obstacle at $\dot{x} = 11\ \text{m/s}$ on an icy surface with $\mu = 0.55$. In these conditions, the drifting maneuver may be the only safe maneuver to navigate the corner.

V. Planning With Motion Primitives

Casting the trajectory planning problem in the motion primitive framework allows us to compute an optimal feasible path by solving a hybrid optimal control problem. It is our desire to track a reference trajectory while avoiding collisions with any obstacles. We use a cost function that is a weighted sum of the tracking error, distance to any obstacles, and the *aggressiveness* of the maneuver measured as a norm of the maximum side slip angle.

Let the cost of a trim q be defined as:

$$C_q = K_r \int_{t_0}^{T_q} (\eta(t) - \eta_{ref}(t))^2 dt + K_o \frac{\dot{X}_{T_q}}{d(T_q) + \epsilon} \quad (11)$$

where $\eta = \left[\dot{x}, \psi, \dot{\psi}, Y\right]$ and η_{ref} is the reference trajectory. K_r and K_o are weighting scalars. $T_q = t_0 + \tau_q$ where trim q is initiated at time t_0 and τ_q is the time spent in trim q. $\dot{X}_{T_q}$ is the vehicle speed in the inertial X-direction at time T_q, $d(T_q)$ is the distance to the obstacle at time T_q (see [8] for the explicit calculation of this term), and ϵ is a small constant. The cost of initiating a maneuver p is defined as:

$$C_p = K_r^m \int_{t_0}^{T_p} (\eta(t) - \eta_{ref}(t))^2 dt + K_p^m \|\alpha_{\max}\| \quad (12)$$

where K_r^m and K_p^m are weighting scalars. $T_p = t_0 + \tau_p$ where t_0 is the start time of the maneuver p and τ_p is the duration of maneuver p. $\alpha_{\max}$ is the maximum slip angle of the front tires during the maneuver. Higher weights K_p^m on $\alpha_{\max}$ favor paths that are less aggressive.

The problem of finding a sequence of maneuvers and trims minimizing the cost $C_p + C_q$ in (11) and (12) can be formulated as a dynamic program:

$$J(\xi_g, q) = \min_{\tau_q, p, (a,L)} \left[C_q + C_p + J(\xi_g^+, q^+)\right] \quad (13)$$

where ξ_g^+ and q^+ are the next vehicle states and trim obtained by taking maneuver p, as defined in Equations (5) and (7), respectively. In (13) $\xi_g = [X, Y, \psi]$, $(\xi_g(0), q(0)) = (\xi_{g,0}, q_0)$ and $J(\xi_{g,f}, q_f) = 0$. τ_q is the time spent in trim q and p is the maneuver executed. For the parameterized clothoid the minimization is also taken over parameters a and L. We solve (13) offline by discretizing the value function over a grid in the (X, Y, ψ, q) space.

VI. Low-Level MPC Path Follower

The trajectory computed by the high-level motion-primitive path planner is sent to the low level path follower. A Model Predictive Controller (MPC) is used to accurately track the path and thus mitigate the effect of model mismatch and exogenous disturbances. We discretize model (3) to obtain

$$\xi_{k+1} = f_\sigma^{4w}(\xi_k, u_k) \quad (14)$$

The MPC controller solves the following optimization problem:

$$\min_{U_t} J_N(\tilde{\xi}_t, U_t, \Delta U_t) \quad (15a)$$

$$\text{subj. to} \quad \xi_{k+1,t} = f_\sigma^{4w}(\xi_{k,t}, u_{k,t}) \quad k = t, ..., t + H_p - 1 \quad (15b)$$

$$\Delta u_{k+1,t} = u_{k+1,t} - u_{k,t} \quad k = t, ..., t + H_p - 1 \quad (15c)$$

$$u_{k,t} \in \mathcal{U} \quad k = t, ..., t + H_p - 1 \quad (15d)$$

$$\Delta u_{k,t} \in \Delta\mathcal{U} \quad k = t + 1, ..., t + H_p - 1 \quad (15e)$$

$$\xi_{t,t} = \xi(t) \quad (15f)$$

where $\tilde{\xi}_t = [\xi_{t,t}\ ,\ \xi_{t+1,t}\ , ...,\ \xi_{t+H_p-1,t}]$ is the sequence of states $\tilde{\xi}_t \in \mathbb{R}^{nH_p}$ over the prediction horizon H_p predicted at time t, and updated according to the discretized dynamics of the vehicle model (15b), and $u_{k,t}$ and $\Delta u_{k,t} \in \mathbb{R}^{m_r}$ is the

k^{th} vector of the input sequence $U_t \in \mathbb{R}^{m_r H_p}$ and $\Delta U_t \in \mathbb{R}^{m_r(H_p-1)}$ respectively,

$$U_t = [u'_{t,t} \ , \ u'_{t+1,t} \ , ..., \ u'_{t+H_u-1,t} \ , \ u'_{t+H_u,t} \ , ..., \ u'_{t+H_p-1,t}]' \tag{16a}$$

$$\Delta U_t = [\Delta u'_{t+1,t}, ..., \Delta u'_{t+H_u-1,t}, \Delta u'_{t+H_u,t}, ..., \Delta u'_{t+H_p-1,t}]' \tag{16b}$$

The cost function to be minimized is

$$J_N(\bar{\xi}_t, U_t, \Delta U_t) = \sum_{k=t}^{t+H_p-1} \|\eta_{k,t} - \eta_{\mathrm{plan_{k,t}}}\|^2_Q + \|u_{k,t}\|^2_R + \|\Delta u_{k,t}\|^2_S \tag{17}$$

where the tracking reference is the output of the motion primitive planner and is the planned path:

$$\eta_{\mathrm{plan_{k,t}}} = [\dot{x}_{\mathrm{plan_{k,t}}}, \psi_{\mathrm{plan_{k,t}}}, \dot{\psi}_{\mathrm{plan_{k,t}}}, Y_{\mathrm{plan_{k,t}}}]' \tag{18}$$

At each time step t, the performance index $J_N(\tilde{\xi}_t, U_t, \Delta U_t)$ is optimized under the constraints (15c)-(15e) starting from the state $\xi_{t,t} = \xi(t)$ to obtain an optimal control sequence, $U_t^* = [u_{t,t}^{*\prime}, ..., u_{t+H_p-1,t}^{*\prime}]'$. The first of such optimal moves $u_{t,t}^*$ is the control action applied to the vehicle at time t. At time $t+1$, a new optimization is solved over a shifted prediction horizon starting from the new measured state $\xi_{t+1,t+1} = \xi(t+1)$.

VII. Simulation And Experimental Results

Extensive simulations and experiments have been performed to test the proposed hierarchical control architecture, composed of a high-level motion primitive planner and a low-level MPC path follower. This section has two objectives: first we will show in simulation the capability of the proposed scheme in generating aggressive feasible motion plans, which include drifting in order to avoid lane departures. Then, we will show the benefit of using parameterized clothoids in semi-autonomus drive, by using simulations and experiments conducted on a vehicle while driving high-speed on ice.

A. Agile Driving

The simulations presented in this section show the utility of the motion primitive planner in its capability of generating aggressive feasible trajectories. The simulated scenario shows the vehicle approaching a right-angle at a velocity of $\dot{x} = 11$ m/s in trim q_1 on an icy surface with $\mu = 0.55$. Figure 7 shows the results of the motion primitive planner and the low-level tracking. In the simulation the target set is $X_f \in [32\ 37]$ and $Y_f \in [38\ 42]$. Using the nonlinear solver NPSOL [19] a feasible solution could not be found. Without considering the drifting primitives, the lane departure is unavoidable with steady-state turning.

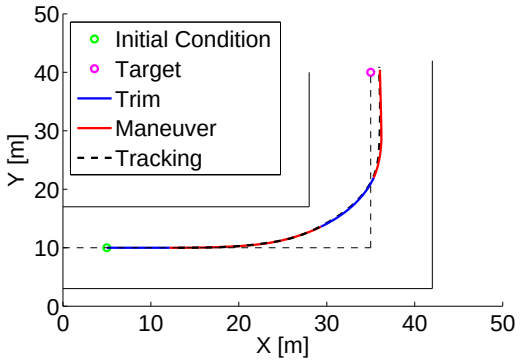

Fig. 7. A successfull navigation of the corner planned with the drifting primitives. The vehicle starts in trim q_1 (straight travel, $\dot{x} = 11$m/s), takes maneuver p_5 to a drifting trim q_4, then takes maneuver p_6 back to q_1. The aggressive path is able to be tracked with no tracking error. $\mu = 0.55$.

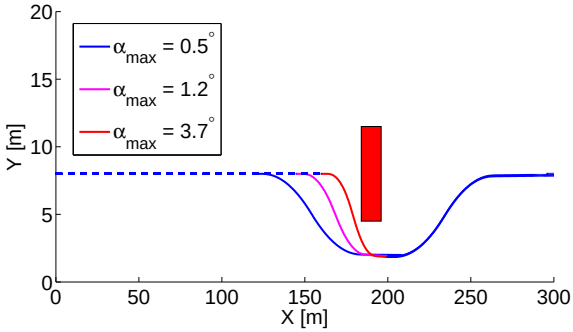

Fig. 8. Various lane change maneuvers are compared. As the vehicle approaches the obstacle the planned paths become more aggressive (high slip angles).

B. Semi-Autonomous Drive - Simulations

The simulations presented in this section are based on single and double lane-change maneuvers to avoid an upcoming obstacle. The vehicle is initially tracking a reference trajectory down the center of the lane. The high-level path planner is generating trajectories at each time step based on the current vehicle position. The autonomous system does not take over from the driver until the aggressiveness of the planned path, defined by its maximum slip angle $\alpha_{\max}$, is greater than a specified threshold.

We use Matlab to simulate the closed-loop system. The MPC optimization problem has been implemented as a C-coded s-Function. The commercial NPSOL software package [19] is used for solving the nonlinear programming problem in Equation (15). The first element of the optimized control sequence is passed to an external block which uses the four-wheel model and Pacejka tire model to simulate the dynamics of the vehicle, and feeds the current state of the vehicle back to the controller.

Figure 8 shows three different lane change maneuvers. They are generated from the high-level path planner starting in a steady-state trim q_1 and $X_0 = 120,\ 140,$ and 160m. As the vehicle approachs the obstacle, the planned paths become

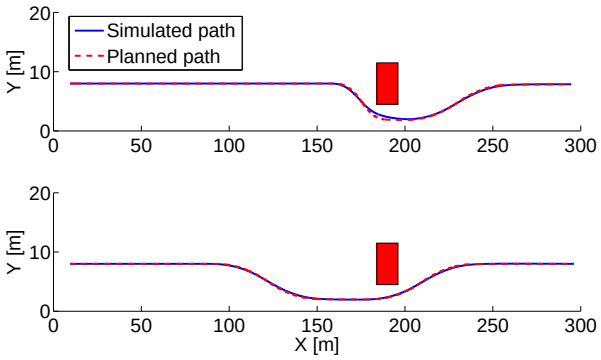

Fig. 9. In the upper plot an attentive driver is assumed. The low-level control takes over when the planned path becomes aggressive. In the lower plot the low-level control takes over for a distracted driver and the result is a smoother and safer path.

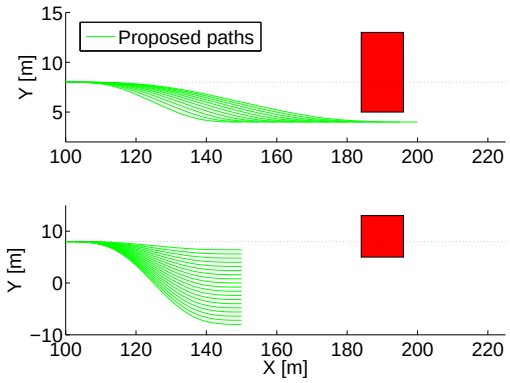

Fig. 10. Various candidate paths considered during the online optimization of the parameterized maneuver.

more aggressive, that is, the maximum slip angle of the front tires approaches the stability limit for the vehicle. The maneuver with $\alpha_{\max} = 3.7°$ is the most aggressive maneuver the vehicle can handle at the given speed. The simulations presented in Figure 9 use a binary driver model. The driver is either attentive, as in the upper plot of Figure 9, and the low-level controller does not assume control until the planned path becomes aggressive, or the driver is distracted, as in the lower plot of Figure 9, and the low-level control begins tracking the planned path with a lower degree of curvature. Because the planned path is generated using the high fidelity four-wheel model, both simulations show accurate tracking by the low-level MPC contoller.

Figure 10 shows some of the candidate paths considered during the online optimization of the parameterized clothoid maneuver, where the full set are all possible combinations of the paths in the upper and lower plots of Figure 10. The candidate path chosen is the one that minimizes the cost function in Equation (13) and satisfies the constraint in (10). Section VII-C below details an experiment conducted on ice for the online optimization of this maneuver.

C. Semi-Autonomous Drive - Experiments

The control framework presented has been tested through simulation and the online optimization of parameterized maneuvers has been tested through experiments conducted on an icy surface. The experiments have been performed at a test center equipped with icy and snowy handling tracks. The MPC controllers have been tested on a passenger car with a mass of $2050\ \mathrm{Kg}$ and an intertia of $3344\ \mathrm{Kg/m^2}$. The controllers were run in a dSPACE Autobox system equipped with a DS1005 processor board and a DS2210 I/O board.

We used an Oxford Technical Solution (OTS) RT3002 sensing system to measure the position and the orientation of the vehicle in the inertial frame and the vehicle velocities in the vehicle body-frame. The OTS RT3002 is housed in a small package that contains differential GPS receiver, inertial measurement unit (IMU), and a DSP. It is equipped with a single antenna to receive GPS information. The IMU inclused three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMP and GPS, utilizes a Kalman filter for sensor fusion, and calculates the position, orientation, and other states of the vehicle such as lateral and longitudinal velocities.

The car is equipped with an Active Front Steering (AFS) and Differential Braking system which utilizes an electric drive motor to change the relation between the hand steering wheel and the road wheel angles. This is done independently from the hand wheel position, thus the front road wheel angle is obstained by summing the driver hand wheel position and the actuator angular movement. Both the hand wheel position and the angular relation between hand and road wheels are measured. The sensor, the dSPACE Autobox, and the actuators communicate through a CAN bus.

The proposed control framework was tested with the high-level motion primitive path planner using parameterized clothoids to construct the path. The test was run on an icy and slippery test track. In the first test shown in Figure 11 the driver was assumed to be distracted and the path tracker assumed control early enough to track a smoothly planned trajectory. In the second experiment in Figure 12 the driver was assumed to be attentive and the controller did not take over until the planned path became aggressive. Both experiments successfully avoided the obstacle.

Note that the low-level path follower might not be able to track the planned path perfectly because of model mismatch and external disturbances. In some cases, the tracking error is large enough that the maneuver becomes infeasible to track. The second experiment in Figure 12 shows this interesting behavior. In the test, when infeasibility appeared, a braking maneuver was invoked to reduce the velocity and thus enlarge the feasible region. In Fig 12 we can see that after a few seconds of braking the path planner is able to find a feasible solution around the obstacle, braking is interrupted and the obstacle is successfully avoided.

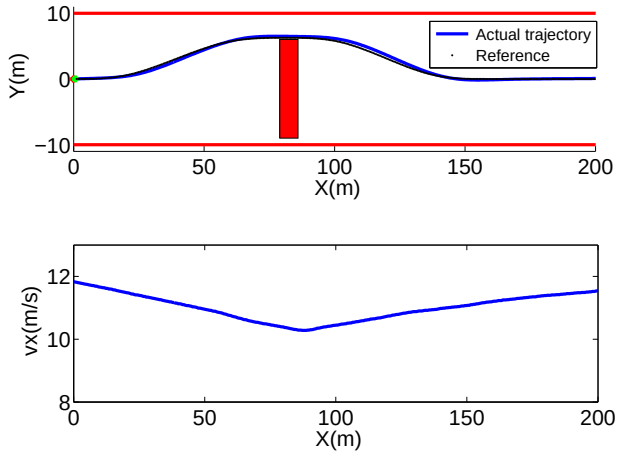

Fig. 11. Vehicle successfully avoids the obstacle using maneuvers based on clothoids

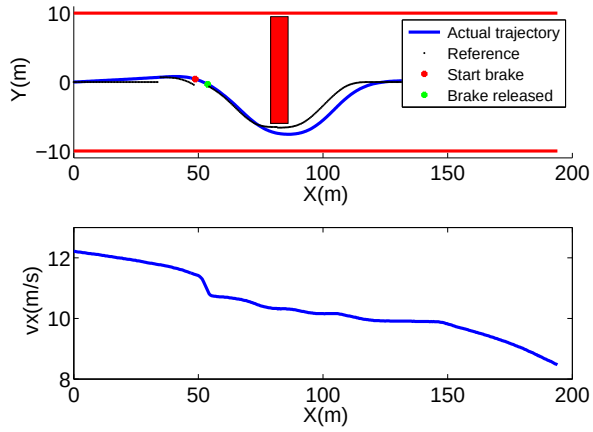

Fig. 12. Actual path of the vehicle deviates from the planned path due to model mismatch and caused infeasibility of tracking. Braking was invoked to enlarge the feasible region in that situation.

VIII. CONCLUSIONS

We have presented a hierarchical control framework used for autonomous and semi-autonomous guidance of ground vehicles. The high-level planner is based on motion primitives created from a four-wheel nonlinear dynamic model. The low-level tracks the planned trajectory with a nonlinear Model Predictive Controller.

The proposed scheme has several advantages. First, in a structured environment, selecting feasible motion primitives might be faster than using real-time nonlinear optimization. Second, the use of parameterized clothoids can be easily integrated in a semi-autonomus system where the driver distraction is quantified by using auxiliary vehicle sensors (such as in-vehicle cameras). Simulation and experimental tests show the effectiveness of the approach and interesting emerging behaviors.

REFERENCES

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Low complexity mpc schemes for integrated vehicle dynamics control problems," 9^{th} *International Symposium on Advanced Vehicle Control*, 2008.

[2] F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, "An MPC/hybrid system approach to traction control," *IEEE Trans. Control Systems Technology*, vol. 14, no. 3, pp. 541–552, May 2006.

[3] F. Borrelli, T. Keviczky, G. J. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science. Springer Verlag, Mar. 2005.

[4] T. Keviczky and G. J. Balas, "Flight test of a receding horizon controller for autonomous uav guidance," in *Proc. American Contr. Conf.*, 2005.

[5] H. J. Ferrau, H. G. Bock, and M. Diehl, "An online active set strategy for fast parametric quadtratic programming in mpc applications," *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk*, 2006.

[6] V. M. Zavala, C. D. Laird, and L. T. Biegler, "Fast solvers and rigorous models: Can both be accomodated in nmpc?" *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems, plenary talk*, 2006.

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

[8] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," *Dynamic Systems and Control Conference, 2010*, 2010.

[9] E. Frazzoli, "Robust hybrid control for autonomous vehicle motion planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[10] Frazzoli, "Maneuver-based motion planning for nonlinear systems with symmetries," vol. 21, Dec. 2005, pp. 1077–1091.

[11] C. Neas and M. Farhood, "A hybrid architecture for maneuver-based motion planning and control of agile vehicles," *IFAC, Internationl Federation of Automatic Control*, vol. 18, 2011.

[12] E. Bakker, L. Nyborg, and H. B. Pacejka, "Tyre modeling for use in vehicle dynamics studies," *SAE paper # 870421*, 1987.

[13] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 116–129, 2002.

[14] R. Hanowski, R. Olson, J. Hickman, and T. Dingus, "The 100-car naturalistic driving study: A descriptive analysis of light vehicle-heavy vehicle interactions from the light vehicle drivers perspective," Virginia Tech Transportation Institute, 3500 Transportation Research Plaza (0536) Blacksburg, Virginia 24061, Tech. Rep. FMCSA-RRR-06-004, Mar. 2006.

[15] D. H. Shin and S. Singh, "Path generation for robot vehicles using composite clothoid segments," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-90-31, dec 1990.

[16] A. Scheuer and T. Fraichard, "Collision-free and continuous-curvat ure path planning for car-like robots," *IEEE Intemational Conference on Robotics and Automation*, April 1997.

[17] Levien and Raph, "The euler spiral: a mathematical history," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-111, sep 2008.

[18] R. Y. Hindiyeh and J. C. Gerdes, "Equilibrium analysis of drifting vehicles for control design," *Dynamic Systems and Control Conference*, Oct. 2009.

[19] P. Gill, W. Murray, M. Saunders, and M. Wright, NPSOL – Nonlinear Programming Software, Stanford Business Software, Inc., Mountain View, CA, 1998.