2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Optimal Decision Making for Automated Vehicles Using Homotopy Generation and Nonlinear Model Predictive Control

Vivian Z. Patterson[1], Francis E. Lewis[2], and J. Christian Gerdes[1]

Abstract— **To navigate complex driving scenarios, automated vehicles must be able to make decisions that reflect higher-level goals such as safety and efficiency, leveraging the vehicle's full capabilities if necessary. We introduce an architecture that is capable of handling combinatorial decision making and control with a high fidelity vehicle model. This is accomplished by solving a nonlinear model predictive control optimization for each maneuver variant, or homotopy, identified in the drivable space. These locally optimal solutions are then evaluated on a criterion that reflects high-level objectives. Experimental results on a full-scale vehicle demonstrate this architecture's effectiveness in an overtaking scenario with oncoming traffic that requires the ego vehicle to decide whether to pass before or after the oncoming traffic passes.**

I. INTRODUCTION

Automated vehicles require both the ability to make intelligent decisions as well as successfully execute the chosen maneuver. Model predictive control (MPC) is a popular and effective technique that allows vehicles to respond to an evolving environment by using fast replanning. Nonlinear MPC (NMPC) in particular can incorporate more precise, nonlinear dynamics and has recently been shown to work experimentally on full-scale vehicles in a number of scenarios. Gao *et al.* use a two-level hierarchy to avoid obstacles on an icy surface [1]. Laurense and Gerdes leverage vehicle models with different fidelities to achieve high performance in racing scenarios [2]. Brown and Gerdes coordinate longitudinal and lateral forces to execute a emergency double lane change to avoid a pop-up obstacle [3].

In these papers, the desired trajectory is obstacle-free either by construction of the scenario or by assuming the existence of an upstream planner. However, this is a key simplification: the presence of obstacles creates a combinatorial problem with multiple possible local minima, such as passing on the left or the right side. A non-convex, gradient-based optimization such as NMPC is not guaranteed to find a global optimum, and the solution is contingent on how the problem is initialized. NMPC provides an effective control scheme but in isolation lacks the decision-making capability necessary for a full self-driving vehicle.

In the literature, decision-making falls to trajectory planners, which can be loosely grouped into combinatorial and sampling-based methods. Sampling-based planners intrinsically consider different maneuvers around obstacles and work well in practice [4]. However, these methods may not yield a trajectory that is in the same maneuver class that contains the globally optimal solution. Moreover, even if the planner happens to find the best class, the solution may still be suboptimal. Therefore, for the purposes of this paper, we will focus on combinatorial methods.

Researchers have methods to systematically partition the environment around obstacles into discrete driving options, using homotopy classes [5], maneuver variants [6], and driving corridors [7] [8]. The existence of homotopies greatly increases the complexity of the problem. Some formulations attempt to solve the decision-making and control problem without further subdivision or abstraction. However, Park *et al.* demonstrate that undecomposed trajectory planning around several obstacles yielded far longer solve times than a divide-and-conquer strategy [5]. Cunningham *et al.* simulate the ego vehicle and surrounding vehicles using several plausible, pre-computed policies, then score the simulations to determine the best maneuver [9]. This strategy considers a fixed number of policies and is not yet able to adapt to quick changes in system state according to the authors.

Other architectures deal with this complexity by partitioning the problem: first solving a decision-making problem using a high-level trajectory planner, then using a low-level trajectory follower to execute the plan. Many combinatorial planners use reasonable heuristics to determine which homotopy is most desirable to pursue prior to running a low-level controller. Ziegler *et al.* select a single homotopy by passing moving obstacles according to road conventions and the direction of travel [7]. They then generate a path that is tracked with feedback control. Plessen *et al.* choose a path that minimizes heading variation before using linear time-varying MPC to track the desired trajectory [10]. Liniger *et al.* use a high-level planner that solves for the spatially shortest path using dynamic programming. They approximate the NMPC problem by linearizing around a state trajectory and apply it to racing with obstacle avoidance [11]. All architectures mentioned above follow the same general method – generate one or more obstacle-free trajectories, select the best one based on an evaluation metric, then execute the plan with fast lower-level controls as closely as possible (Fig. 1, top). However, the maneuver is chosen without considering the full capability of the controller, which can ultimately lead to an optimal solution in a suboptimal homotopy.

Some architectures attempt to compare solutions derived with the underlying controller across different homotopies. Erlien develops a convex MPC formulation for shared steering control near vehicle stability limits, assuming constant longitudinal velocity [12]. The controller enumerates all

This work was supported by the Ford-Stanford Alliance.

[1]Department of Mechanical Engineering, Stanford University, Stanford, CA zhangv@stanford.edu

[2]Department of Computer Science, Stanford University, Stanford, CA

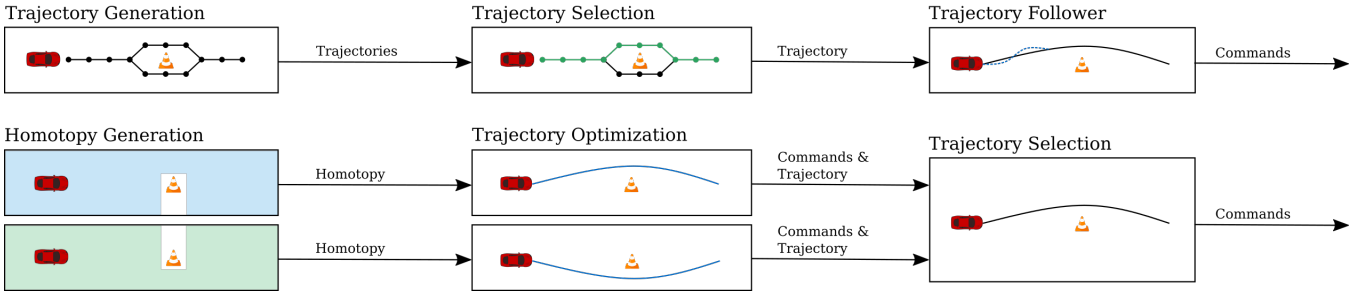

Fig. 1. Block diagrams for a traditional decision-making then control architecture (top) compared to an integrated architecture, presented in this paper (bottom). In our proposed system, the output of a high fidelity controller is evaluated.

possible left-right options around static obstacles, serially determines the optimal trajectory for each option, and applies the input associated with the lowest cost trajectory. Park *et al.* utilize a similar architecture, using cell decomposition methods and passing relevant homotopy boundary information to a mixed-integer quadratic program (MIQP) [5]. They demonstrate how the MIQP could be incorporated in a linear time-varying MPC optimization and show results in simulation. However, the ability to understand and control the highly-coupled lateral and longitudinal dynamics is critical at all levels of driving. Especially closer to the limits of friction, tire and vehicle model behaviors become increasingly nonlinear and difficult to approximate accurately using linearized systems. NMPC is a natural extension, as it can handle more expressive models and constraints.

We present a homotopy-based architecture for structured environments capable of making high-level decisions regarding obstacles while employing a full nonlinear model of the vehicle's capabilities. The overview of the structure can be seen in Fig. 1 (bottom). Drawing inspiration from Erlien and Park, our proposed strategy includes leveraging the road structure to quickly construct possible driving homotopies, defined in detail in Sec. II. Then, parallel NMPC optimizations described in Sec. III are solved to find locally optimal trajectories. Finally, the trajectory that best satisfies goals such as safety and mobility highlighted in Sec. IV is selected. Finally, the efficacy of this controller is demonstrated experimentally on a full-scale vehicle, and the results are presented in Sec. V.

II. HOMOTOPY GENERATION

The drivable space in structured environments is represented relative to a road. A road is defined in global coordinates by a descriptor line (for example, the centerline) and road edges, which are obtained from maps a priori. Fig. 2 shows the curvilinear coordinate system defined by the distance s along this road descriptor line and lateral position e perpendicular to the line used to describe the position states of an obstacle or vehicle. Separating a scene into possible homotopies relies on a having a prediction of future obstacle motion. We assume that we receive the output of a perception and motion prediction algorithm and regard the information as deterministic. While obstacle accelerations are currently assumed to be constant, this architecture can leverage more sophisticated prediction models parameterized by time.

A. Definition

A homotopy H_i is the set of longitudinal and lateral box constraints defined by

$$H_i(t,s) := E_i(t,s) \times S_i(t) \tag{1}$$

where $E_i(t,s)$ is the spatiotemporal region that the ego vehicle is constrained to be within

$$E_i(t,s) := \{e \mid e_{\min,i}(t,s) \leq e \leq e_{\max,i}(t,s)\}, \tag{2}$$

and $S_i(t)$ is the target longitudinal position of the ego vehicle relative to nearby obstacles

$$S_i(t) := \{s \mid s_{\min,i}(t) \leq s \leq s_{\max,i}(t)\}. \tag{3}$$

These constraints may arise from static road elements such as stationary obstacles or road geometry, or from dynamic road elements such as moving obstacles. Distinct homotopy classes must differ in at least one of the constraints. For any single obstacle, we will use the terms "left" and "right" denoted $H_{\{l,r\}}$ for differentiating the options for (2), and "pass" and "follow" or $H_{\{p,f\}}$ respectively for (3). A scenario with no obstacle within a specified distance is deemed a "neutral" homotopy H_n.

Both conditions are necessary to effectively find different maneuver variants. In a scenario with a lead vehicle in front of the ego vehicle in the same lane, a passing homotopy with only the lateral bound constraint (2) is not guaranteed to actually pass the vehicle. Without (3), the optimization could yield solutions where the ego vehicle is in front, to the side, or even behind the other vehicle, degenerating into

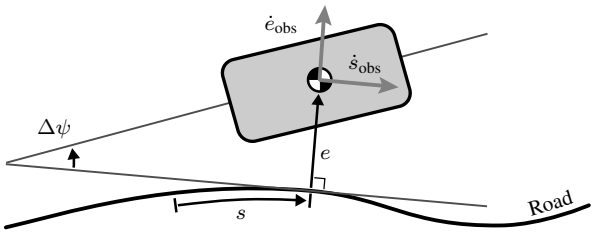

Fig. 2. Rectangular obstacle on a road in path coordinates.

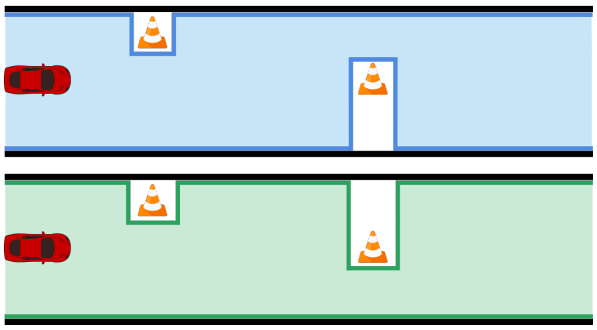

Fig. 3. A scenario with two static obstacles with two possible homotopies. $H_l(s)$ passes the center obstacle on the left (top) and $H_r(s)$ passes on the right (bottom). The lateral bounds $E_i(s)$ are not a function of t.

a following maneuver. Constraining the solutions to different homotopies guarantees different outcomes to be explored.

We assume that the vehicle with position states $x = [s, e]^\top$ will be monotonically increasing in s, which is appropriate for structured roads. Then for all finite trajectories $\vec{x}$ within a single homotopy, each trajectory must satisfy

$$e(t, s) \in E_i(t, s) \qquad \forall t \in [t_0, t_0 + T] \quad (4)$$

$$s(t_0 + T) \in S_i(t_0 + T) \quad (5)$$

Equation (5) is only active at the terminal state of a trajectory. This definition does not require the end state of all trajectories $x(t_0 + T)$ within a single homotopy to be equal, which differs from the topographic definition.

B. Implementation Details

The homotopy generation module identifies potential homotopies, outputting static lateral boundaries and a label to indicate the homotopy type. This label is used in the NMPC formulation to set spatiotemporal constraints.

1) Static Obstacles: We calculate static (non-time dependent) homotopy boundaries by using the road edges augmented with static obstacles. For example, if an obstacle is on the right side of the road, the right boundary will be modified to exclude the obstacle from the safe driving region. If the obstacle can be passed on either side safely, this will yield H_l and H_r, as shown in Fig. 3. The vehicle's monotonic progress in s lends itself well to using trapezoidal decomposition, as described by LaValle in [13] and utilized by Park *et al.* in [5]. In this implementation, a line is swept across the s dimension in the driving space and decomposes the space into distinct cells based on obstacle locations. Then, a directed acyclic graph is constructed from the cells and traversed to build all possible homotopies.

2) Moving Obstacles: When the ego vehicle approaches a moving obstacle ahead of it in the same lane, the homotopy layer generates both H_p and H_f, replacing the neutral option H_n. We can calculate future obstacle positions to determine the parameters for $S_i(t_0 + T)$, as highlighted in Fig. 4. This 3D representation of the drivable space is inspired by the visualization of maneuver variants in Bender *et al* [6].

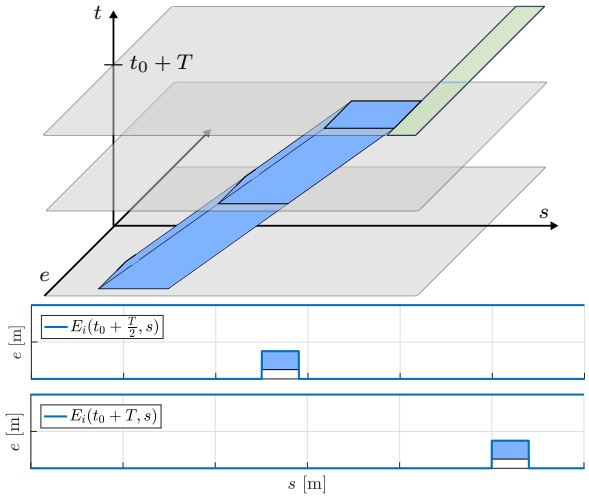

Fig. 4. By evaluating obstacle motion at discrete slices in time t (top), $E_i(t, s)$ can be represented as functions of s and e. The middle and bottom figures show the obstacle with $E_i(t_0 + \frac{T}{2}, s)$ and $E_i(t_0 + T, s)$ boundaries, respectively. The region $S_p(t_0 + T)$ is shaded in green.

Representing the spatiotemporal lateral bounds for a continuous, multivariable function $E_i(t, s)$ is less straightforward. If evaluated at a particular time t_j, the function can be collapsed into a univariable function $E_i(s)$. However, since the vehicle's longitudinal motion is variable and determined in the downstream NMPC optimization, it is impossible to predict the exact time (or location) at which the vehicle will overtake the obstacle at the homotopy generation level. Successful obstacle avoidance will necessarily be a function of both obstacle and vehicle motion.

Instead, the homotopy generation system attaches a label indicating the homotopy type – one of "neutral", "passing", or "following". Then at the NMPC level (described in Sec. III), a prediction horizon of N discrete points in time into the future is chosen prior to the optimization. We use these points in evaluating $E_i(t, s)$ to produce $E_i(t_k, s), \forall k \in [0, N]$ as shown in Fig. 4. These constraints are explicitly a function of obstacle motion and implicitly a function of time, and can set the appropriate terms in the NMPC optimization.

3) Other Details: This method of homotopy generation produces an exponential number of possibilities with respect to the number of obstacles. However, that number remains small since the architecture only considers the obstacles that are within a reasonable passing distance. Additionally, infeasible or undesirable options can be pre-pruned, which is a viable strategy found in many decision-making schemes that can be employed here as well.

After identifying possible homotopies, a nominal trajectory is generated within each option, consisting of lateral positions e_{nom} and speed profile $u_{x,\text{des}}$ parameterized by s. Combined with small tracking weights in the NMPC cost function, we can encourage specific driving styles while retaining the flexibility that NMPC has to make performance optimizations and tradeoffs in real time. This nominal trajectory can be generated using a variety of existing planning

strategies in the literature, highlighting the modularity and versatility of this hierarchical architecture.

III. NONLINEAR MPC FORMULATION

An NMPC optimization is run in each homotopy in parallel. It uses a planar dynamic single-track vehicle model and solves for optimal inputs over a prediction horizon of length N. The vehicle state x includes the position states s and e introduced in Sec. II, and the inputs are steer angle δ and total longitudinal force F_x, or $u = [\delta, F_x]^\top$. Additional details on the full vehicle model can be found in [3].

The homotopy type not only informs what constraints are active but also which terms in the cost function are present. This is an advantage to have tailored objectives — a passing and a following maneuver have different goals in terms of longitudinal velocity. The specific differences are discussed in the following sections.

A. Objective Function

The objective function reflects the goals based on the type of homotopy. Shared goals include smoothness of input rates $\dot{\delta}$ and $\dot{F}_x$ and tracking the nominal lateral position generated by the homotopy layer.

$$J_{\dot{u}}^k = Q_{\dot{\delta}}(\dot{\delta}^k)^2 + Q_{\dot{F}_x}(\dot{F}_x^k)^2 \tag{6}$$

$$J_e^k = Q_e(e^k - e_{\text{nom}}^k)^2 \tag{7}$$

All homotopies also share the same safety metrics. The optimization quadratically penalizes being within a certain distance to any obstacle or road boundaries. The bounds from moving obstacles are incorporated here.

$$J_{\text{env}}^k = p_{\text{obs}}(x^k, x_{\text{obs}}) + p_{\text{e}}(x^k, e_{\text{min}}^k) + p_{\text{e}}(x^k, e_{\text{max}}^k) \tag{8}$$

For faster solve times appropriate for real-time optimization, the vehicle is represented with two circles and the obstacles as a single circle. Further details on obstacle representation and distance calculation can be found in [3].

The longitudinal velocity objectives differ between passing and following maneuvers. We design a passing homotopy to have a speed profile tracking objective.

$$J_{ux}^k = Q_{ux}(u_x^k - u_{x,\text{des}}^k)^2 \tag{9}$$

In contrast, a following maneuver's velocity should depend on obstacle motion. We choose to track a distance corresponding to a 2-second time headway with a 5 m buffer. The shape of the cost functional is asymmetric: if the vehicle is ahead of the desired distance, it is closer to the obstacle and should be penalized more heavily than being far behind. Two softplus functions are used to achieve twice-differentiability (see Fig. 5).

$$J_{\Delta s}^k = Q_s\Big(3\log(1+\exp(-3(\Delta s^k + 2))) + \log(1+\exp(\Delta s^k - 5))\Big) \tag{10}$$

B. Nonlinear Program

The full nonlinear optimization for a passing maneuver is:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{N}\left(J_{\dot{u}}^k + J_e^k + J_{\text{env}}^k + J_{ux}^k\right) && \text{(11a)}\\
\text{subject to} \quad & x^{k+1} = f(x^k, u^k) && \text{(11b)}\\
& u_{\text{min}} \le u^k \le u_{\text{max}} && \text{(11c)}\\
& \dot{u}_{\text{min}} \le \dot{u}^k \le \dot{u}_{\text{max}} && \text{(11d)}\\
& x^k \in H_p(t^k, s^k) && \text{(11e)}
\end{aligned}$$

Upper and lower bounds are included for actuators and their rates of change. For a following maneuver, the program is identical with the exception of the objective function 12a and the constraint 12e:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{N}\left(J_{\dot{u}}^k + J_e^k + J_{\text{env}}^k + J_{\Delta s}^k\right) && \text{(12a)}\\
\text{subject to} \quad & x^{k+1} = f(x^k, u^k) && \text{(12b)}\\
& u_{\text{min}} \le u^k \le u_{\text{max}} && \text{(12c)}\\
& \dot{u}_{\text{min}} \le \dot{u}^k \le \dot{u}_{\text{max}} && \text{(12d)}\\
& x^k \in H_f(t^k, s^k) && \text{(12e)}
\end{aligned}$$

The optimization is solved with the interior point optimizer IPOPT [14] using automatic differentiation from CasADi [15], both of which are open-source packages.

IV. TRAJECTORY SELECTION

After all optimizations have returned from the parallel NMPC formulations, a clearly defined selection criterion is needed to compare the multiple trajectories from different homotopies. The objective function of the NMPC optimization and the trajectory evaluation are related but not identical. The NMPC objective focuses on finding a feasible trajectory within a homotopy, omitting additional factors that would unnecessarily increase the difficulty of the optimization. During the selection process, we can consider other high-level goals such as fuel efficiency, minimizing jerk, or driver preferences. For example, encoding that human drivers prefer to pass on the left on the highway could be represented by a constant small cost in the objective. While it could be incorporated in the NMPC optimization, it is more appropriate at the selector level. It is up to the user to determine which objectives should be present at each stage of the architecture.

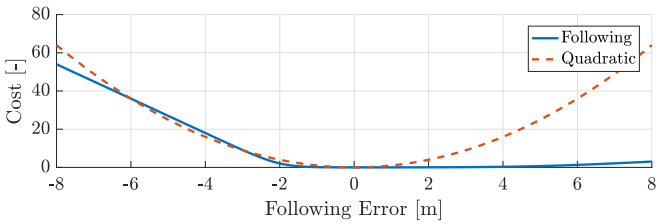

Fig. 5. Asymmetric and twice-differentiable following distance cost function using two softplus functions (negative values correspond to being closer to the obstacle than desired). Contrast with shape of quadratic penalty.

Fig. 6. The experimental platform X1.

The selection function for trajectory evaluation is:

$$J_{\text{select}} = \sum_{k=1}^{N} \left(J_{\dot{u}}^{k} + J_{\text{env}}^{k} + J_{\text{speed}}^{k} \right) + J_{\text{hyst}} \tag{13}$$

To represent safety, the distance to obstacles and road boundaries is included, as is the change in inputs for comfort. We omit the nominal trajectory-tracking terms that are present in the NMPC function: lateral position, speed, and following distance tracking. These are useful when solving an MPC problem but do not necessarily represent higher-level objectives. They are an especially weak proxy for safety, since the initial trajectory is an approximation, and the downstream controller may alter the final trajectory. For mobility, a reward is added for higher speeds u_x as

$$J_{\text{speed}}^{k} = -0.1 Q_{ux} (u_x^k)^2 \tag{14}$$

Finally, a small, constant reward is added for choosing to remain in the same homotopy as the previous solution, adding hysteresis when the trajectories are similar in quality.

V. EXPERIMENTAL RESULTS

The ego vehicle, drawn as a rectangle in Fig. 7, approaches a slower-moving obstacle in its lane, both moving in the increasing s direction. The obstacle is moving at 7 m/s, and the vehicle's desired speed is a constant 14 m/s. There is another obstacle in the oncoming lane, moving in the decreasing s direction at 10 mps. Two homotopies can be identified: following the in-lane obstacle and passing it on the left, with the static homotopy bounds equal to the road edges. The full architecture was run on an i7-6700 CPU and implemented within the Robot Operating System (ROS) framework [16]. The system replans every 50 ms or, in the case of a longer solve time, after the previous optimization is complete. Our formulation sees solve times of 28.4 ms on average and a maximum of 99.9 ms for each replanned trajectory, which allows the experiment to run in real-time on X1, a full-scale student-built vehicle shown in Fig. 6.

Fig. 7 shows the planned passing and following maneuvers when the oncoming obstacle is 88.2 m away from the ego vehicle in the longitudinal direction. At this moment, the NMPC optimization is unable to find a passing solution that avoids collision whereas the following plan allows the vehicle to remain a safe distance behind the obstacle in its lane. The selection function chooses the planned following maneuver, waiting for the oncoming vehicle to pass by first. A few time steps later, the predicted gap between the two

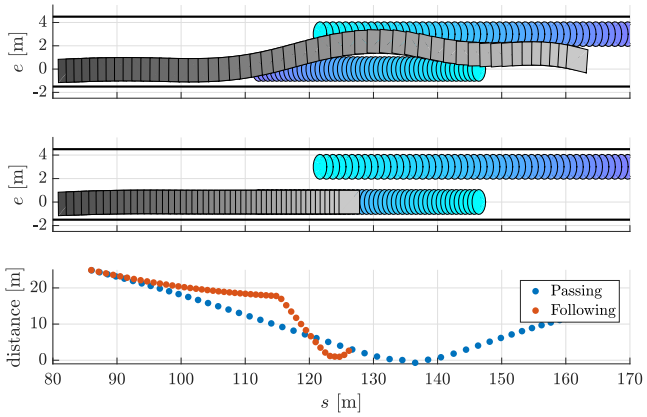

Fig. 7. Overhead plot of the planned passing (top) and following (middle) options visualized with a rectangular vehicle and circular obstacles. The bottom plot shows the minimum distance from the vehicle to either obstacle. The optimization is unable to find a collision-free passing solution.

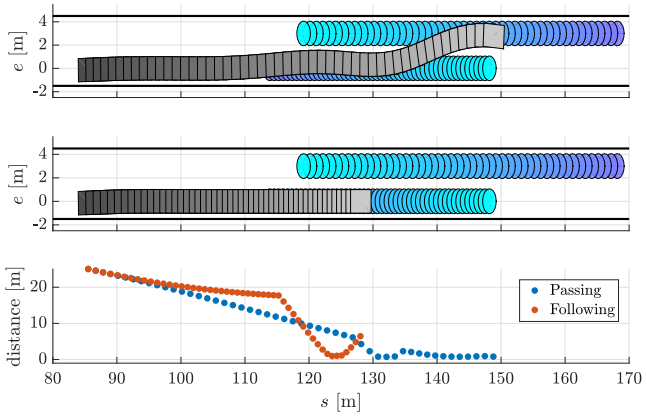

Fig. 8. Overhead plot of the planned passing (top) and following (middle) maneuvers 250 ms after the point in Fig. 7. The passing trajectory oscillates slightly, an artifact of nominal trajectory generation that has been resolved in simulation and will be tested in-vehicle at the next opportunity. The passing maneuver plans for the vehicle to remain at least 0.63 m from the obstacle.

obstacles widens, and the vehicle can minimally pass through the resulting space without collision (see Fig. 8). With the reward for mobility and smoothness, the passing homotopy outperforms following by our selection metrics. After this point, the passing homotopy continues to be the lower-cost option for the remainder of the experiment.

The overall closed-loop maneuver yields a smooth trajectory, with intuitive steering and longitudinal inputs shown in Fig. 9. At the transition point from following to passing, the vehicle continues to slow its longitudinal velocity by braking gently. This allows the oncoming obstacle to fully pass and also continues the trajectory from the following homotopy seamlessly. A few time steps later, the vehicle accelerates to return to the desired speed and complete the overtaking maneuver. This smooth transition between homotopies along with the decisiveness of the selection are desirable qualities in a decision-making and control architecture. Fig. 10 shows that the vehicle maintains a minimum distance of 0.5 m from either obstacle and 0.3 m from the road edges.

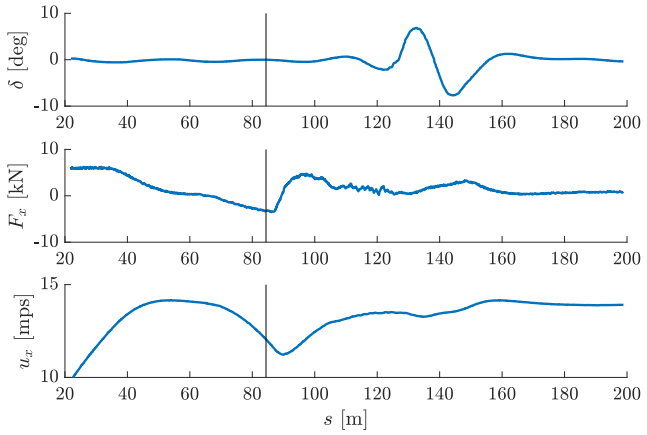

Fig. 9. Closed-loop inputs and longitudinal speed over the whole experiment. The gray line indicates the point at which the lowest-cost homotopy transitions from follow to pass.

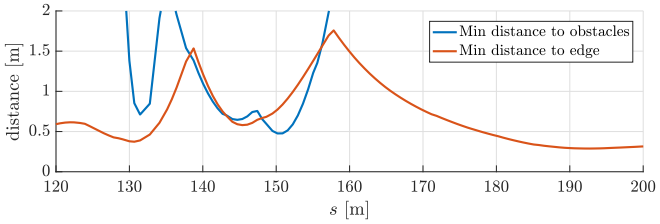

Fig. 10. Minimum distances from the vehicle to obstacles and road edges over the whole experiment.

VI. CONCLUSIONS

In this paper, we present an architecture designed to make high-level driving decisions by directly evaluating the output from parallel nonlinear model predictive controllers. The environment is efficiently separated into homotopies, and optimizations within each homotopy are solved with a high-fidelity model that capture the vehicle's full capabilities. In a scenario involving both in-lane and oncoming traffic, the architecture is able to evaluate the options of passing before or after the oncoming obstacle passes. Experimental results exhibit a smooth overtaking maneuver, transitioning from following to passing the obstacle in the ego lane while maintaining a safe distance from both obstacles. This framework is modular and flexible: every level of the system can be modified for individual applications.

Some hierarchies benefit from checking whether a lane change is possible or beneficial prior to planning the trajectory. For example, Ulbrich and Maurer use a signal processing network that takes as inputs factors such as the relative states of the vehicles and also mention other potential strategies using fuzzy logic or a Bayesian network [17]. In this architecture, we avoid performing an additional check and rely on the NMPC optimization to test for feasibility. Similarly, the selection process evaluates whether the maneuver is beneficial. We can be more confident in the selection outcome because the evaluated trajectories make full use of the vehicle's capabilities. Future experiments will test this framework in conjunction with more sophisticated obstacle motion models and tailor the selection function to consider environmental uncertainties.

VII. ACKNOWLEDGMENT

The authors thank their colleagues at Ford Motor Company for consistently thoughtful feedback, especially Eric Tseng, Michael Hafner, and Doug Rhode. The authors also thank the students and staff of the Dynamic Design Lab for their support in lab and at Thunderhill Raceway Park.

REFERENCES

[1] Y. Gao, A. Gray, J. V. Frasch, T. Lin, E. Tseng, J. K. Hedrick, and F. Borrelli, "Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles," in *International Symposium on Advanced Vehicle Control (AVEC)*, 2012.

[2] V. A. Laurense and J. C. Gerdes, "Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity," *IEEE Transactions on Control Systems Technology*, 2021.

[3] M. Brown and J. C. Gerdes, "Coordinating Tire Forces to Avoid Obstacles Using Nonlinear Model Predictive Control," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 1, pp. 21–31, 2020.

[4] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, Planning, Control, and Coordination for Autonomous Vehicles," *Machines*, vol. 5, no. 1, p. 6, 2017.

[5] J. Park, S. Karumanchi, and K. Iagnemma, "Homotopy-Based Divide-and-Conquer Strategy for Optimal Trajectory Planning via Mixed-Integer Programming," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1101–1115, 2015.

[6] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *IEEE Intelligent Vehicles Symposium, Proceedings*, (Seoul, Korea (South)), pp. 1386–1392, 2015.

[7] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha - A local, continuous method," in *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 450–457, IEEE, 2014.

[8] S. Sontges and M. Althoff, "Computing possible driving corridors for automated vehicles," in *IEEE Intelligent Vehicles Symposium, Proceedings*, (Redondo Beach, CA), pp. 160–166, 2017.

[9] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "MPDM: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1670–1677, IEEE, 2015.

[10] M. G. Plessen, D. Bernardini, H. Esen, and A. Bemporad, "Spatial-Based Predictive Control and Geometric Corridor Planning for Adaptive Cruise Control Coupled With Obstacle Avoidance," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 38–50, 2018.

[11] A. Liniger, A. Domahidi, M. Morari, and O. C. Applications, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, no. 36, pp. 628–647, 2015.

[12] S. M. Erlien, *Shared Vehicle Control Using Safe Driving Envelopes for Obstacle Avoidance and Stability*. PhD thesis, Stanford University, 2015.

[13] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.

[14] A. Wächter and L. T. Biegler, *On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*, vol. 106. 2006.

[15] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *International Conference on Robotics and Automation*, vol. 3, 2009.

[17] S. Ulbrich and M. Maurer, "Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving," in *IEEE Intelligent Transportation Systems*, pp. 2063–2070, 2013.