2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Optimal Assigner Decisions in a Hybrid Predictive Control of an Autonomous Vehicle in Public Traffic

Qian Wang[1], *Student Member, IEEE*, Beshah Ayalew[1], *Member IEEE* and Thomas Weiskircher[2]

***Abstract*— The complex public traffic environment requires an autonomous vehicle to have the ability of planning and executing a sequence of different maneuvers, such as maintaining cruising speed, changing speed to follow a vehicle in front or to lead a vehicle in the rear, and changing a lane when necessary and possible. This paper presents a hierarchical hybrid predictive control framework that integrates the discrete optimization problem of maneuver selection with particle motion-based model predictive trajectory guidance for an autonomous road vehicle. To address the challenge of solving the resulting mixed integer nonlinear programming (MINP) problem efficiently for online implementation, a relaxation method is introduced that transforms the MINP problem into a nonlinear programming problem with improved feasibility for online implementation. The performance of the proposed framework is illustrated via simulations of the autonomous vehicle in highway scenarios.**

I. INTRODUCTION

The basic task facing the controller of an autonomous vehicle in public traffic is the planning and guidance of the vehicle's motion while rapidly and systematically accommodating a number of possibly changing constraints. These include tire/road friction limits, vehicle actuation limits, avoiding stationary and moving obstacles, obeying traffic rules, considering passengers' comfort, and so on. This task must be addressed by designing computationally efficient motion planning algorithms that can determine the best sequence of the vehicle's maneuvers and constituent motion while satisfying the listed constraints.

Model predictive control (MPC)-based motion planning is receiving significant attention because it's able to generate feasible trajectories by solving control and state constrained optimization in a receding prediction horizon. In [1], a high-fidelity nonlinear vehicle model was implemented in a nonlinear MPC (or NMPC) for motion planning, but the computational burden was found to be too high in the presence of tightened constraints such as needed for obstacle avoidance. To reduce the computational complexity, a linear time-variant (LTV) model was used in [2] to approximate the vehicle dynamics for on-line MPC. This resulted in a multi-parametric programming problem. Previous works in [3] proposed a hierarchical NMPC framework which decomposed the motion planning work into a high-level MPC-based re-planner that uses a low-fidelity point-mass vehicle model and a low-level MPC-based follower that uses the high-fidelity vehicle model employed in [1, 2]. In addition to possible hierarchical model consistency issues, the computational burden of the two MPC optimization loops of this approach could be prohibitive for practical use. In [4], a reduced-order nonlinear particle motion model is used to design a high-level NMPC and either the predicted control output or the predicted state trajectories were configured to be tracked using traditional vehicle dynamics controllers at the lower-level [5]. This approach showed good performance in path tracking and obstacle avoidance, but the solver could get trapped at local minima in complex traffic environments if the setup for the optimal control problem in the NMPC is not re-configured by a higher-level decision making module. To this end, in [6], the authors introduced a hierarchical hybrid system control framework with a set of finite state machines (FSMs) embedded in the higher-level decision-making module (called assigner module) above the NMPC. Therein, the assigner module uses the information from a navigation module (route), surrounding traffic signs or obstacles and other constraints to select discrete maneuver states through pre-defined switching rules. In the present paper, we configure these decisions of the assigner module to be selected optimally and predictively so as to change the maneuver states and the setup of the NMPC as frequently as necessary for the prevailing driving scenario.

Similar hierarchical designs can be found in plenty of previous works. A hierarchical FSM concept with meta-state and sub-state machine for vehicle maneuver states was designed in [7] and has been implemented in an autonomous vehicle in the DARPA Urban Challenge [8]. In [9], a rule-based automaton (FSM) was designed to regulate the longitudinal motion maneuvers of autonomously controlled vehicles (ACVs) to avoid collision under cruising and merging scenarios. A game theory method was used in [10] to design a robust hybrid controller, which guarantees safety in vehicle platooning under some uncertainties. The method was applied later by [11] with non-deterministic automaton to regulate an intersection problem. In most of these existing works, the discrete decisions of maneuver selection were not done optimally in a predictive horizon fashion.

The optimization of such hybrid systems involves solving for the optimal discrete states (maneuvers) as well as their optimal local-control inputs (such as accelerations) [12]. This leads to a mixed-integer nonlinear programming problem (MINP) [13], which requires specific heuristic-based solvers to find a solution. However, the NP-complete nature of the MINP problem makes it hard to solve efficiently for real-time implementation, especially when the problem at hand is complex with nonlinearity and fast dynamics [14]. This is the case for ACVs in many public traffic scenarios. Another contribution of the present paper, therefore, is the introduction of relaxation constraints to transform the MINP into a regular nonlinear programming,

1 Qian Wang and Beshah Ayalew are with the Applied Dynamics& Control Group at the Clemson University – International Center for Automotive Research (CU-ICAR), 4 Research Dr., 29607, Greenville, SC,USA, {qwang8, beshah}@clemson.edu
2 Thomas Weiskircher is currently at Daimler R&D, Germany.

which can be solved efficiently in millisecond time-scale with multi-shooting methods and auto-generated code suitable for real-time implementation [15].

The rest of the paper is organized as follows. Section II introduces the proposed hierarchical hybrid predictive control framework. Section III introduces reformulation of the MPC problem for maneuver planning, discusses how it results in a MINP problem and outlines the relaxation method used to approximate and solve it. Results and discussions are included in Section V to illustrate the workings of the proposed framework and brief conclusions are offered in Section VI.

II. Hybrid Predictive Control Framework

A. Hierarchical Framework

Fig. 1 shows a schematic of the proposed hierarchical hybrid control framework for the autonomous road vehicle. It consists of the route navigator module, the environment recognition module, the higher-level maneuver planning module and the lower-level maneuver execution module.

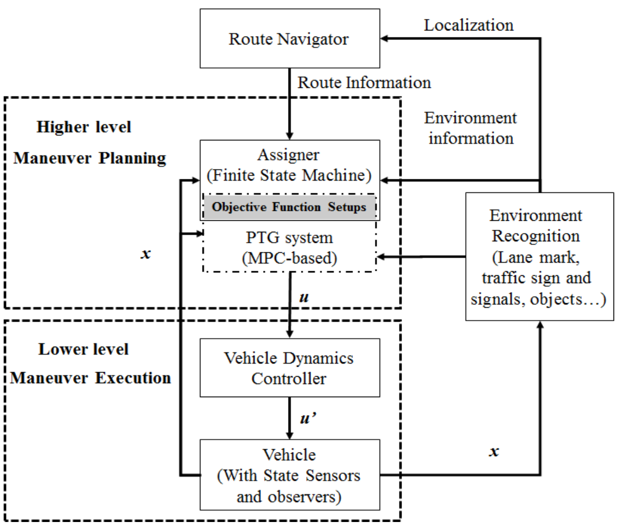

Figure. 1. Hierachical control framework

The route navigator module (such as a GPS navigator) plans the route from initial position to the final position based on a map and localization of the controlled vehicle. The environment information, such as lane marks, traffic signs or signals, the size or states of moving objects, are assumed captured in the environment recognition module via camera, radar, lidar or wireless devices (V2V, V2I). The details of these two modules are beyond the scope of this paper and their output information is simply assumed available to the high-level maneuver planner.

The assigner module is responsible for the selection of the discrete maneuver states. Different maneuvers are designed and grouped as (sets of) finite state machine (s) for the assigner to select from. At each time step, the assigner can choose a maneuver for each lane using the information from the navigation and environmental modules. The decision on the maneuvers for lane selection and the generation of the necessary trajectories and/or control outputs (e.g. accelerations) for the selected maneuver is then computed by the MPC-based predictive trajectory guidance (PTG) system. The MPC in the PTG system uses a 2D curvilinear particle motion description of the vehicle, the associated path references, obstacle descriptions, tire-road friction constraints and/or traffic rules (see, e.g.[4]). From the assigner, we associate specific objective function setups (maneuver-tracking references and corresponding weights) with each maneuver and pass them to a single versatile MPC configuration of the PTG. In our previous work [6], a rule-based maneuver selection strategy has been used inside the assigner, which selected a single maneuver and merely passed the corresponding objective function setup to the PTG. In this paper, we apply rules to pre-select a set of maneuver candidates for *each lane*, and then we introduce optimality for further selection of the best maneuver from this set. To this end, we reformulate the MPC problem in the PTG with a new objective function set up. This endows the PTG with the ability to intelligently switch the maneuver plan within the prediction horizon. Effectively, this arrangement moves part of the maneuver selection tasks of the assigner to the PTG (Fig.1). It also converts the optimization problem for the MPC into a MINP problem. We shall detail this in Section III.

The control outputs computed by the PTG will be sent to the lower-level controllers of the continuous vehicle dynamics for execution via the available lower-level vehicle dynamics controllers, whose discussion is omitted here. The reader is referred to [5] and other standard references for this topic.

B. Hybrid System Modeling of the Vehicle's Maneuvers

Here, we briefly discuss the modeling of the vehicle's motion maneuvers as a hybrid system. A hybrid system involves both continuous dynamics and discrete state dynamics whose evolutions depend on each other. It can often be viewed as a collection $\boldsymbol{\Sigma}$ of indexed continuous dynamical systems. There often exist some switching or jumping maps among them, such as when the state enters a specified subset of the state space, named a jump set. The jump sets can be defined either by the inherent evolution of the system (autonomous jump sets) or by external control commands (controlled jump sets) [12].

In this vein, the autonomous road vehicle switching or jumping between different maneuvers can be characterized as a hybrid system with the automaton depicted in Fig. 2. In general, the assigner module may consist of several finite state machines (FSMs) at the top. Based on the scenario the ACV is in, a relevant FSM will be chosen. In each FSM, for example for the highway case shown, the ACV can jump or switch its maneuver among the constituent dynamical systems denoted by the elliptical nodes (see maneuver definitions in paragraph below). Each arrowed-edge represents a jump between the different maneuvers with the labels describing the required jumping map/condition. The black dashed edges represent jumps defined by pre-defined embedded switching rules like those listed in our previous work [6], while the solid red edges indicate optimization-based jumps we detail in the present paper. Both of these sets of jumps can be viewed as autonomous jumps.

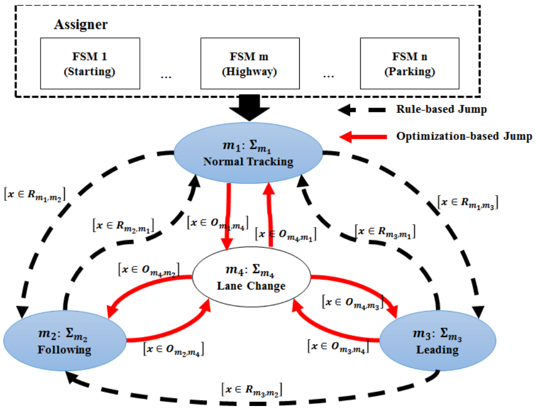

Figure. 2. Maneuver automaton in the assigner. Rule-based jump sets are denoted by ***R*** and Optimization-based ones by ***O***.

For the highway FSM example, for *each lane*, we define three reference maneuvers (see the blue elliptical nodes in Fig.2) for the ACV: normal tracking, following or leading. The jump sets between these maneuvers are defined by target speed assignment rules shown in Fig 3. These jumps are denoted by ***R*** in Fig. 2. A normal tracking maneuver is defined as tracking a normal cruise speed v_{ref} . A following maneuver for the ACV refers to tracking the speed $v_{t,of}$ of a detected front object vehicle on the same lane with $v_{t,of} < v_{ref}$, and a leading maneuver refers to tracking the speed $v_{t,or}$ of a detected rear vehicle on the same lane with $v_{t,or} > v_{ref}$. These are depicted in Fig. 3, where v_t is the longitudinal speed of the ACV, $v_{t,r}$ is the reference/target speed of the ACV assigned to the specific lane. A lane change maneuver is defined by switching the reference lane from the current one to another. The jump transition to this maneuver will be determined by the optimization in the (re-formulated MPC of the) PTG module based on the target speed for each lane and the current state of the ACV.

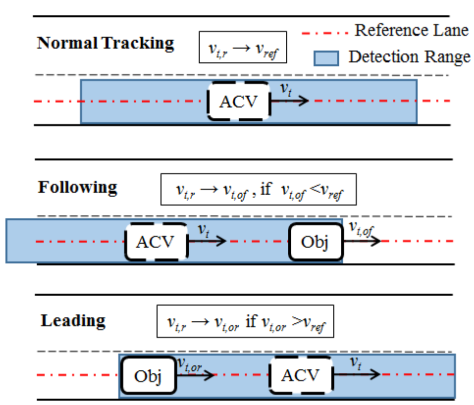

Figure 3. Target speed assignment.

The switching of target speed $v_{t,r}$ for each lane can also be considered as a controlled jump when it is imposed by traffic signs and other exogenous traffic control devices. This case is not pursued in this paper.

III. MPC Problem Re-Formulation

A. Particle Motion Based Vehicle Model

The 2D curvilinear particle motion based MPC proposed in [4] is re-formulated here as maneuver planner integrating some of the assigner functions. The vehicle states and reference path defined in the Frenet frame are shown in Fig. 4. The motion of the particle/vehicle with respect to the local reference path (lane centerline) is given by the angular alignment error and lateral error. The following equations summarize the resulting nonlinear dynamics model describing the motion as well as the evolution of the path coordinate s:

$$\dot{v}_t = a_t \tag{1}$$

$$\dot{\psi}_e = \dot{\psi}_p - v_t \cos(\psi_e)\left(\frac{\kappa(s)}{1-y_e\kappa(s)}\right) \tag{2}$$

$$\dot{y}_e = v_t \sin(\psi_e) \tag{3}$$

$$\dot{a}_t = (a_{t,d} - a_t)/T_{a_t} \tag{4}$$

$$\ddot{\psi}_p = (v_t\kappa(s) + \Delta\dot{\psi}_{p,d} - \psi_p)/T_{\psi_p} \tag{5}$$

$$\dot{s} = v_t \cos(\psi_e)\left(\frac{1}{1-y_e\kappa(s)}\right) \tag{6}$$

In these motion equations, the desired acceleration $a_{t,d}$ and the desired deviation from the reference yaw rate $\Delta\dot{\psi}_{p,d}$ are the treated as the inputs used to control the particle along the path. The reference path curvature κ is assumed to be known along the reference path coordinate s. v_t, a_t are the particle speed and acceleration along the path. $\dot{\psi}_p$ is the yaw rate, ψ_e, y_e are the alignment error and lateral error to the reference path, and T_{a_t}, T_{ψ_p} are the time constants of the first-order approximation of the longitudinal and lateral vehicle dynamics. $\dot{s}$ is the projection of the speed v_t on the reference path.

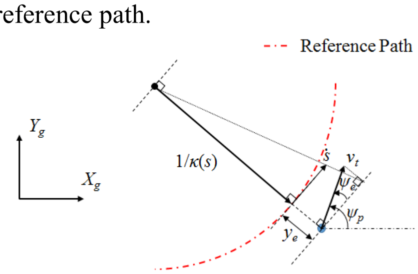

Figure 4. Particle motion description for the vehicle

The nearby objects (obstacles or moving vehicles) are also considered on the road, as shown in Fig. 5. The motion of object i is given by:

$$s_{o_i} = s_{o_i,0} + v^s_{t,o_i} t + \frac{1}{2} a^s_{t,o_i} t^2 \tag{7}$$

$$y_{e,o_i} = y_{e,o_i,0} + v^s_{n,o_i} t + \frac{1}{2} a^s_{n,o_i} t^2 \tag{8}$$

where, t is the internal time in the prediction model (of the MPC). Equation (7) and (8) are then used to estimate the position of the objects in the prediction horizon based on the current measurement of the longitudinal velocity v^s_{t,o_i}, longitudinal acceleration a^s_{t,o_i}, lateral velocity v^s_{n,o_i} and lateral acceleration a^s_{n,o_i} in path coordinate. The a^s_{t,o_i} and a^s_{n,o_i} are held constant for the prediction horizon, but are to be updated at each MPC update. The initial positions of object i (at prediction) are denoted ($s_{o_i,0}, y_{e,o_i,0}$).

B. Constraints

The constraint to keep a safe distance between the ACV and any nearby object i is modeled by the elliptic inequality:

$$\left(\frac{y_e - y_{e,o_i}}{\Delta y_{e,o_i}}\right)^2 + \left(\frac{s - s_{o_i}}{\Delta s_{o,ss} + f_{\zeta,Do}\zeta_{Do}}\right)^2 \geq 1 \tag{9}$$

which is also depicted in Fig. 5. $\zeta_{Do} \geq 0$ is a slack variable, which allows the solver to find a feasible solution in emergency situations. $f_{\zeta,Do}$ is an optional tuning parameter (has a unit of time). $\Delta y_{e,oi}$ and $\Delta s_{o,ss}$ are calculated by incorporating the geometry (length and width) of the objects and the ACV. These are assumed available from sensing and/or V2V communication.

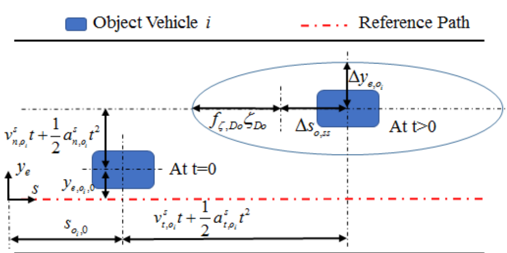

Figure 5. Object motion definition in road reference frame

The control input must also be limited to physical constraint of the acceleration according to the friction ellipse of a real vehicle's tire/road contact:

$$\left(v_t\left(\kappa(s)v_t + \Delta\dot{\psi}_{p,r}\right)/a_{n,gg}\right)^2 + \left(a_{t,d}\right)^2 \leq \left(\mu_H g - \zeta_{gg}\right)^2 \tag{10}$$

Here, μ_H is the limiting tire-road friction coefficient, g is the gravitational constant. $a_{n,gg} \in [0,1]$ is the scaling of the ellipse for lateral acceleration. The slack variable ζ_{gg} enables the formulation of the limit value of the combined accelerations as a soft constraint.

Other state constraints like the lane boundaries, speed limits and the minimum turning radius, etc are also considered but not described here due to lack of space. For complete details, please refer to [4].

C. MPC Problem Re-Formulation

Unlike previous work [4], where a fixed suite of references are considered in the MPC objective for the predictive trajectory guidance(PTG) module, here we configure the MPC to have the ability to intelligently switch between maneuver states and their corresponding references. To this end, we introduce a lane index q and an associate weight variable Z_q. The optimization problem to be solved over the prediction horizon [0, H_p] is then re-formulated as:

$$\min_{x_k,u_k,Z_{q,k}} \sum_{k=1}^{N_p}\sum_{q\in Q}\left\|Z_{q,k}\left(y_{1,k} - r_{1,q,k}\right)\right\|_{P_1}^2 + \sum_{k=1}^{N_p}\left\|y_{2,k} - r_{2,k}\right\|_{P_2}^2 + \sum_{k=0}^{N_p-1}\left\|u_k\right\|_R^2 \tag{11}$$

subject to :

$$\dot{x} = f\left(x,u,Z_q\right),\ u \in U\ ,\ x \in X \tag{12}$$

$$x(0) = x_0 \tag{13}$$

$$0 \leq c\left(x,u\right) \tag{14}$$

$$\sum_{q\in Q} Z_q = 1, \quad Z_q \in \{0,1\} \tag{15}$$

Here, $k \in (0, 1, \ldots, N_p)$, where N_p is the prediction step length, is the equidistantly sampled time index of the discrete time version of the continuous system (12) representing the vehicle model given by (1)-(6) and auxiliary equations(integrators) for the slack variables. x covers all the state variables of the particle motion and slack variables; and X represents the state space for x. x_0 denotes the current/initial state. The prediction horizon H_p is defined by $H_p = N_p \Delta T$ and ΔT is the minimum sample time of the MPC. q is lane index and Q is the set of lanes. $r_{1,q}$, contains the speed and lateral position references for maneuver candidates for each lane and r_2 contains references for the slack variables. P_1, P_2 and R are the weighting matrices for the candidate maneuver tracking error, slack variable reference tracking error and control efforts, respectively. y=[y_1, y_2] is the system output, including the speed v_t and lateral position y_e of the ACV which are grouped in y_1, and the slack variable outputs ζ_{Do}, ζ_{gg} , grouped in y_2. See (19-22) in the next section for an example of how these variables are defined.

The control variable u is applied in a piecewise constant fashion, as u_k , and only the first step u_0 will be used to control the system for each MPC update step. In (12), U denotes the admissible set for u. All the nonlinear constraints such as the collision avoidance and friction limits are included in (14).

In the optimization problem (11), Z_q is the weight variable for selection of the q-th lane and the reference speed on that lane denoted by $r_{1,q}$. At each time k, only one maneuver is chosen, so Z_q should satisfy the integrality constraints (15). This formulation leads to a MINP problem to be solved at each MPC update step.

D. Relaxation Method

The MINP problem is hard to solve efficiently at real time since it is NP-complete. But some relaxations can be made to approximate the problem as a nonlinear programming problem that is tractable for real-time implementations. This is achieved by extending the feasible solution set of the problem by relaxing the integrality constraints in (15) as:

$$\sum_{q\in Q} Z_q = 1, \quad Z_q \in [0,1] \tag{18}$$

where, the weight variable Z_q becomes a real variable which can then be regarded as an additional state in (12) with the auxiliary dynamics:

$$\dot{Z}_q = u_{zq} \tag{19}$$

where u_{zq} is a virtual input used to manipulate Z_q. This becomes one part of the control vector u in (14). Then, the new optimization problem at each MPC step is given by:

$$\min_{x_k, u_k} \sum_{k=1}^{N_p} \sum_{q \in Q} \left\| Z_{q,k} \left(y_{1,k} - r_{1,q,k} \right) \right\|_{P_1}^2 + \sum_{k=1}^{N_p} \left\| y_{2,k} - r_{2,k} \right\|_{P_2}^2 + \sum_{k=0}^{N_p - 1} \left\| u_k \right\|_R^2 \quad (20)$$

subject to the modified constraints, including (18) and (19).

IV. Results and Discussions

A. Simulation Setting

To illustrate the performance of the proposed hybrid predictive control approach, we consider two highway scenarios on a section with three straight lanes, see Fig. 6.

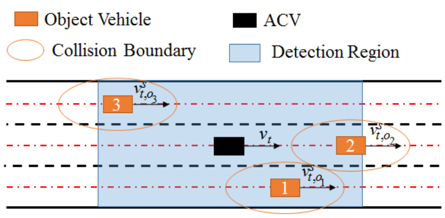

Figure 6. Highway scenario with 3 lanes

The jump event of a lane change maneuver is activated based on the value of weight variable Z_q optimized according to (18). For the lanes in Fig. 6 and the finite set of maneuvers in Fig. 2, all the discrete maneuvers for the hybrid system are listed in Table I.

TABLE I. Discrete Maneuver States

State index	Maneuver Description	Weight Coefficient
1	Normal Tracking in Lane 1	
2	Following in Lane 1	Z_1
3	Leading in Lane 1	
4	Normal Tracking in Lane 2	
5	Following in Lane 2	Z_2
6	Leading in Lane 2	
7	Normal Tracking in Lane 3	
8	Following in Lane	Z_3
9	Leading in Lane 3	

For the three-lane highway scenario, Q={1, 2, 3}, $q \in Q$. The reference state of each candidate maneuver will be assigned in optimization setups of the reformulated MPC. To this end, the rest of the terms in (20) are given by:

$$y_{1,k} = \left[y_{e,k}, v_{t,k} \right]^T \quad (19)$$

$$y_{2,k} = \left[Z_{1,k}, Z_{2,k}, Z_{3,k}, \zeta_{gg,k}, \zeta_{Do,k} \right]^T \quad (20)$$

$$r_{1,q,k} = \left[y_{e,r,q,k}, v_{t,r,q,k} \right]^T \quad (21)$$

$$r_{2,k} = \left[1,1,1, \zeta_{gg,r,k}, v_{t,k} \right]^T \quad (22)$$

$$u_k = \left[a_{t,d,k}, \Delta\dot{\psi}_{p,d,k}, u_{z1}, u_{z2}, u_{z3} \right] \quad (23)$$

Due to the problem formulation in MPC, the optimization can be "trapped" in one lane with following maneuver or a leading maneuver instead of normal tracking. This might lead to significant deviation from the desired speed v_{ref}. As these maneuvers represent parts of the global minimums of the problem, changes must be made to the objective function to force the state flow jump from the current minimum and converge to other desired global minimums. Therefore, a forced maneuver from the maneuver candidates is added to the MPC by making its weight variable Z_q track a value of 1 in (20) and (22), with related weight p_{zq} in P_2 for control activation. The jump set to this special maneuver is defined as the violation of a tolerance speed range [v_{lcl}, v_{lch}] around v_{ref} with $v_{lcl} \leq v_{ref} \leq v_{lch}$ and the availability of an open lane for normal tracking:

$$p_{zq} = \begin{cases} w_z, \ if \ v_t \notin [v_{lcl}, v_{lch}] \ \wedge \ v_{t,r,q} = v_{ref} \\ 0, \ else \end{cases}, q = 1,2,3 \quad (26)$$

where w_z is a positive value large enough to activate the forced lane change.

To eliminate the redundancy of selecting multiple maneuvers as the forced maneuver, a priority of the lane selection should be followed for lanes with p_{zq}=w_z in descending order from lane 2, lane 3 to lane 1. Thus, a p_{zq} with lower priority will be reset to zero. Such a priority design keeps the ACV preferring lane 2 (middle lane) for cruise speed tracking. The priority can also be set to accommodate other local traffic rules/customs.

The control approach and simulation settings above are applied to controlling the high fidelity vehicle model used by [8] in two highway scenarios with v_{ref} = 30m/s. The nonlinear programming problem in the re-formulated MPC is solved efficiently via the ACADO Toolkit [16] which implements sequential quadratic programming. For details, also see [17].

B. Results

Fig. 7 shows the results for scenario 1: the ACV with a faster initial speed at 30m/s changes lane from lane 3 to lane 2 (Z_3 reduces and Z_2 rises up significantly) and follows the slower object vehicle (OV), OV2, traveling with a speed of 25 m/s to avoid the OV3 in the front on lane 3 travelling at the even slower speed of 20 m/s. In this procedure, ACV doesn't change to lane 1 because OV1 occupies lane 1 in parallel with OV2 at the same forward speed, which makes following OV1 more costly. The observation of the ACV slowing down to the speed, which violates the tolerance speed range [27.5, 32.5] m/s around v_{ref} is due to the unavailability of all lanes. When it passes OV3 and lane 3 becomes available for normal tracking, a forced maneuver of changing the lane back to lane 3 and accelerating to track v_{ref} is activated. Finally, the ACV passes OV2 via lane 3.

Fig. 8 shows the highway scenario 2 to avoid a faster rear OV3 that changes lane suddenly. Initially, ACV goes on lane 3 at speed of 30m/s in parallel with OV2 on lane 2. When the ACV detects OV3 approaching at 35m/s from behind. The reference speed for lane 1 is switched to 35m/s and the controller decides to change lane to lane 2 to minimize the global cost. However, as lane 2 has been occupied by OV2 nearby, the ACV has to accelerate and bypass the elliptical collision avoidance boundary (9) to reach lane 2. During the bypass, the ACV goes a long way to change lane from lane 3 to lane 2, thus Z_2 changes slowly from 0 to 1. At around t=26s, when the ACV successfully reaches lane 2 and OV3 catch up with the ACV in longitudinal direction, OV3 suddenly changes its lane from lane 3 to lane 1, which forces the ACV to brake sharply to avoid collision.

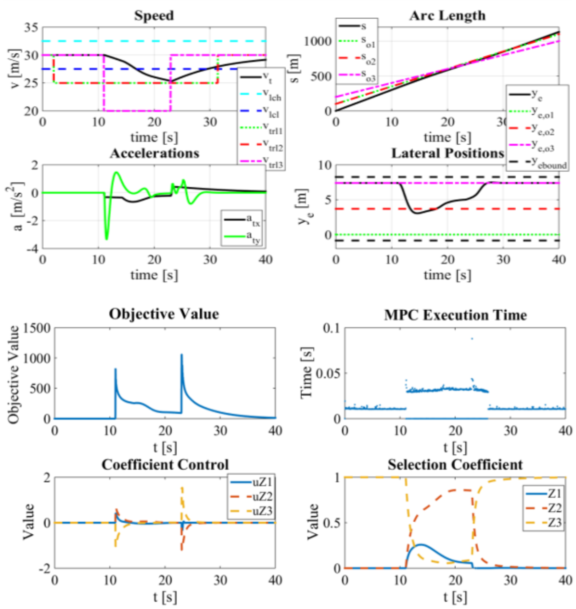

Figure 7. Highway scenario 1

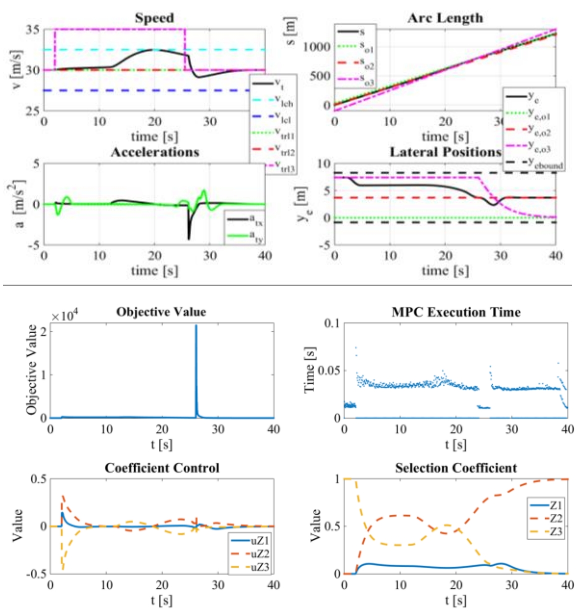

Figure 8. Highway scenario 2

Note that in both scenarios, the MPC execution times are of the order of 40ms or less; increasing mainly when the obstacle inequality constraints are engaged.

V. Conclusion and Future Work

This paper presented a hierarchical hybrid predictive control framework that integrates the discrete optimization problem of maneuver selection with particle motion-based model predictive trajectory guidance for an autonomous road vehicle. A convex relaxation method is used to transform the resulting MINP problem into a nonlinear programming problem with improved feasibility for online implementation. Two highway scenario simulations of the ACV controlled with this framework have been included to show how it successfully guides the vehicle with reasonable MPC execution times.

Future work will focus on: 1) Extending the optimality to the pre-selection of the maneuver candidates; 2) More strategies for eliminating local minimums in the domain of the objective function; 3) Stability, robustness and convergence analysis of the framework.

References

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

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

[3]. P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat. "A hierarchical model predictive control framework for autonomous ground vehicles." In *American Control Conference, 2008*, pp. 3719-3724. IEEE, 2008.

[4]. T. Weiskircher and B. Ayalew. "Predictive Control for Autonomous Driving in Dynamic Public Traffic," Proceedings of the American Control Conference, July 1-3, Chicago, IL

[5]. T. Weiskircher and B. Ayalew. "Frameworks for Interfacing Trajectory Tracking with Predictive Trajectory Guidance for Autonomous Road Vehicles." Proceedings of the American Control Conference, July 1-3, Chicago, IL

[6]. Q. Wang, T. Weiskircher, B. Ayalew, *Hierarchical Hybrid Predictive control of an Autonomous Road Vehicle*, Proceedings of the ASME 2015 Dynamic System and Control Conference, Oct. 28-30, Columbus, OH.

[7]. K. Arda, and Ü. Özgüner. Hierarchical finite state machines for autonomous mobile systems. Control Engineering Practice vol. 21, no.2, 2013, pp. 184-194.

[8]. C. Urmson, et al. "Tartan racing: A multi-modal approach to the DARPA Urban Challenge." ,2007.

[9]. A. Girault. A hybrid controller for autonomous vehicles driving on automated highways. *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 6, pp.421-452, Dec. 2004

[10]. L., John, D. N. Godbole, and S. Sastry. "Verified hybrid controllers for automated vehicles." *Automatic Control, IEEE Transactions on* vol. 43, no. 4, 1998, pp. 522-539.

[11]. V. Rajeev, and D. Del Vecchio. "Safety control of hidden mode hybrid systems." *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, 2012, pp. 62-77.

[12]. M. S. Branicky, V. S. Borkar, and S. K. Mitter. "A unified framework for hybrid control: Model and optimal control theory." *Automatic Control, IEEE Transactions on,* vol. 43, no. 1, 1998, pp. 31-45.

[13]. P. Belotti, et al. Mixed-integer nonlinear optimization. Acta Numerica, no.22, 2013, pp. 1-131

[14]. A. Richards, and J. How. "Mixed-integer programming for control."*American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005.

[15]. M. Vukov, et al. "Auto-generated algorithms for nonlinear model predictive control on long and on short horizons." *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013.

[16]. B. Houska, H. J. Ferreau, and M. Diehl. "ACADO toolkit—An open-source framework for automatic control and dynamic optimization."*Optimal Control Applications and Methods*, vol. 32, no. 3, 2011, pp. 298-312.

[17]. H. J. Ferreau, et al. "qpOASES: A parametric active-set algorithm for quadratic programming". *Mathematical Programming Computation*, vol. 6, no. 4, 2014 , pp. 327-363.