A Hierarchical MPC Approach to Car-Following via Linearly Constrained Quadratic Programming

Fangyu Wu
Dept. of EECS
University of California, Berkeley
Berkeley, USA
fangyuwu@berkeley.edu

Alexandre M. Bayen
Dept. of EECS
University of California, Berkeley
Berkeley, USA
bayen@berkeley.edu

Abstract—Single-lane car-following is a fundamental task in autonomous driving. A desirable car-following controller should keep a reasonable range of distances to the preceding vehicle and do so as smoothly as possible. To achieve this, numerous control methods have been proposed: some only rely on local sensing; others also make use of non-local downstream observations. While local methods are capable of attenuating high-frequency velocity oscillation and are economical to compute, non-local methods can dampen a wider spectrum of oscillatory traffic but incur a larger cost in computing. In this article, we design a novel non-local tri-layer MPC controller that is capable of smoothing a wide range of oscillatory traffic and is amenable to real-time applications. At the core of the controller design are *1)* an accessible prediction method based on ETA estimation and *2)* a robust, light-weight optimization procedure, designed specifically for handling various headway constraints. Numerical simulations suggest that the proposed controller can simultaneously maintain a variable headway while driving with modest acceleration and is robust to imperfect traffic predictions.

Index Terms—Autonomous vehicles, predictive control for linear systems, hierarchical control.

I. Introduction

Despite significant progress in recent years, autonomous driving still remains a challenging problem. Among many problems in this area, single-lane car-following is arguably a fundamental quest.

It is widely known that human car-following is inherently sub-optimal. Suigiyama and Tadaki have demonstrated in [1], [2] that stop-and-go waves could emerge in a circular platoon of human vehicles without the presence of physical bottlenecks or lane changes.

To improve, various *local* controllers have been proposed [3]–[7]. For example, Luo has described a model predictive control (MPC) formulation for fuel-efficient, fixed-headway adaptive cruise control [5]. Kerner et. al. have presented a variable-headway controller that can attenuate propagation of stop-and-go waves [6]. In addition, Stern et. al. have demonstrated that a single vehicle controlled by a proportional-integral controller is able to dampen stop-and-go waves induced by human drivers in a circular platoon [7].

Beyond local methods, various approaches on *non-local* controllers have been proposed in an effort to attenuate a wider spectrum of oscillations, such as [8]–[10]. The key advantage of non-local controllers lies at the fact that with non-local knowledge of downstream traffic it can react in advance to dampen non-local oscillations. For instance, [9], [10] introduced various variable-headway MPC methods to achieve smoother car-following by using information from downstream traffic lights.

A common feature of the non-local methods above is a bi-layer structure consisting of a prediction layer and a control layer. To predict, these methods require dedicated sensing infrastructure and either vehicle-to-infrastructure [9], [10] or vehicle-to-vehicle [8] communication. While non-local methods may have higher performance upper bound, they are also more prone to errors [11]. Yet, mitigating the impacts of erroneous predictions has been largely overlooked in previous works.

To address prediction errors, a common idea is to subdivide the control layer into a planning layer and a tracking layer. First, the planning layer produces a reference trajectory using the forecast from the prediction layer. Then, the tracking layer attempts to track this reference without violating certain performance constraints. This approach has found many applications in transportation outside of car-following. For example, [12], [13] have applied this idea for autonomous intersection coordination.

Adopting this idea of planning-then-tracking, we propose a novel hierarchical MPC controller, consisting of a prediction layer, a planning layer, and a tracking layer. The core MPC formulation assumes the form of a linearly constrained quadratic program (LCQP). Due to the tri-layer design, the controller is robust to prediction errors. In contrast to [9], [14], the method does not require dedicated sensing and communication infrastructure for prediction. Rather, it only uses an estimated time of arrival (ETA) estimator, which is commonly accessible from mainstream map service providers and needs only cellular network. In addition, compared to [9], [10], our method admits a more flexible formulation of headway constraints and is better understood when it comes to theoretical properties such as optimality and feasibility.

We regard the primary contribution of the paper as a novel

tri-layer MPC design, along with a comprehensive numerical study that demonstrates its robustness to prediction errors. At the code of the MPC design are: *1)* A simple, accessible prediction method using an ETA estimator and *2)* A robust, light-weight LCQP formulation.

We state the problem of the two-vehicle car-following in Section II. Next, we formally define and analyze the tri-layer hierarchical MPC control scheme in Section III. We evaluate the performance of our MPC controller via numerical simulations in Section IV and discuss the numerical results in Section V. Lastly, we conclude our findings in Section VI.

II. PROBLEM STATEMENT

Consider a two-vehicle setup, where a considered vehicle follows a preceding vehicle that travels along a path of length L. Denote the initial and final times of the trip as t_0 and t_f, respectively.

Let the state of a vehicle be $\mathbf{x} := \begin{bmatrix} s & v \end{bmatrix}^\top$, where s and v represent the position and speed of a vehicle, respectively. Define a as the acceleration of a vehicle. Let ℓ be the length of a vehicle. We use superscript to identify to which vehicle a variable belongs, where the considered vehicle is indexed as one and the preceding vehicle zero. Thus, $\mathbf{x}^1$ refers to the state of the considered vehicle. Let the initial position of the considered vehicle be w_0, that is, $s^1(t_0) := w_0$. Let the initial position of the preceding vehicle be w_1, that is, $s^0(t_0) := w_1$. Let $w_f := w_1 + L$. Then, $s^0(t_f) := w_f$. The two-vehicle scenario is illustrated in Figure 1.
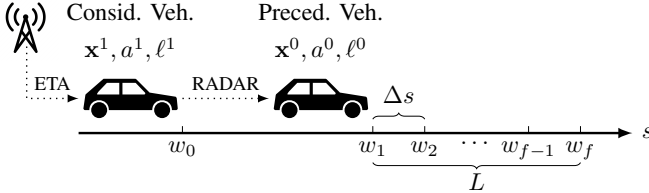

Fig. 1: Scenario of the two-vehicle car-following problem. At t_0, the considered vehicle is at w_0 and the preceding vehicle at w_1. The preceding vehicle travels through a set of way points $(w_1, w_2, \ldots, w_f)$, where each pair of consecutive way points are spaced Δs apart, barring boundary condition.

For the considered vehicle to properly follow the preceding vehicle, we impose the following five requirements. First and foremost, to prevent collision, we constrain the space headway between the two vehicles to be greater than some minimal headway $h_{\min}(t) : \mathbb{R} \to \mathbb{R}_{\geq 0}$. Next, to *follow* the preceding vehicle, we constrain the space headway to be smaller than some maximum headway $h_{\max}(t) : \mathbb{R} \to \mathbb{R}_{\geq 0}$. The third constraint requires the speed of the considered vehicle to fall within some speed limits $[v_{\min}, v_{\max}]$. Next, we constrain the acceleration of the considered vehicle to be bounded within some range $[a_{\min}, a_{\max}]$. Last but not least, among all trajectories that satisfies the constraints above, we select one that is the smoothest, as measured by ℓ^2-norm of the acceleration.

We summarize the above specifications in Problem 1.

Problem 1. *Provided $s^0(t)$ for $t \in [t_0, t_f]$ with some finite t_f, determine a trajectory of least acceleration in ℓ^2-norm for the considered vehicle $s^1(t)$ for $t \in [t_0, t_f]$ such that 1) $\min_{\forall t} s^0(t) - s^1(t) \geq h_{\min}(t)$, 2) $\max_{\forall t} s^0(t) - s^1(t) \leq h_{\max}(t)$, 3) $v^1(t) \in [v_{\min}, v_{\max}]$ for all t, and 4) $a^1(t) \in [a_{\min}, a_{\max}]$ for all t.*

With the problem stated above, we present the proposed controller in the next section.

III. OPTIMAL CAR-FOLLOWING CONTROLLER

In this section, we present a hierarchical MPC scheme consisting of 1) a prediction layer, 2) a planning layer, and 3) a tracking layer. The layered controller design is illustrated in Figure 2. We present the details of the three layers below.
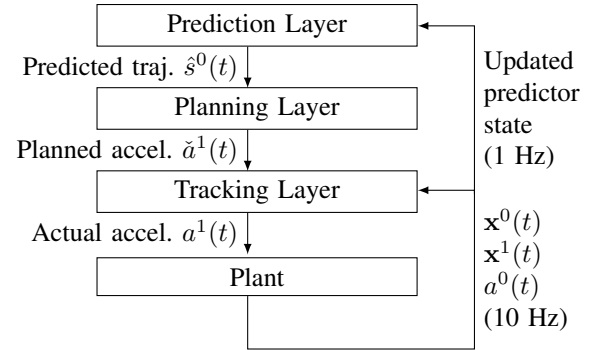

Fig. 2: Diagram of the MPC controller design. Prediction layer and planning layer are invoked every 1 second. Tracking layer is triggered every 0.1 second.

A. Prediction layer

The prediction layer predicts the trajectory of the preceding vehicle based on an ETA estimator. Because the ETA estimator uses downstream information, it renders the controller non-local.

Define a way point every Δs from $s^0(t)$ to $s^0(t) + l \cdot \Delta s$ at some time $t \in [t_0, t_f]$ and for some spatial receding horizon $l \cdot \Delta s > 0$. Non-integer l indicates that the spatial horizon cannot be evenly divided by Δs. Collect all way points into a sequence, we have $\mathcal{W}(t) := (w_1(t), w_2(t), \ldots, w_{l+1}(t)) = (s^0(t) + i \cdot \Delta s \mid i = 0, 1, \ldots, l)$. For convenience, we assume $w_{l+1} \leq w_f$.

Define $\widehat{\mathfrak{T}}^0(t)$ as the corresponding ETA set for the preceding vehicle, that is, $\widehat{\mathfrak{T}}^0(t) := (\hat{t}_1^0(t), \hat{t}_2^0(t), \ldots, \hat{t}_{l+1}^0(t))$, where $t_i^0, \hat{t}_i^0$ are the *true* and *estimated* arrival times of the preceding vehicle at the way point $w_i \in \mathcal{W}(t)$. For the problem to be well-posed, we require both true and estimated arrival time sequences to be strictly increasing.

We assume there exists an ETA estimator[1] $f : \mathcal{W}(t) \to \widehat{\mathfrak{T}}^0(t)$ that estimates the *arrival* time of the preceding vehicle at each way point in $\mathcal{W}(t)$. To model prediction errors, we assume

---

[1]Common ETA service providers include Google Maps, Waze, INRIX, and Garmin.
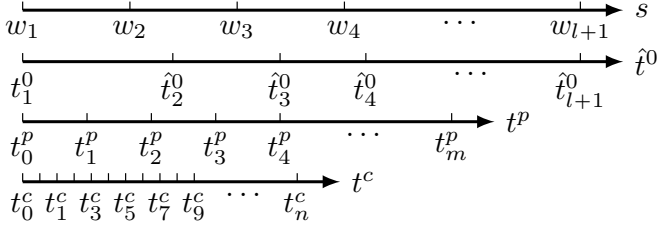
Fig. 3: Specification of the receding-horizon control scheme. The prediction layer has a *fixed* spatial horizon and a *variable* temporal horizon. The planning layer has a temporal horizon m. The tracking layer has a fixed temporal horizon n.
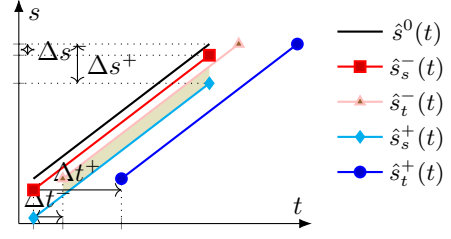

Fig. 4: Headway envelope design. $\hat{s}^0(\cdot)$ is the predicted trajectory of the proceeding vehicle. $\hat{s}_s^-(\cdot), \hat{s}_s^+(\cdot)$ are estimated minimum and maximum *space* headway envelopes, respectively. $\hat{s}_t^-(\cdot), \hat{s}_t^+(\cdot)$ are estimated minimum and maximum *time* headway envelopes, respectively. The shaded region represents the feasible set in space and time.

that the relative estimation error is *uniformly* distributed with an average at one and a radius of σ, that is,

$$\frac{\hat{t}_{i+1}^0 - \hat{t}_i^0}{t_{i+1}^0 - t_i^0} \sim \mathcal{U}(1 - \sigma, 1 + \sigma), \quad i = 1, 2, \ldots, l,$$

where $\hat{t}_1^0 = t_1^0$. We use f_σ to denote that an ETA estimator f has an error radius of σ.

With $\mathcal{W}(t)$ and $\widehat{\mathcal{T}}^0(t) = f_\sigma(\mathcal{W}(t))$, we can then generate a *predicted* trajectory for the preceding vehicle $\hat{s}^0 : [t_1^0, \hat{t}_l^0] \to [s^0(t), s^0(t) + l \cdot \Delta s]$ using a standard interpolation method. The above receding horizon design is illustrated in the s and $\hat{t}^0$ axes of Figure 3.

B. Planning layer

The planning layer plans a reference trajectory for the considered vehicle based on the predicted trajectory of the preceding vehicle.

With $\hat{s}^0(t)$ predicted by f_σ, we can derive the following four trajectories: *1)* $\hat{s}_t^-(t) := \hat{s}^0(t - \Delta t^-)$: predicted envelope of some minimal time headway Δt^-, *2)* $\hat{s}_t^+(t) := \hat{s}^0(t - \Delta t^+)$: predicted envelope of some maximal time headway Δt^+, *3)* $\hat{s}_s^-(t) := \hat{s}^0(t) - \Delta s^-$: predicted envelope of some minimal space headway Δs^-, and *4)* $\hat{s}_s^+(t) := \hat{s}^0(t) - \Delta s^+$: predicted envelope of some maximal space headway Δs^+. The above four trajectories are illustrated in Figure 4.

While many possible designs are possible, in this study we select the minimal and maximal headway envelopes as follows

$$
\begin{aligned}
\hat{s}_{\min}\left(t \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right) &:= \max\left(\min\left(\hat{s}_s^-(t), \hat{s}_t^-(t)\right), \hat{s}_s^+(t)\right), \\
\hat{s}_{\max}\left(t \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right) &:= \min\left(\max\left(\hat{s}_s^+(t), \hat{s}_t^+(t)\right), \hat{s}_s^-(t)\right).
\end{aligned}
\tag{1}
$$

The area between $\hat{s}_{\min}(t)$ and $\hat{s}_{\max}(t)$ is shaded in green in Figure 4. Consequently, the minimum and maximum headways are

$$
\begin{aligned}
h_{\min}(t) &= \hat{s}^0(t) - \hat{s}_{\min}\left(t \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right), \\
h_{\max}(t) &= \hat{s}^0(t) - \hat{s}_{\max}\left(t \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right).
\end{aligned}
\tag{2}
$$

The objective of the design in (1) is to generate smooth car-following without falling far behind nor getting too close. Compared to [9], [10], our formulation is stricter, which leads to a tighter yet still safe and smooth car-following.

Now, let m be the temporal planning horizon and Δt_p be the temporal planning resolution. For all $i_p = 0, \ldots, m$, denote

$t_{i_p}^p := t_1^0 + i_p \Delta t_p$. Non-integer m indicates that $\hat{t}_m^0 - t_1^0$ cannot be evenly divided by Δt_p.

Define the planned states $\check{\mathcal{X}}^{1p}$ and the planned accelerations $\check{\mathcal{U}}^{1p}$ of the considered vehicle as follows:

$$
\begin{aligned}
\check{\mathcal{X}}_m^{1p} &:= \{\check{\mathbf{x}}^1(t_0^p), \check{\mathbf{x}}^1(t_1^p), \ldots, \check{\mathbf{x}}^1(t_m^p))\}, \\
\check{\mathcal{U}}_{m-1}^{1p} &:= \{\check{a}^1(t_0^p), \check{a}^1(t_1^p), \ldots, \check{a}^1(t_{m-1}^p)\}.
\end{aligned}
\tag{3}
$$

Here, we introduce the check accent $\check{x}$ to denote that a variable x is a *planned* decision variable.

With everything defined above, we plan for the states and accelerations of the considered vehicle via the following LCQP:

$$
\min_X \quad \alpha \sum_{i_p=0}^{m-1} \left(\check{a}^1(t_{i_p}^p)\right)^2 + \beta \sum_{j_p=1}^m \left(\xi_{j_p}^p\right)^2 + \gamma \sum_{j_p=1}^m \left(\zeta_{j_p}^p\right)^2 \tag{4a}
$$

$$\text{s.t.} \quad \check{\mathbf{x}}^1(t_0^p) = \mathbf{x}_0^1, \tag{4b}$$

$$\check{\mathbf{x}}^1(t_{i_p+1}^p) = \mathbf{A}_{i_p}^p \cdot \check{\mathbf{x}}^1(t_{i_p}^p) + \mathbf{B}_{i_p}^p \cdot \check{a}^1(t_{i_p}^p), \tag{4c}$$

$$0 \leqslant \xi_{j_p}^p, \ \check{s}^1(t_{j_p}) - \hat{s}_{\min}\left(t_{j_p} \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right) \leqslant \xi_{j_p}^p, \tag{4d}$$

$$0 \leqslant \zeta_{j_p}^p, \ -\check{s}^1(t_{j_p}) + \hat{s}_{\max}\left(t_{j_p} \mid \mathcal{W}, \widehat{\mathcal{T}}^0\right) \leqslant \zeta_{j_p}^p, \tag{4e}$$

$$v_{\min} \leqslant \check{v}^1(t_{j_p}^p) \leqslant v_{\max}, \tag{4f}$$

$$a_{\min} \leqslant \check{a}^1(t_{i_p}^p) \leqslant a_{\max}, \tag{4g}$$

where

$$X := \left\{\check{\mathcal{X}}_m^{1p}, \check{\mathcal{U}}_{m-1}^{1p}, \{\xi_{j_p}^p\}_1^m, \{\zeta_{j_p}^p\}_1^m\right\},$$

$$\mathbf{A}_{i_p}^p := e^{\mathbf{A}(t_{i_p+1}^p - t_{i_p}^p)}, \quad \mathbf{A} = \left[[0 \ 0]^\top \ [1 \ 0]^\top\right],$$

$$\mathbf{B}_{i_p}^p := \int_{t_{i_p}^p}^{t_{i_p+1}^p} e^{\mathbf{A}(t_{i_p+1}^p - \tau)} d\tau \cdot \mathbf{B}, \quad \mathbf{B} = [0 \ 1]^\top,$$

for $i_p = 0, \ldots, m-1$ and $j_p = 1, \ldots, m$. The receding horizon design is illustrated in the t^p axis of Figure 3. We have $\mathcal{W}, \widehat{\mathcal{T}}^0, \alpha, \beta, \gamma$, and $\mathbf{x}_0^1$ as the inputs to the optimization program, where $\alpha, \beta, \gamma \in \mathbb{R}_{>0}, \alpha + \beta + \gamma = 1$, and $\mathbf{x}_0^1$ is the initial state of the considered vehicle.

In (4a), the first term penalizes large accelerations, while the second and third term, coupled with (4d) and (4e), regulate the vehicle inside the admissible space headway ranges. Constraints (4b) and (4c) impose kinematic constraints using

zero-order hold. Lastly, constraints (4f) and (4g) impose speed and acceleration limits.

Proposition 1. *If the feasible set is nonempty, problem* (4) *has unique global optimum.*

Proof. Because $\alpha, \beta, \gamma > 0$, the objective function (4a) is strongly convex. By construction, the feasible set is a polyhedron, which is also convex. Therefore, a direct application of Lemma 8.2 and Theorem 8.6 in [15] reveals that if the feasible set is nonempty, problem (4) has an unique global optimum. □

Proposition 2. *For some initial time t_0, if the feasible set of problem* (4) *is nonempty, then its corresponding receding-horizon policy has a solution for all $t > t_0$, that is, it is persistently feasible.*

Proof. To show that problem (4) is persistently feasible, it suffices to show that it has a solution at every future time step. Because the feasible set is initially nonempty, the problem has a solution at t_0. For every other time step $t > t_0$, it is easy to verify that zero acceleration is a feasible point due to the fact that constraints (4d) and (4e) are soft. Therefore, it follows that problem (4a), when repeatedly applied in receding horizon, always stays feasible. □

Remark 1. *When the initial velocity is too high or too low, the feasible set of* (4) *will become empty. For example, when the initial velocity is greater than $v_{\max} + a_{\max}\Delta t^p$, there will be no acceleration within the actuation limits that can steer the velocity below $v_{\max}$ in the next time step. To prevent any chance of infeasibility, we can replace the hard constraints on velocity with suitable soft constraints.*

Remark 2. *Because the constraints on minimum headway are soft, it is possible for the considered vehicle to violate constraint* (4d). *Therefore, it is theoretically permissible for the preceding vehicle to collide with the preceding vehicle. Note that this is chosen by design because the hard constraint version of* (4d) *can never be guaranteed in practice due to irregularity of human driving and inevitable errors in prediction. Therefore, the presented soft constraints in* (4d) *should be viewed as a "best-effort" attempt to respect the minimum headway requirement.*

Furthermore, we briefly comment on how constraints (4d) and (4e) can be extended to account for cut-ins and cut-outs. For example, a possible design is illustrated in Figure 5. Under this design, the considered vehicle will not overreact when a discontinuous change occurs in the position of the preceding vehicle. Rather, it gradually recovers to a comfortable headway over time. With the help a switching condition, this policy can be combined with that of (1) to form a hybrid MPC controller to perform car-following with cut-ins and cut-outs.

C. Tracking layer

To handle prediction errors, an additional tracking layer is introduced to track the planned accelerations and to guard the considered vehicle from imminent collisions.
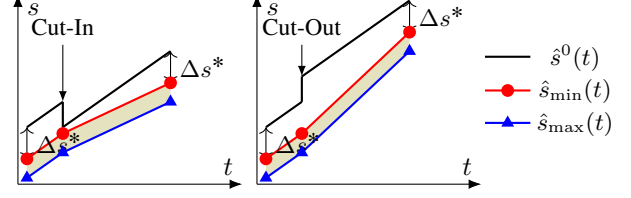

Fig. 5: Headway envelope design to handle cut-ins and cut-outs. Left design is for cut-ins and right for cut-outs. The proposed designs aim to to keep a comfortable headway Δs^* without overreaction.

Concretely, the layer takes the following form, which is a modified version of (4).

$$\min_{Z} \quad \lambda \sum_{i_c=0}^{n-1} \left(\check{a}^1(t_{i_c}^c) - \bar{a}^1(t_{i_c}^c) \right)^2 + \mu \sum_{j_c=1}^{n} \left(\xi_{j_c}^c \right)^2 \tag{5a}$$

$$\text{s.t.} \quad \bar{\mathbf{x}}^1(t_0^c) = \mathbf{x}_0^1, \tag{5b}$$

$$\bar{\mathbf{x}}^1(t_{i_c+1}^c) = \mathbf{A}_{i_c}^c \cdot \bar{\mathbf{x}}^1(t_{i_c}^c) + \mathbf{B}_{i_c}^c \cdot \bar{a}^1(t_{i_c}^c), \tag{5c}$$

$$0 \leqslant \xi_{j_c}^c, \quad s^1(t_{j_c}^c) - \tilde{s}_{\min}\left(t_{j_c}^c \mid a^0(t_0^c)\right) \leqslant \xi_{j_c}^c, \tag{5d}$$

$$v_{\min} \leqslant \bar{v}^1(t_{j_c}^c) \leqslant v_{\max}, \tag{5e}$$

$$a_{\min} \leqslant \bar{a}^1(t_{i_c}^c) \leqslant a_{\max}, \tag{5f}$$

where

$$t_{i_c}^c := t + i_c \Delta t^c, \quad Z := \left\{ \overline{\mathcal{X}}_n^{1c}, \overline{\mathcal{U}}_{n-1}^{1c}, \{\xi_{j_c}^c\}_1^n \right\},$$

$$\mathbf{A}_{i_c}^c := e^{\mathbf{A}(t_{i_c+1}^c - t_{i_c}^c)}, \quad \mathbf{B}_{i_c}^c := \int_{t_{i_c}^c}^{t_{i_c+1}^c} e^{\mathbf{A}(t_{i_c+1}^c - \tau)} d\tau \cdot \mathbf{B},$$

for some horizon n, resolution Δt^c, $i_c = 0, \ldots, n-1$, and $j_c = 1, \ldots, n$. $\overline{\mathcal{X}}_n^{1c}, \overline{\mathcal{U}}_{n-1}^{1c}$ are defined similar to those in (3). Note that here we use accent $\bar{x}$ to indicate that a variable x is an internal variable of the tracking controller. The receding horizon design is illustrated in the t^c axis of Figure 3. Unlike the planning layer, the minimum headway envelope $\tilde{s}_{\min}(\cdot \mid a^0(\cdot))$ is generated by assuming that the preceding vehicle accelerates *constantly* from t_0^c to t_n^c. Similar to the planning layer once again, we have $a^0(t_0)$, λ, μ, and $\mathbf{x}_0^1$ as inputs to the optimization program, where $\lambda, \mu \in \mathbb{R}_{>0}, \lambda + \mu = 1$.

The modified LCQP in the tracking layer in (5) is structurally identical to (4) except for: *1)* we remove the soft constraint on maximum headway; *2)* we adopt a shorter planning horizon n and a smaller temporal resolution Δt^c; and *3)* we predict the trajectory of the preceding vehicle by assuming that it keeps its current acceleration for the entire tracking horizon.

Note that the last change above is made to ensure safety. Because constraints (5d) only takes vehicular RADAR measurements, which are generally highly reliable, the collision avoidance of the car-following is mostly *decoupled* from the ETA estimation. Hence, a bad prediction from the ETA estimator cannot severely impact the safety of the vehicle.

Remark 3. *Because the optimization problem* (5) *may be viewed as a special case of problem* (4) *where $\gamma = 0$, it has the same properties as those highlighted in Propositions 1,2 and Remarks 1,2.*

Remark 4. *We assume that we can directly control the acceleration of the vehicle. Consequently, dynamic constraints* (5c) *are purely kinematic.*

Lastly, we briefly describe how our proposed method can be extended for multi-vehicle platooning. For example, consider a platoon of vehicles with indices $0, 1, \ldots, k$, where index 0 indicates the preceding human vehicle and indices $1, \ldots, k$ indicate the k following automated vehicles. As before, we can use an ETA estimator to generate a prediction for vehicle zero. For all the vehicles that *follow*, we can substitute the predicted trajectories of the preceding vehicles by the corresponding planned trajectories. That is, for $i = 1, 2, \ldots, k$, we can use $\check{a}^{i-1}(t)$ to compute $\check{s}^{i-1}(t)$ and approximate $\hat{s}^{i-1}(t)$ with $\check{s}^{i-1}(t)$. Integrating the above prediction method with the planning and tracking methods presented in this paper, we have consequently produced a complete platoon controller design.

IV. NUMERICAL SIMULATION

To evaluate the proposed controller, we test its performance through an array of numerical experiments, each equipped with an ETA estimator of an unique combination of spatial resolution Δs and noise level σ.

For all simulations, we emulate the preceding vehicle by replaying a recorded drive from [16]. The velocity and acceleration of the recorded drive are shown in the second and third rows of Figure 6.

Spatial horizon of the ETA estimators are fixed to $l \cdot \Delta s = 3000\,\text{m}$. We run an array of simulations by varying the spatial resolution Δs between $10\,\text{m}$ and $500\,\text{m}$ and noise level σ between 0.01 and 0.25. To establish a performance baseline, we run an additional simulation with an intelligent driver model (IDM) controller [17]. Parameters of the IDM are chosen to roughly match that of the MPC controller: $a = 1.5, b = 3, \delta = 4, s_0 = 3.5, \ell = 4.65$. Likewise, to establish a performance upper bound, we run one more simulation with a controller that can perfectly foresee the trajectory of the preceding vehicle. We call this controller the oracle controller.

Shared parameters of the planning and tracking layers are: $v_{\min} = 0, v_{\max} = 35\,\text{m s}^{-1}, a_{\min} = -1.5\,\text{m s}^{-2}, a_{\max} = 3\,\text{m s}^{-2}, \Delta s^- = 5\,\text{m}, \Delta s^+ = 100\,\text{m}, \Delta t^- = 0.6\,\text{s},$ and $\Delta t^+ = 3.0\,\text{s}$. Unique parameters of the planning layer are: $\alpha = 0.2, \beta = 0.7, \gamma = 0.1, \Delta t^p = 1\,\text{s},$ and $m = 60$. Unique parameters of the tracking layer are: $\lambda = 0.1, \mu = 0.9, \Delta t^c = 0.1\,\text{s},$ and $n = 30$.

To evaluate performance, we use two measures e and f to quantify tracking error and fuel saving, respectively. The measure e is defined to be the standard deviation of $\check{v}^{1*} - \bar{v}^1$ for all sampled time instants, where $\check{v}^{1*}$ denotes the planned velocity under *perfect* prediction. The measure f is defined to be the ratio of the fuel consumption of the baseline IDM controller to that of the MPC controller, using the energy model described in [18].

Position, velocity, and acceleration of the oracle controller are shown in Figure 6. As the oracle vehicle has demonstrated in Figure 6, the optimal behavior is to avoid accelerating or decelerating unless there is an imminent violation of at least one of the two headway constraints.
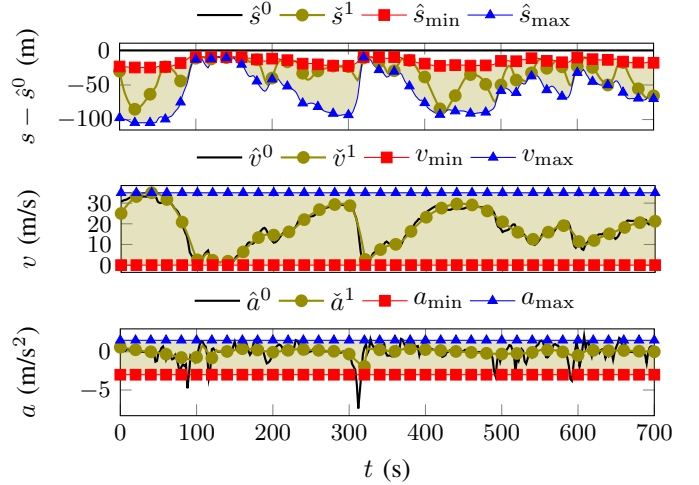

Fig. 6: Demonstration of the planning layer. The oracle controller assumes perfect prediction over the entire planning horizon.
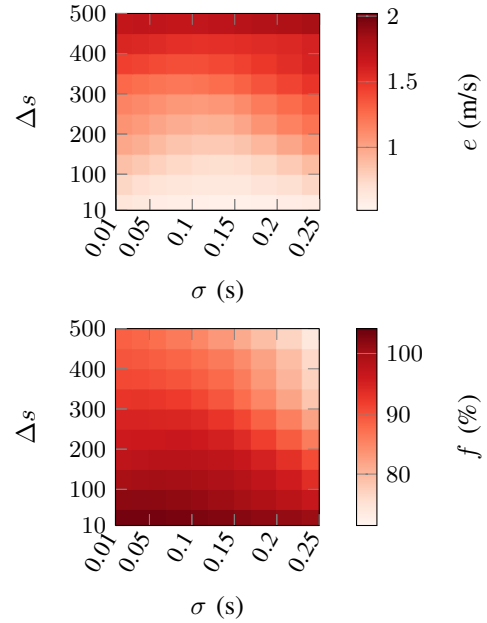

Fig. 7: Performance heat maps of the MPC controllers. The top heat map measures tracking error e, while the bottom heat map measures fuel saving f.

Tracking error and fuel saving are visualized in the top and bottom heat maps in Figure 7. As expected, tracking error e is proportional to spatial resolution Δs and noise level σ. Likewise, fuel saving f is *inversely* proportional to spatial resolution Δs and noise level σ. Note that both performance measures platoon near the origin, indicating that the MPC controller is robust to prediction errors and especially so to that induced by σ.

Lastly, we comment on the running times of the planning and tracking layers. The optimization solver is part of a python package called CVXOPT [19]. Benchmarked on an Intel i7-6700K CPU, the average running time of the planning layer

is about 0.12 s and of the tracking layer about 0.04 s. From Figure 2, we find that both times fit comfortably into their allocated time budgets.

V. Discussions

In this section, we discuss two of the most consequential features in the design of the controller, namely, *1)* the space between the maximum and minimum headway constraints and *2)* the resolution and noise level of the ETA estimators.

It is clear that the larger the space between the maximum and minimum headway constraints, the smoother the car-following could be. For example, if the considered car is allowed to be arbitrarily far behind the preceding vehicle without incurring any penalties, it can simply wait for the preceding vehicle to exit the road and then accelerate to a very small constant velocity to complete the trip. Nevertheless, we know from common sense that such behavior is not acceptable in most of the real-world applications.

On the contrary, when the space between the two headway constraints is small, the MPC controller will become sensitive to prediction errors. Consider a situation where the allowable headway gap is thin and the prediction errors are significant. Because the considered vehicle is already close to the boundaries of the headway constraints, an error in prediction could easily mislead the planning layer to falsely believe that it will soon violate one of the headway constraints. In an effort to steer away from imminent constraint violation, the planning layer overreacts, leading to undesirably large acceleration or deceleration.

To resolve the above problem, one could modify the headway constraints proposed in (1). One possible solution is to enforce a minimum space gap between the maximum and minimum headway envelopes and to add a small penalty to encourage the considered vehicle to drive at the center of the allowable headway constraints.

Last but not least, we comment on how to choose an ETA estimator. As illustrated in Figure 7, the performance of the controller changes significantly with the resolution and noise level of the ETA estimator. In practice, one may first define a desirable tracking error e_{des} and a desirable fuel saving f_{des}. With e_{des} and f_{des} defined, one can then draw two level sets in the two heat maps of Figure 7. Any ETA estimator with e and f that simultaneously fall within the two level sets may be deemed sufficient to meet the design specifications.

VI. Conclusions

In this article, we propose a hierarchical MPC control scheme based on a LCQP. We show via simulations that the controller can achieve smooth and tight car-following and is robust to prediction errors. Additional constructions can be added to further enhance its robustness to erroneous forecasts. Possible future works include field tests with various ETA estimators, extension to handle cut-ins and cut-outs, and modification for platooning.

References

[1] Yuki Sugiyama, Minoru Fukui, et al. Traffic jams without bottle-necks—experimental evidence for the physical mechanism of the formation of a jam. *New Journal of Physics*, 10(3):033001, 2008.

[2] Shin-ichi Tadaki, Macoto Kikuchi, et al. Phase transition in traffic jam experiment on a circuit. *New Journal of Physics*, 15(10):103034, 2013.

[3] John-Jairo Martinez and Carlos Canudas-de Wit. A safe longitudinal control for adaptive cruise control and stop-and-go scenarios. *IEEE Transactions on Control Systems Technology*, 15(2):246–258, 2007.

[4] Arne Kesting, Martin Treiber, Martin Schönhof, and Dirk Helbing. Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C: Emerging Technologies*, 16(6):668–683, 2008.

[5] Li-hua Luo, Hong Liu, Ping Li, and Hui Wang. Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following. *Journal of Zhejiang University SCIENCE A*, 11(3):191–201, 2010.

[6] Boris S Kerner. Physics of automated driving in framework of three-phase traffic theory. *Physical Review E*, 97(4):042303, 2018.

[7] Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221, 2018.

[8] Fanping Bu, Han-Shue Tan, and Jihua Huang. Design and field testing of a cooperative adaptive cruise control system. In *Proceedings of the 2010 American Control Conference*, pages 4616–4621. IEEE, 2010.

[9] Roman Schmied, Harald Waschl, Rien Quirynen, Moritz Diehl, and Luigi del Re. Nonlinear mpc for emission efficient cooperative adaptive cruise control. *IFAC-PapersOnLine*, 48(23):160–165, 2015.

[10] Yanzhao Jia, Tomohiro Saito, Yutaro Itoh, Yernar Nukezhanov, and Daniel Görges. Energy-optimal adaptive cruise control in time domain based on model predictive control. *IFAC-PapersOnLine*, 51(31):846–853, 2018.

[11] Eunjeong Hyeon, Youngki Kim, Niket Prakash, and Anna G Stefanopoulou. Influence of speed forecasting on the performance of ecological adaptive cruise control. In *Dynamic Systems and Control Conference*, volume 59148, page V001T08A003. American Society of Mechanical Engineers, 2019.

[12] Andreas A Malikopoulos, Christos G Cassandras, and Yue J Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93:244–256, 2018.

[13] Shashwat Shivam, Yorai Wardi, Magnus Egerstedt, Aris Kanellopoulos, and Kyriakos G Vamvoudakis. Intersection-traffic control of autonomous vehicles using newton-raphson flows and barrier functions. *IFAC-PapersOnLine*, 53(2):15733–15738, 2020.

[14] Eunjeong Hyeon, Youngki Kim, Niket Prakash, and Anna G Stefanopoulou. Short-term speed forecasting using vehicle wireless communications. In *2019 American Control Conference (ACC)*, pages 736–741. IEEE, 2019.

[15] Giuseppe C Calafiore and Laurent El Ghaoui. *Optimization Models*. Cambridge University Press, 2014.

[16] Matthew Nice, Nathan Lichtle, et al. The I-24 Trajectory Dataset, September 2021.

[17] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.

[18] Jonathan Lee, George Gunter, Rabie Ramadan, et al. Integrated framework of vehicle dynamics, instabilities, energy models, and sparse flow smoothing controllers. In *DI-CPS'21: Proceedings of the Workshop on Data-Driven and Intelligent Cyber-Physical Systems*, New York, NY, USA, 2021. Association for Computing Machinery.

[19] Martin Andersen, Joachim Dahl, and Lieven Vandenberghe. CVXOPT: Convex optimization. *Astrophysics Source Code Library*, pages ascl–2008, 2020.