Fuel-efficient heavy-duty vehicle platooning by look-ahead control

Valerio Turri, Bart Besselink, Jonas Mårtensson, Karl H. Johansson

Abstract— The operation of groups of heavy-duty vehicles at close intervehicular distances (known as platoons) has been shown to be an effective way of reducing fuel consumption. For single vehicles, it is also known that the availability of preview information on the road topography can be exploited to obtain fuel savings. The current paper aims at the inclusion of preview information in platooning by introducing a two-layer control system architecture for so-called look-ahead platooning. Here, the layers are responsible for the inclusion of preview information and real-time vehicle control for platooning, respectively. Within this framework, a control strategy is presented, where dynamic programming is used for the calculation of fuel-optimal speed profiles, while a model predictive control approach is exploited for the real-time vehicle control. The feasibility of this approach is illustrated by means of the simulation of relevant scenarios.

I. INTRODUCTION

The transportation of goods is fundamental to the global economy and the demand for road freight transport is expected to increase. However, the road transport sector is also responsible for a large part of the world's energy consumption and greenhouse gas emissions. For example, road transport amounts to roughly 27% of the energy consumption of the European Union [8]. As a result, there is a large research effort aimed at the reduction of fuel consumption for heavy-duty vehicles (HDVs), which provides a clear economical benefit for haulage companies as well.

An effective method for the reduction of fuel consumption is given by heavy-duty vehicle platooning. By operating groups of vehicles at close inter-vehicular distances (known as platoons), the overall aerodynamic drag can be reduced. As about one fourth of the fuel consumption of a heavy-duty vehicle is related to aerodynamic drag, this can have a large effect on the fuel consumption. Indeed, experimental results in [1] and [4] have shown a reduction in fuel consumption of up to about 7%. Apart from savings in fuel consumption, platooning also has the potential to increase road capacity and enhance traffic flow [17]. However, in order to safely operate heavy-duty vehicles at the short inter-vehicular distances required for platooning, automation of the longitudinal dynamics is necessary. This control problem was first studied in [12] and many results have appeared since, see, e.g., [14], [7], [13].

These existing works on platooning typically focus on inter-vehicular control and do neither take the environment into account nor explicitly focus on the reduction of fuel consumption. However, due to the large mass and limited

The authors are with the ACCESS Linnaeus Centre and Department of Automatic Control, KTH Royal Institute of Technology, Stockholm, Sweden, email: turri@kth.se, bart.besselink@ee.kth.se, jonas1@kth.se, kallej@kth.se.

power of heavy-duty vehicles, the road slope has a large influence on admissible speed profiles and, consequently, on fuel consumption. For single vehicles, it is shown in [9] (see also [15] for some early work) that the availability of preview information about the road topography allows for the optimization of the speed profile to save fuel. In fact, experimental results on this so-called look-ahead control have shown a fuel reduction of about 3.5% without increasing trip time [10].

The huge variety of possible engine powers and vehicle masses, however, would require the vehicles to follow significant different optimal speed profiles. In order to fully exploit the potential for reducing platoon fuel consumption, therefore, it becomes intuitive to combine the benefits of platooning and look-ahead control by introducing some level of coordination between vehicles. This problem has not received much attention in the literature, but fuel-efficient approaches for look-ahead control for vehicle platooning have been studied in [2]. There, it is shown that existing look-ahead control strategies for single vehicles are not necessarily suitable for platoons and that a dedicated approach is required. However, [2] focuses on look-ahead control only and does not explicitly include the inter-vehicular control for platooning.

In the current paper, an approach towards look-ahead control for heavy-duty vehicle platooning is presented, which includes both look-ahead and inter-vehicle control for platooning. This leads to the following contributions.

First, a control architecture for look-ahead control for platooning is presented. This architecture presents a decomposition of the overall problem into two layers, which are called the platoon coordinator layer and the vehicle trajectory tracking layer. Here, the coordinator layer is responsible for generating a fuel-optimal speed profile for the entire platoon, hereby taking preview information on the road topography into account. This information is taken as a reference for the trajectory tracking layer, which performs the real-time vehicle control needed to track this speed profile as well as guarantee the desired inter-vehicular distances.

Second, a predictive control strategy is developed within this control architecture. Here, the coordinator layer relies on a dynamic programming (DP) approach [3] to compute a fuel-optimal speed profile for the platoon, based on preview road topography information. In this approach, the desired speed profile is specified as a function of space (i.e., the position on the road). By combining this with a desired inter-vehicular distance specified as a pure time gap, it is ensured that this *single* speed profile can be tracked by every heavy-duty vehicle. Stated differently, there is no need to compute optimal speed profiles for each vehicle separately. Finally, it
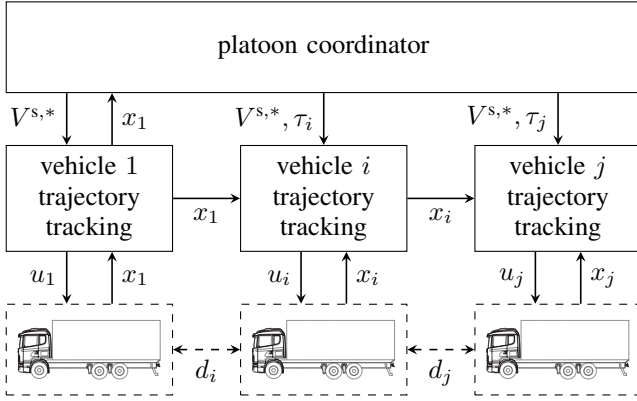
Fig. 1. System architecture for look-ahead platooning.

is noted that this trajectory is constructed such that it satisfies the constraints (e.g., on available power or braking capacity) of each vehicle and it is optimal for the whole platoon.

The trajectory tracking layer consists out of a model predictive controller (MPC) [5], which naturally exploits the available preview information from the coordinator layer. Moreover, this approach allows for the inclusion of constraints on both the inputs and state, where the latter is used to ensure safety by guaranteeing that following heavy-duty vehicles do not collide.

The effectiveness of this control architecture and predictive control strategy is confirmed by simulations, hereby considering both a route over a hill and an emergency braking scenario.

The rest of this paper is organized as follows. The two-layer architecture for look-ahead control for platooning will be introduced in Section II, whereas the vehicle and platoon models are discussed in Section III. The platoon coordinator and trajectory tracking controllers are discussed in Section IV and Section V, respectively, and their performance is evaluated by means of simulations in Section VI. Finally, conclusions are stated in Section VII.

II. CONTROL ARCHITECTURE

In order to decompose the look-ahead platooning control problem into manageable subproblems, the architecture in Figure 1 is introduced. This architecture consists of two layers, being the platoon coordinator layer and the vehicle trajectory tracking layer.

The coordinator layer exploits available information on the topography of the planned route to find a fuel-optimal speed profile for the entire platoon. Hereby, it typically considers a horizon of several kilometers and takes the constraints of all vehicles in the platoon into account. As a result, it can be guaranteed that every vehicle in the platoon is able to track the required speed profile. Moreover, it is stressed that a single speed trajectory is computed, representing the speed of the platoon. However, when this speed profile is specified as a function of space (i.e., position on the road) and the inter-vehicle spacing is chosen according to a pure time delay, it can be shown that every individual vehicle in the platoon

can track this single speed profile. It is remarked that this layer can typically operate in a receding horizon fashion, providing an updated speed profile roughly every 10 seconds or when the recalculation is needed due to a strong deviation from the original profile. Finally, as this layer influences the entire platoon, it would typically be implemented at the lead vehicle of a platoon. In Section IV, a dynamic programming approach towards this problem is discussed.

The trajectory tracking layer is responsible for the real-time vehicle control, which is aimed at tracking the desired speed profile as resulting from the coordinator layer. It also exploits measurements of the inter-vehicular distances between heavy-duty vehicles to ensure the proper spacing strategy. Moreover, this layer should guarantee safety of platooning operations and avoid collisions between trucks. Therefore, this layer is physically implemented on each individual heavy-duty vehicle and operates in real-time, using instantaneous measurements and potentially a short predictive horizon. This latter approach is taken in the current paper, where a model predictive controller is developed in Section V.

The presented control architecture requires the platoon coordinator layer implemented in the leading vehicle to send information (i.e., reference speed trajectory and time gaps) with slow refresh time to the trajectory tracking layers of the follower vehicles. This communication relays on wireless vehicle-to-vehicle (V2V) communication. However it is not an aim of this paper to discuss requirements on the specific V2V communication technology.

III. VEHICLE AND PLATOON MODELS

In this section we present the general models of the single vehicle and of the platoon that will be used in the formulation of the two control layers. Since the aim of these models is to capture the components that play a major role in the fuel consumption, we will focus on the longitudinal dynamics only.

Using Newton's second law, the longitudinal dynamics of a single vehicle can be expressed by:

$$\begin{aligned} m_i \dot{v}_i =& F_{\text{engine},i} - F_{\text{brake},i} - F_{\text{gravity},i}(\alpha(s_i)) \\ & - F_{\text{roll},i}(\alpha(s_i)) - F_{\text{drag},i}(v_i, d_i), \\ \dot{s}_i =& v_i, \end{aligned} \tag{1}$$

where v_i and s_i denote the states of vehicle i, respectively the speed and the longitudinal position, m_i denotes its mass and $F_{\text{engine},i}$, $F_{\text{brake},i}$, $F_{\text{gravity},i}$, $F_{\text{roll},i}$ and $F_{\text{drag},i}$ denote the forces acting along the longitudinal direction [1]. More in detail, $F_{\text{engine},i}$ and $F_{\text{brake},i}$ represent the vehicle inputs and are the forces generated respectively by the powertrain and by the braking system; $F_{\text{gravity},i}$ is the force generated by the gravity and is given as $F_{\text{gravity},i}(\alpha(s_i)) = m_i g \sin(\alpha(s_i))$, where g is the gravity acceleration and $\alpha(s_i)$ is the road slope at position s_i; $F_{\text{roll},i}$ represents the rolling resistance and is defined as $F_{\text{roll},i}(\alpha(s_i)) = c_r m_i g \cos(\alpha(s_i))$, where c_r is the rolling coefficient. Finally, $F_{\text{drag},i}$ is the aerodynamic resistance and is defined as $F_{\text{drag},i}(v_i, d_i) = \frac{1}{2}\rho A_v C_D(d_i)v_i^2$, where ρ is the air density, A_v is the cross-sectional area of the vehicle, C_D is the airdrag coefficient and d_i is the distance between

vehicle i and the preceding one (i.e., $d_i = s_{i-1} - s_i - l_{i-1}$ where l_i is the length of vehicle i for $i \geq 2$ and $d_i = \infty$ for $i = 1$). In our model, the airdrag coefficient C_D depends only on the distance to the preceding vehicle because of its dominant role and it is defined as:

$$C_D(d_i) = C_{D_0}\left(1 - \frac{C_{D_1}}{C_{D_2} + d_i}\right), \tag{2}$$

where C_{D_0}, C_{D_1} and C_{D_2} are three parameters obtained approximating the experimental data presented in [11]. Note that the dependence of the airdrag coefficient on the distance plays an essential role in the platooning concept. In fact, by decreasing the distance between vehicles, the aerodynamic forces and consequently the fuel consumption are reduced.

The complete model of a platoon of N vehicles can be now defined by equations (1) for $i = 1, ..., N$, combined with the distance definition.

We finally introduce the powertrian model used in the platoon coordinator layer. The powertrain force has been modeled by

$$F_{\text{engine},i} = \frac{H_e}{v_i}\gamma_i, \tag{3}$$

where γ_i denotes the engine fuel flow and H_e denotes the energy obtained at the wheels-level per each mass-unit of fuel. For simplicity, a constant efficiency engine is assumed and gear changes are not taken into account.

IV. PLATOON COORDINATOR

This section focuses on the control routine running in the platoon coordinator layer.

As shown in Figure 1, the platoon coordinator layer is the higher layer of the control architecture and it considers the whole platoon. It takes as inputs the real-time position $s_1(t)$ and speed $v_1(t)$ of the first vehicle and, by exploiting available information on the planned route (i.e., slope and speed limits), it computes a single feasible and fuel-efficient speed reference profile $V^{\text{s},*}(s)$ defined in the space domain, that all the vehicles within the platoon should follow. The coordinator layer has been implemented through a dynamic programming routine [3] that runs in closed-loop. The parameters that characterize the DP are the discretization space Δs_{DP}, the horizon length H_{DP} and the refresh frequency f_{DP}. We also define the horizon space length as $S_{\text{DP}} = H_{\text{DP}}\Delta s_{\text{DP}}$.

In accordance with the definition of a single space-dependent speed trajectory for the whole platoon, the natural spacing strategy is to use a pure time gap policy, i.e., adjacent vehicles are passing through the same position with a fixed time delay [6]. The computation of τ_i relative to vehicle i is not explicitly performed in this paper and it can be assumed to be given.

In the coming subsections we will introduce the platoon model, the constraints on the input and states, the cost function definition to finally arrive to the complete dynamic programming formulation.

A. Platoon model

The platoon coordinator layer uses a discretized version of the vehicle (1) and the powertrain model (3). Since the slope and the speed limits depends on the position, the discretization has been carried out in the space domain using the implicit Euler approximation. As it will be shown later, the space discretization allows to reduce significantly the computational complexity of the control algorithm. The discretized vehicle model is:

$$\begin{aligned}
v_i^{\text{s}}(\tilde{s}_k)&\frac{v_i^{\text{s}}(\tilde{s}_k) - v_i^{\text{s}}(\tilde{s}_{k-1})}{\Delta s_{\text{DP}}} = \frac{H_e}{v_i^{\text{s}}(\tilde{s}_k)}\gamma_i(\tilde{s}_k) \\
&- m_i g \sin(\alpha(\tilde{s}_k)) - c_r m_i g \cos(\alpha(\tilde{s}_k)) \\
&- \tfrac{1}{2}\rho A_v C_D(d_i^{\text{s}}(\tilde{s}_k))v_i^{\text{s}\,2}(\tilde{s}_k) - F_{\text{brake},i}(\tilde{s}_k),
\end{aligned} \tag{4}$$

where $\tilde{s}_k$ denotes the discretized space, $v_i^{\text{s}}(\tilde{s}_k)$ denotes the speed and $\gamma_i(\tilde{s}_k)$ and $F_{\text{brake},i}(\tilde{s}_k)$ denote respectively the fuel flow and the braking force and, together, define the input vector $u_i^{\text{s}}(\tilde{s}_k) = [\gamma_i(\tilde{s}_k), F_{\text{brake},i}(\tilde{s}_k)]^{\text{T}}$. In the DP formulation we will refer to the model expressed in (4) as $v_i^{\text{s}}(\tilde{s}_{k-1}) = f_{v,i}^{\text{s}}(v_i^{\text{s}}(\tilde{s}_k), u_i^{\text{s}}(\tilde{s}_k))$.

Since (4) is space-dependent, it is not possible to define the inter-vehicular distance for the follower vehicles as function of the state and, therefore, the following approximated expression, as function of the current vehicle speed $v_i^{\text{s}}(\tilde{s}_k)$, has been used: $d_i^{\text{s}}(\tilde{s}_k) = v_i^{\text{s}}(\tilde{s}_k)\tau_i - l_{i-1}$. Note that, because of the DP nature, a more realistic nonlinear characteristic of the powertrain could have been used, without increasing the complexity of the computation.

B. Model constraints

Model (4) has been constrained by introducing bounds on the input and the speed.

1) Input constraints: The fuel flow and the braking force have been bounded by the following constraints:

$$0 \leq \gamma_i(\tilde{s}_k) \leq \frac{P_{\text{max},i}}{H_c}, \tag{5}$$
$$m_i \eta_i g \mu \cos(\alpha(\tilde{s}_k)) \leq F_{\text{brake}_i}(\tilde{s}_k) \leq 0,$$

where $P_{\text{max},i}$ denotes the maximum power at the wheel-level, μ denotes the road friction coefficient and η_i denotes the braking system efficiency. In the DP formulation, we will refer to these constraints as $u_i^{\text{s}}(\tilde{s}_k) \in \mathcal{U}_i^{\text{s}}(\tilde{s}_k)$.

2) State constraints: In order to take the road speed limits into account, the state has been bounded as:

$$v_{\min}(\tilde{s}_k) \leq v_i^{\text{s}}(\tilde{s}_k) \leq v_{\max}(\tilde{s}_k). \tag{6}$$

We will refer to this constraint as $v_i^{\text{s}}(\tilde{s}_k) \in \mathcal{V}^{\text{s}}(\tilde{s}_k)$.

Moreover, in order to require all the vehicle to follow the same speed profile, the constraint

$$v_i^{\text{s}}(\tilde{s}_k) = v^{\text{s}}(\tilde{s}_k), \quad i = 1, ..., N. \tag{7}$$

has been introduced. The inclusion of the constraint (7) significantly reduces the search space dimension as used in the dynamic programming algorithm, enabling fast computation.

C. Cost function

The objective of the platoon coordinator layer is to define an optimal speed profile that minimizes the fuel consumption of the whole platoon, while maintaining a certain average speed. A straightforward approach to handle the average speed requirement is to introduce the time as an additional state and express it as a state constraint. However, because of

the exponential growth of the computational complexity with the number of states, known as the *curse of dimensionality* [3], this formulation cannot be real-time implemented on today's hardware. In order to overcome this obstacle, therefore, we use an approach similar to that one presented in [10]. The cost function is split into two terms, a first term $J_{\text{fuel}}(V^{\text{s}}(\tilde{s}_h)), U^{\text{s}}(\tilde{s}_h))$ representing the effective fuel consumed by the platoon and a second term $J_{\text{time}}(V^{\text{s}}(\tilde{s}_h))$ representing the required time to drive along the horizon; $V^{\text{s}}(\tilde{s}_h) = \{v^{\text{s}}(\tilde{s}_h), ..., v^{\text{s}}(\tilde{s}_{h+H_{\text{DP}}-1})\}$ and $U^{\text{s}}(\tilde{s}_h) = \{u_1^{\text{s}}(\tilde{s}_h), ...u_N^{\text{s}}(\tilde{s}_h), ..., u_1^{\text{s}}(\tilde{s}_{h+H_{\text{DP}}-1}), ..., u_N^{\text{s}}(\tilde{s}_{h+H_{\text{DP}}-1})\}$ denote respectively the set of the speeds and inputs over the horizon. The cost function can be then expressed as:

$$
\begin{aligned}
J_{\text{DP}}(V^{\text{s}}(\tilde{s}_h)), U^{\text{s}}(\tilde{s}_h)) =& J_{\text{fuel}}(V^{\text{s}}(\tilde{s}_h)), U^{\text{s}}(\tilde{s}_h)) \\
&+ \beta J_{\text{time}}(V^{\text{s}}(\tilde{s}_h)),
\end{aligned}
$$

where β is a trade-off parameter, whose computation is described in [10]. The term $J_{\text{fuel}}(V^{\text{s}}(\tilde{s}_h)), U^{\text{s}}(\tilde{s}_h))$ is defined as the total fuel consumed by the HDVs plus an extra component taking in account of the kinematic energy of the vehicles at the end of the horizon:

$$
\begin{aligned}
J_{\text{fuel}}(V^{\text{s}}(\tilde{s}_h)), U^{\text{s}}(\tilde{s}_h)) =& \sum_{k=h}^{h+H_{\text{DP}}-1} \sum_{i=1}^{N} \gamma_i(\tilde{s}_k) \frac{\Delta s_{\text{DP}}}{v^{\text{s}}(\tilde{s}_k)} \\
&- \frac{1}{2H_e} \sum_{i=1}^{N} m_i (v^{\text{s}}(\tilde{s}_{h+H_{\text{DP}}-1}))^2
\end{aligned}
$$

On the other hand, $J_{\text{time}}(V^{\text{s}}(\tilde{s}_h))$ can be expressed as

$$
J_{\text{time}}(V^{\text{s}}(\tilde{s}_h)) = \sum_{k=h}^{h+H_{\text{DP}}-1} \frac{\Delta s_{\text{DP}}}{v^{\text{s}}(\tilde{s}_k)}.
$$

D. Dynamic programming formulation

We now have all the elements to formulate the DP problem:

$$
\min_{U^{\text{s}}(\tilde{s}_h)} J_{\text{DP}}(V^{\text{s}}(\tilde{s}_h), U^{\text{s}}(\tilde{s}_h)) \tag{8a}
$$

$$
\text{subj. to} \quad v_i^{\text{s}}(\tilde{s}_{k-1}) = f_{v,i}^{\text{s}}(v_i^{\text{s}}(\tilde{s}_k), u_i^{\text{s}}(\tilde{s}_k)), \tag{8b}
$$

$$
u_i^{\text{s}}(\tilde{s}_k) \in \mathcal{U}_i^{\text{s}}(\tilde{s}_k), \tag{8c}
$$

$$
v_i^{\text{s}}(\tilde{s}_k) = v^{\text{s}}(\tilde{s}_k) \in \mathcal{V}^{\text{s}}(\tilde{s}_k), \tag{8d}
$$

$$
\tilde{s}_h = s_1(t), \tag{8e}
$$

$$
v^{\text{s}}(\tilde{s}_h) = v_1(t), \tag{8f}
$$

where $k = h, ..., h+H_{\text{DP}}-1$ and the equations (8e) and (8f) represent the initial conditions of the DP formulation.

That constraint (8d) plays an essential role in the DP routine, since it reduces the search space of each DP step to one dimension (i.e., $v^{\text{s}}(\tilde{s}_k)$) and, therefore, it significantly decreases the computational complexity. This allows to solve the DP problem in a reasonable short time. We will refer to the optimal speed trajectory as $V^{\text{s},*}(\tilde{s})$.

V. VEHICLE TRAJECTORY TRACKING

This section focuses on the control routine running in the vehicle trajectory tracking layer.

As shown in Figure 1, the vehicle trajectory tracking layer is a local controller running in each HDV. This layer tracks the speed trajectory and the time gap provided by the platoon coordinator layer and computes the real-time braking and traction forces that lower level controllers will implement. The tracking is implemented through an MPC formulation, which guarantees the safety of the platoon. More in detail the trajectory tracking layer receives the reference speed trajectory $V^{\text{s},*}(s)$ and time gap τ_i from the coordinator layer and the current speed and position of the own and preceding vehicles (respectively $v_i(t)$, $s_i(t)$, $v_{i-1}(t)$ and $s_{i-1}(t)$) from local sensors (i.e., GPS and radar); the outputs are the optimal scaled traction and braking forces $f_{\text{e},i}^*(t)$ and $f_{\text{b},i}^*(t)$.

The parameters that characterize the MPC formulation are the discretization time Δt_{MPC}, the horizon steps number H_{MPC} and the refresh frequency f_{MPC}. We also define the length of the horizon as $T_{\text{MPC}} = H_{\text{MPC}}\Delta t_{\text{MPC}}$.

In the coming subsections we will present the conversion of the reference trajectories from space to time, the vehicle model, the model constraints, the cost function to finally conclude with the complete MPC formulation.

A. Conversion of the speed reference trajectory

Before using the speed reference trajectory $V^{\text{s},*}(s_1(t))$ provided by the platoon coordinator, each vehicle needs to convert it in the time-domain, while taking into account the time gap requirement.

In order to do that each follower vehicle i (i.e., $i \geq 2$) computes the assumed state trajectory of the preceding vehicle $\hat{X}_{i-1}(t) = \{\hat{x}_{i-1}(t), \hat{x}_{i-1}(t + \Delta t_{\text{MPC}}), ..., \hat{x}_{i-1}(t + T_{\text{MPC}})\}$, where $\hat{x}_i(t) = \{\hat{v}_i(t), \hat{s}_i(t)\}$, and the own reference state trajectory $\bar{X}_i(t)$ where similar notation is used. Both trajectories $\hat{X}_{i-1}(t)$ and $\bar{X}_i(t)$ are computed directly from $V^{\text{s},*}(s_1(t))$ through function manipulation and integration relatively initialized at $s_{i-1}(t)$ and $s_{i-1}(t - \tau_i)$. This definition guarantees that the two trajectories are spaced by a pure time gap τ_i along the whole length.

In the case of the leading vehicle the computation of the own reference state trajectory $\bar{X}_1(t)$ is computed from $V^{\text{s},*}(s_1(t))$ by function manipulation and integration directly initialized at $s_1(t)$.

B. Vehicle model

The model used in the vehicle trajectory tracking layer is an affine model obtained by simplifying and discretizing the model in (1). An affine model, as it will discuss in the next section, allows the MPC problem to be solved in reasonable short time. The equations defining the model are

$$
\begin{aligned}
\frac{v_i(t_{k+1}) - v_i(t_k)}{\Delta t_{\text{MPC}}} &= f_{\text{e},i}(t_k) + f_{\text{b},i}(t_k) + f_{\text{ext},i}(t_k), \\
\frac{s_i(t_{k+1}) - s_i(t_k)}{\Delta t_{\text{MPC}}} &= v_i(t_k),
\end{aligned} \tag{9}
$$

where $f_{\text{e},i}(t)$ and $f_{\text{b},i}(t)$ denote respectively the engine and braking scaled forces and are defined as $f_{\text{e},i} = F_{\text{engine},i}/m_i$ and $f_{\text{b},i} = F_{\text{brake},i}/m_i$. $f_{\text{ext},i}(t_k)$ is a known disturbance and represents an approximation of all the external scaled forces acting on the vehicle (i.e., $f_{\text{ext},i} \approx (-F_{\text{gravity},i} - F_{\text{roll},i} - F_{\text{drag},i})/m_i$); its value can be precomputed using the vehicle

reference trajectories as follows:

$$f_{\text{ext},i}(t_k) = g\sin(\alpha(\bar{s}_i(t_k))) - c_r g\cos(\alpha(\bar{s}_i(t_k)))$$
$$- \frac{1}{2m_i}\rho A_v C_D(\bar{d}_i(t_k))\bar{v}_i^2(t_k),$$

where $\bar{d}_i(t_k) = \hat{s}_{i-1}(t_k) - \bar{s}_i(t_k) - l_{i-1}$ for $i \geq 2$ and $\bar{d}_i(t_k) = \infty$ for $i = 1$. In the MPC formulation, we will refer to the model in (9) as $x_i(t_{k+1}) = A_i x_i(t_k) + B_i u_i(t_k) + C_i(t_k)$, where $x_i = [v_i, u_i]^{\text{T}}$ and $u_i = [f_{\text{e},i}, f_{\text{b},i}]^{\text{T}}$ denote respectively the state and input vectors.

C. Model constraints

The model has been bounded by time-varying constraints on input and state that have been linearized by using the reference state trajectories.

1) Input constraints: the scaled engine and braking force has been bounded by:

$$0 \leq f_{\text{e},i}(t_k) \leq \frac{P_{\max,i}}{m_i \bar{v}_i(t_k)},$$
$$m_i \eta_i g\mu\cos(\alpha(\bar{s}_i(t_k))) \leq f_{\text{b},i}(t_k) \leq 0.$$

In the MPC formulation, we will refer to these constraints as $u_i(t_k) \in \mathcal{U}_i(t_k)$.

2) Speed constraint: in order to guarantee that the vehicle respects the speed limit, the speed has been bounded by the following constraint:

$$v_{\min}(\bar{s}_i(t_k)) \leq v_i(t_k) \leq v_{\max}(\bar{s}_i(t_k)).$$

In the MPC formulation, we will refer to this constraint as $v_i(t_k) \in \mathcal{V}_i(t_k)$.

3) Safety constraint: in case an HDV driver takes control of the vehicle, speed and position trajectories can deviate significantly from the reference ones. Therefore it is necessary to introduce a constraint that guarantees that the platoon state lies always in a safety region. This is done locally in each MPC formulation by guaranteeing that in the worst case of a preceding vehicle full-braking, each follower vehicle is able to stop behind the preceding one, i.e.:

$$s_i(t_{h+h_{\text{d}}}) - \frac{1}{2}\frac{v_i^2(t_{h+h_{\text{d}}})}{-\underline{a}_{\text{b}}} \leq s_{i-1}(t_h) - \frac{1}{2}\frac{v_i^2(t_h)}{-\bar{a}_{\text{b}}} - l_{i-1}, \quad (10)$$

where $h_{\text{d}}\Delta t_{\text{MPC}}$ is a reaction delay due to computation time and sensor and actuator latencies; $\underline{a}_{\text{b}}$ and $\bar{a}_{\text{b}}$ denote respectively the lower and upper bounds on the maximum achievable braking acceleration (in absolute value), taking in account the uncertainty on slope, air drag coefficient C_{D} and braking system efficiency η_i. Inequality (10) is a quadratic convex constraint and in the MPC formualtion we will refer to it as $q_{\text{safe}}(x_i(t_{h+h_{\text{d}}})) \leq 0$.

D. Cost function

The objective of the vehicle trajectory tracking layer is to follow the optimal trajectory provided by the platoon coordinator layer, while minimizing the braking action. The

cost function has been therefore defined as follows:

$$J_i^{\text{MPC}}(X_i(t_h), U_i(t_h)) =$$
$$= \sum_{k=h}^{h+H_{\text{MPC}}-1}(x_i(t_k) - \bar{x}_i(t_k))^{\text{T}}\begin{bmatrix} q_v & 0 \\ 0 & q_s \end{bmatrix}(x_i(t_k) - \bar{x}_i(t_k))$$
$$+ \sum_{k=h}^{h+H_{\text{MPC}}-1}u_i(t_k)^{\text{T}}\begin{bmatrix} q_e & 0 \\ 0 & q_b \end{bmatrix}u_i(t_k),$$

where $X_i(t_h) = \{x_i(t_h), ..., x_i(t_{h+H_{\text{MPC}}-1})\}$ and $U_i(t_h) = \{u_i(t_h), ..., u_i(t_{h+H_{\text{MPC}}-1})\}$ respectively denote the set of the states and inputs over the horizon. While the weights q_v, q_s and q_e have been chosen in order to have a reasonable trade-off between tracking performance and engine force fluctuation, the weight q_b has been chosen large enough such that only the activation of the safety constraint $q_{\text{safe}}(x_i(t_{h+h_{\text{d}}})) \leq 0$ could require a significant braking force.

E. Model predictive control formulation

We now have all the elements to formulate the MPC problem:

$$\min_{U_i(t_h)} J_i^{\text{MPC}}(X_i(t_k), U_i(t_k)) \tag{11a}$$
$$\text{subj. to} \quad x_i(t_{k+1}) = A_i x_i(t_k) + B_i u_i(t_k) + C_i(t_k) \tag{11b}$$
$$u_i(t_k) \in \mathcal{U}_i(t_k) \tag{11c}$$
$$v_i(t_k) \in \mathcal{V}_i(t_k), \tag{11d}$$
$$q_{\text{safe}}(x_i(t_{h+h_{\text{d}}})) \leq 0, \text{ if } i \geq 2 \tag{11e}$$
$$x_i(t_h) = [v_i(t), s_i(t)]^{\text{T}}, \tag{11f}$$

where $k = h, ..., h+H_{\text{MPC}}-1$ and (11f) represents the initial condition of the MPC problem.

Since the model is affine and the constraints are linear or quadratic convex, the MPC formulation can be recasted as a convex quadratic constraint quadratic programming (QCQP) problem. In the simulations showed in the following section, the safety constraint (11e) has been approximated by a set of linear inequalities in order to solve the MPC problem with efficient quadratic programming (QP) solvers. We will refer to the outputs of the vehicle trajectory tracking layer as $f_{\text{e},i}^*(t)$ and $f_{\text{b},i}^*(t)$, which represent the optimal scaled engine and braking forces, respectively.

VI. SIMULATION RESULTS

In this section we illustrate the functioning of the whole control architecture, by presenting the simulation results for three relevant scenarios. For all scenarios, the simulated platoon is composed by two HDVs having both a mass m_i of 36 tons, a length l_i of 10 m and a maximum power $P_{\max,i}$ of 200 kW. The parameters used in the two control layers are displayed in table I.

TABLE I

CONTROLLERS' PARAMETERS

Δs_{DP}	S_{DP}	f_{DP}	Δt_{MPC}	T_{MPC}	f_{MPC}
6 m	2004 m	0.25 Hz	50 ms	2 s	20 Hz

A. Hill scenario

In this scenario, as illustrated in Figure 2, a platoon of two vehicles overcomes a hill. The hill is composed of a uphill stretch of 250 m with 3% slope, a flat stretch and a downhill stretch of 250 m with -3% slope.

Since the vehicles are not able to keep a constant speed during the uphill, as shown in the second plot of Figure 2, the coordinator layer requires the platoon to increase the speed before the uphill, to full-fuel while decelerating during the uphill and increase again the speed after the uphill. Note that the uphill acceleration computed by the platoon coordinator is defined by the maximum power of the first HDV (see the inputs plot in Figure 2) and that the second HDV needs less power due to the reduced air drag.

In the downhill stretch we can notice a symmetric behavior. In order to avoid braking (i.e., converting kinematic energy in heat) and have minimum fueling, the coordinator layer requires to the platoon to reduce the speed before the downhill, minimize the fueling, while accelerating during the downhill and reduce the speed again after the downhill. Note that the speed trajectory is such that the second vehicle is coasting, while the first one needs to fuel a little bit in order to compensate the higher aerodynamic resistance (see the inputs plot in Figure 2).

We can finally notice how the vehicle trajectory tracking layers follow the trajectories without requiring any braking. This is due to the fact that the platoon coordinator simultaneously takes both look-ahead and platooning into account and represents one of the main contribution of this work.

B. Manual driving scenario

In this scenario, as illustrated in Figure 3, a platoon of two vehicles is driving in traffic conditions. In order to handle the traffic situation, the first HDV's driver needs to take control of his vehicle, while the second HDV remains under the control of the trajectory tracking layer. More in detail the first HDV's driver coasts for 7.2 s, full-accelerates for 1.8 s and keeps constant speed for 2 s before returning the control of the vehicle to the controller.

We can notice how, as shown in the second and third plots of Figure 4, the vehicle trajectory tracking layer of the second vehicle does not require it to brake, even if the distance error grows; this can be explained by the fact that, because of the relative high weight q_b on $f_{b,i}$ in the cost function (11), the vehicle trajectory tracking layer would require a noticeable braking action only if the safety constraint are is activated. Finally we can notice how the platoon trajectory generating layer reschedules new trajectories starting from the real-time position and speed of the first HDV. In this scenario, as illustrated in Figure 4, a platoon of two vehicles is driving in traffic conditions. Because of a car invading his lane, the first HDV's driver suddenly takes control of his vehicle by hard-braking and, after that, returns the vehicle control to the controller. More in detail the driver brakes with a deceleration of -4 ms^{-2} for 1.4 s.

Unlike the previous scenario, as shown in the third plot of Figure 4, the change of speed of the first vehicle is large enough to activate the safety constraint of the second vehicle. Therefore in order not to violate the safety constraint,

the trajectory tracking layer of the second HDV naturally requires the vehicle to brake. In addition, since the speeds of the HDVs is decreasing during the braking we can notice how the distance between the vehicles is allowed also to decrease (due to the time gap strategy) and, therefore, the second vehicle needs to brake less intensely than the first one.

VII. CONCLUSIONS

A two-layer control architecture for look-ahead platooning is introduced in this paper, as well as a control strategy within this framework. In the first layer, a dynamic programming approach is presented to exploit available preview information on the road topography to obtain a fuel-optimal trajectory for a platoon of heavy-duty vehicles. This information is used in the second layer, in which decentralized model predictive controllers perform the real-time vehicle control, hereby guaranteeing safe operations. Simulations are performed to confirm the suitability of this approach.

Future work will focus on a more detailed analysis of the control architecture, both from a practical and theoretical points of view. The platoon behavior will be studied for different vehicles weight, engine powers and platoon lengths, analyzing also string stability related aspects [16]. In addition the performance of the presented controller will be compared with other approaches using more realistic scenarios and platoon models.

REFERENCES

[1] A. Alam. *Fuel-efficient distributed control for heavy duty vehicle platooning*. Licentiate thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2011.

[2] A. Alam, J. Mårtensson, and K.H. Johansson. Look-ahead cruise control for heavy duty vehicle platooning. In *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems, The Hague, The Netherlands*, pages 928–935, 2013.

[3] R. Bellman. *Dynamic programming*. Princeton University Press, New Jersey, USA, 1957.

[4] C. Bonnet and H. Fritz. Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing. In *Proceedings of the Future Transportation Technology Conference, Costa Mesa, USA*, SAE Technical Paper 2000-01-3056, 2010.

[5] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. In preparation, available online at http://www.mpc.berkeley.edu/mpc-course-material, 2004.

[6] L. Bühler. Fuel-efficient platooning of heavy duty vehicles through road topography preview information. Master's thesis, KTH - Royal Institute of Technology, 2013.

[7] W.B. Dunbar and R.M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[8] European Commission. *EU transport in figures – Statistical pocketbook*. Publications Office of the European Union, Luxembourg, 2013.

[9] E Hellström, J. Åslund, and L. Nielsen. Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice*, 18(11):1318–1327, 2010.

[10] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17(2):245–254, 2009.
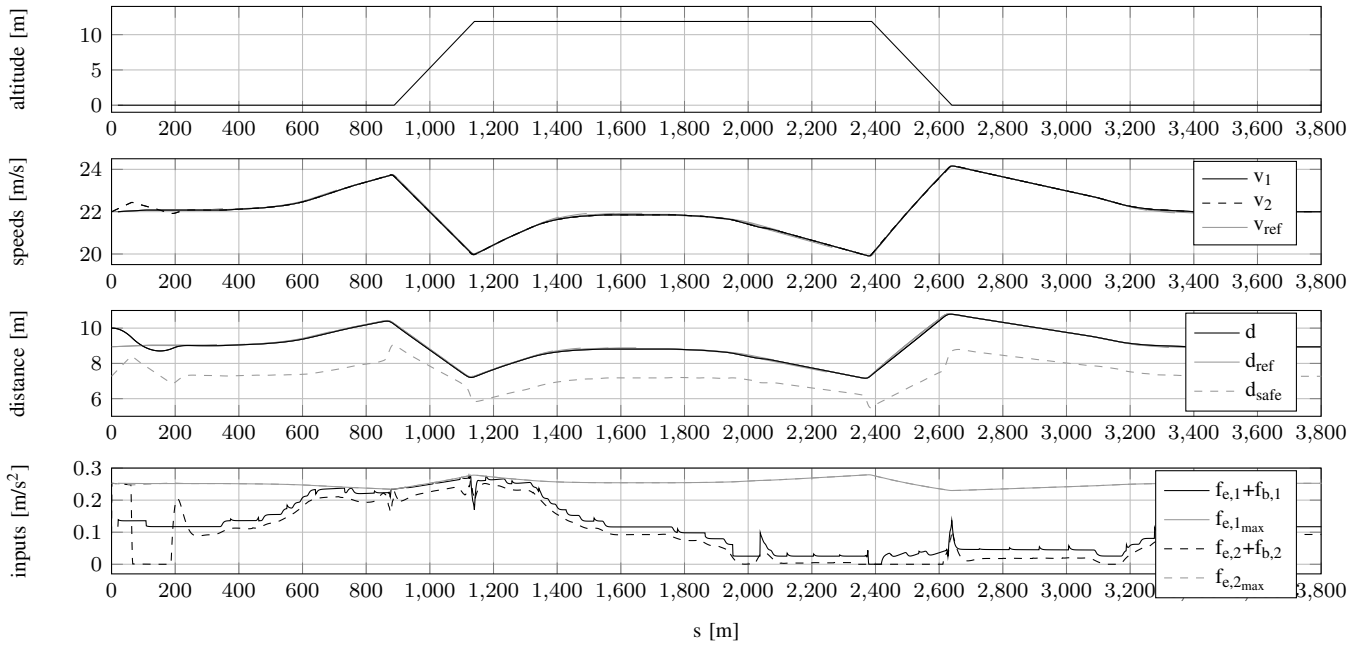
Fig. 2. In this figure we present the result of the simulation for a platoon of two vehicles overcoming a hill. The hill is composed by a uphill stretch of 250 m with 3% slope, a flat stretch and a downhill stretch of 250 m with −3% slope. In the first plot the road altitude is shown; in the second plot the speeds of the two vehicles and the reference speed trajectories are shown; in the third plot, the distance, the reference distance trajectories and the safe distance are shown; finally, in the fourth plot, the inputs and the inputs' limits are shown; note that the scaled braking and traction forces are plotted together: when their sum is negative the HDV is braking, while when it is positive, the HDV is fueling. All variables have been plotted against position.
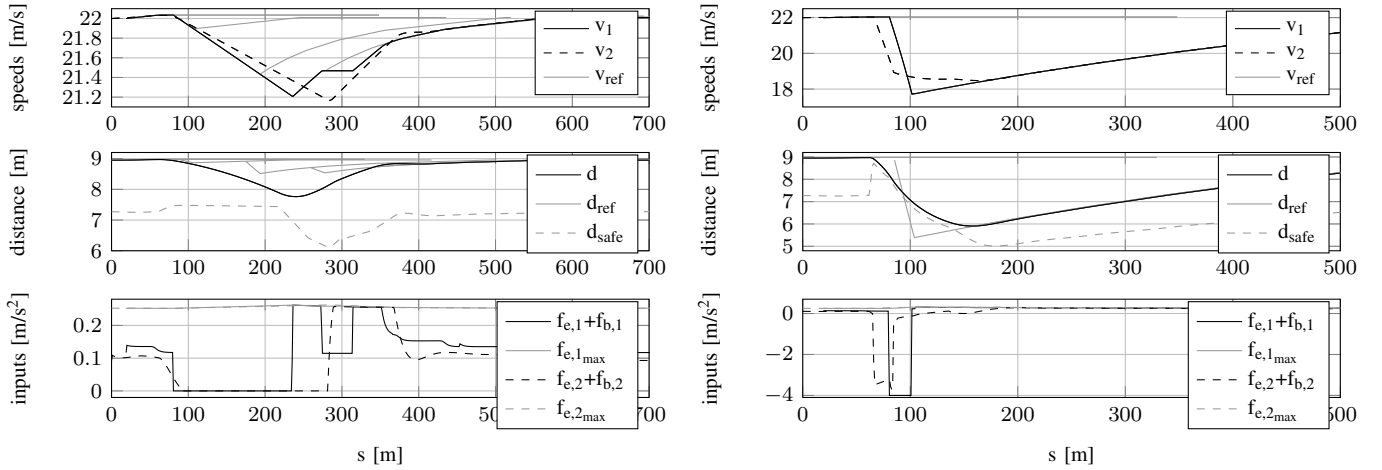

Fig. 3. In this figure we present the result of the simulation for a platoon of two vehicles driving in traffic conditions. In order to handle the traffic situation, the first HDV's driver takes control of his vehicle. More in detail the driver coasts for 7.2 s, full-accelerates for 1.8 s and keeps constant speed for 2 s before returning the vehicle control to the controller. The plots' description corresponds to that one given in the caption of Figure 2.

Fig. 4. In this figure we present the result of the simulation for a platoon of two vehicles driving in traffic conditions. Because of an unexpected event, the first HDV's driver suddenly takes control of his vehicle by hard-braking with a deceleration of $-4 \frac{m}{s^2}$ for 1.4 s. The plots' description corresponds to that one given in the caption of Figure 2.

[11] W.-H. Hucho, editor. *Aerodynamics of road vehicles*. Society of Automotive Engineers, USA, 4th edition, 1998.

[12] W. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, AC-11(3):355–361, 1966.

[13] G.J.L. Naus, R.P.A. Vugts, J. Ploeg, M.J.G. Van de Molengraft, and M. Steinbuch. String-stable CACC design and experimental validation: a frequency-domain approach. *IEEE Transactions on Vehicular Technology*, 59(9):4268–4279, 2010.

[14] L.E. Peppard. String stability of relative-motion PID vehicle control systems. *IEEE Transactions on Automatic Control*, 19(5):579–581, 1974.

[15] A.B. Schwarzkopf and R.B. Leipnik. Control of highway vehicles for minimum fuel consumption over varying terrain. *Transportation Research*, 11(4):279–286, 1977.

[16] D. Swaroop and J.K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[17] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.