Fuel-efficient Model Predictive Control for Heavy Duty Vehicle Platooning using Neural Networks

Gustav Ling, Klas Lindsten, Oskar Ljungqvist, Johan Löfberg, Christoffer Norén and Christian A. Larsson

Fuel-efficient Model Predictive Control for Heavy Duty Vehicle Platooning using Neural Networks

Gustav Ling[1], Klas Lindsten[1], Oskar Ljungqvist[2], Johan Löfberg[2], Christoffer Norén[1], Christian A. Larsson[1]

Abstract— **The demand for fuel-efficient transport solutions are steadily increasing with the goal of reducing environmental impact and increasing efficiency. Heavy-Duty Vehicle (HDV) platooning is a promising concept where multiple HDVs drive together in a convoy with small intervehicular spacing. By doing this, the aerodynamic drag is reduced which in turn lowers fuel consumption. We propose a novel Model Predictive Control (MPC) framework for longitudinal control of the follower vehicle in a platoon consisting of two HDVs when no vehicle-to-vehicle communication is available. In the framework, the preceding vehicle's velocity profile is predicted using artificial neural networks which uses a topographic map of the road as input and is trained offline using synthetic data. The gear shifting and mass of consumed fuel for the controlled follower vehicle is modeled and used within the MPC controller. The efficiency of the proposed framework is verified in simulation examples and is benchmarked with a currently available control solution.**

I. Introduction

In 2015, Heavy Duty Vehicles (HDVs) accounted for 30% of the total vehicular CO_2 emissions in the EU while constituting only 5% of the vehicle fleet [1]. Scania CV AB stated in 2010 that one third of the costs related to an operational truck is fuel consumption [2]. These facts together with improving safety for HDVs and their surrounding traffic have motivated the development of Advanced Driver Assistance Systems (ADAS). Two such systems that are available in HDVs are Adaptive Cruise Control (ACC) and Look-Ahead Cruise Control (LACC). ACC uses information from onboard sensors to maintain a constant speed as well as a constant distance to a vehicle ahead. The LACC is a similar system that in addition uses information about the upcoming road grade to adjust the speed of the HDV in an energy efficient manner [3].

An even more sophisticated ADAS with applications to HDVs, and a hot research topic, is *platooning*. The concept of platooning is that two or more HDVs drive together in a convoy at short intervehicular distances. By doing this, a reduction in aerodynamic drag is achieved which offers the possibility of reduced fuel consumption. Independent studies have shown that a reduction of fuel consumption up to 10% for the follower vehicles is possible [4], [5]. Furthermore, platooning can be considered as a large step toward fully autonomous HDVs. Up to this day there is no established communication standard that enables platooning with vehicle-to-vehicle communication (V2V). Therefore, concepts with two vehicle platoons have been developed when no V2V is considered. One such concept is the ACC-LACC platoon which consists of a leading HDV controlled with an LACC and a follower HDV controlled using ACC.

We propose a novel Model Predictive Control (MPC) framework for longitudinal control of the follower vehicle in a platoon consisting of two HDVs and no V2V is available. In this method, the preceding HDV velocity profile is predicted using techniques from Machine learning, in this case Artificial Neural Networks (ANNs). Additionally, gear shifting and the mass of consumed fuel for the controlled follower truck are modeled and used within the MPC controller. The mass of consumed fuel is added to the objective function as a final cost in the MPC formulation to achieve fuel efficient control of the follower truck.

A. Related work

The nonlinear longitudinal dynamics for HDVs in platoons including powertrain modeling and external forces such as gravitation, aerodynamic drag and rolling resistance are well understood and models can for example be found in [6], [7], [8]. The inclusion of gear shifting in the longitudinal HDV models significantly add to their complexity. It is nonetheless desirable to include this in the model in order to be able to describe properties such as powertrain inertia and gear ratios which both vary between different gears, and the absence of torque output to the powertrain during a shift. In [9], the gear shift process is modeled by setting some period of time with zero torque output during a shift for an HDV. In the same work, the extended HDV model is used within a dynamic programming framework to design an energy efficient LACC for a single HDV. A mixed integer approach for cruise control of a single HDV has been considered in [10], however the period of time with torque loss during a gear shift is neglected.

When no V2V exists, a significant part of the control problem is to predict the future behavior of the preceding vehicle. Methods for online estimation of the preceding HDV mass and maximum torque have been developed in [11].

Concerning the development of different control structures for HDV platooning, a distributed control structure for vehicle platooning using receding horizon control is presented in [12]. In [6] a hierarchical control architecture is proposed that splits the control problem into two layers. The top layer is responsible for computing fuel-optimal speed profiles for the entire platoon, while the lower layer is responsible for low level control. Both methods above rely on V2V and can thus not be utilized in this work. An MPC controller for longitudinal control of the follower truck for non-cooperative platooning on flat road segments is presented in [13]. In the proposed framework, the future velocity profile of the preceding vehicle is assumed to be known or well

[1] Scania CV, Södertälje, Sweden.
(e-mail: {klas.lindsten, christian.y.larsson, gustav.ling, christoffer.noren}@scania.com)
[2] Division of Automatic Control, Linköping University, Sweden,
(e-mail: oskar.ljungqvist@liu.se, johan.lofberg@liu.se)

approximated as constant within the prediction horizon. The results are promising and compared to a standard ACC controller a significant energy reduction is achieved that mainly originates from less unnecessary breaking. However, a framework for prediction of the future velocity profile of the preceding vehicle is omitted.

The novelties of the proposed methods is based on results from [14] which consist of the development and integration of an MPC controller, augmented system description with gear shift dynamics as well as systems for the prediction of the preceding vehicle speed profile. Also, an objective function which allows for the balancing of fuel consumption and a look-ahead strategy to remain within a platooning distance of the preceding vehicle is proposed.

The outline of the remainder of the paper is as follows. In Section II the nonlinear longitudinal HDV and platooning models are derived. Also, mass of consumed fuel and the gear shift dynamics of the controlled follower HDV are modeled. In Section III the framework used for prediction of the preceding HDV velocity profile is presented. In Section IV an MPC controller is proposed for longitudinal control of the follower HDV. Finally, Section V and Section VI present the simulation results and conclusions, respectively.

II. MODELING

We consider a platoon consisting of a controlled HDV and a preceding HDV as depicted in Figure 1. Denote the control signals $u = (T_e, F_b)$ where T_e is the net engine torque and F_b is the brake force. For a constant gear the controlled HDV acceleration is given by

$$\frac{dv}{ds} = f(s, v, G, d, u) \tag{1}$$

where s is the position, v is the velocity, G is the gear number and d is the distance to the preceding vehicle. To model the platooning dynamics the time headway gap $\tau_{hw} = d/v$ between the two vehicles is modeled as

$$\frac{d\tau_{\text{hw}}}{ds} = h(s, v, v_p, G, d, u) \tag{2}$$

where v_p is the velocity of the preceding HDV which is considered as an exogenous input. Moreover, the fuel consumption is modeled as

$$\frac{dm_f}{ds} = \xi(v, G, u), \tag{3}$$

where m_f denotes the mass of consumed fuel. Now, explicit models are presented.

A. Longitudinal model

As in [7] the controlled follower truck is modeled as a point mass translating in one dimension. Define the total inertial mass $m_t(G)$ as

$$m_t(G) = m + \frac{i_G^2 i_f^2 \eta_G \eta_f J_e + J_w}{r_w^2} \tag{4}$$

where all model parameters are explained in Table II and the subscript G denotes that the parameter varies with the gear number. Using Newton's second law, the longitudinal dynamics in spatial coordinates can be expressed as [7]:

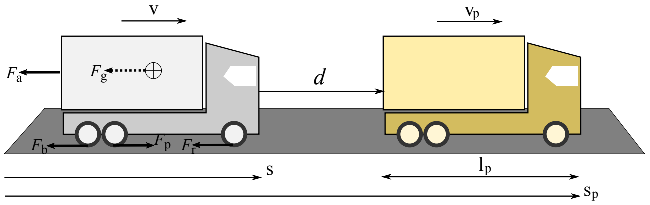

Fig. 1: Conceptual visualization of a two vehicle platoon.

TABLE I: Longitudinal forces acting on the controlled HDV.

Force	Explanation	Expression
$F_a(d,v)$	Air resistance	$\frac{1}{2}c_D A_a \rho_a \phi_d(d) v^2$
$F_r(s)$	Rolling resistance	$mgc_r \cos\alpha(s)$
$F_g(s)$	Gravitational force	$mg \sin\alpha(s)$

$$\begin{aligned} m_t(G) v \frac{dv}{ds} &= m_t(G) v f(s, v, G, v\tau_{\text{hw}}, u) \\ &= \frac{i_G i_f \eta_G \eta_f}{r_w} T_e - F_b - F_a(v\tau_{\text{hw}}, v) - F_r(s) - F_g(s) \end{aligned} \tag{5}$$

The road slope at position s is given by $\alpha(s)$ and the resisting forces are presented in Table I. In modeling of the air resistance F_a, the reduction from platooning needs to be taken into account. This is done by multiplying an intervehicular distance dependent reduction factor $\phi(d)$ to the standard air resistance model. The reduction factor is achieved by fitting a nonlinear model to measurement data from wind tunnel experiments as presented in [15]. The powertrain is modeled from the engine to the wheels via the clutch, gearbox and final drive. The propeller and drive shafts are assumed stiff and are therefore left out. The engine and the wheels are modeled as rotational masses and other components are modeled using static relationships. For a complete derivation of the powertrain model we refer to [7]. All model parameters are explained in Table II.

B. Modeling of platoon dynamics

To model the dynamic in the distance between the two HDVs the time headway gap τ_{hw} is used, see Figure 1. This velocity independent spacing policy is preferable to use compared to constant distance spacing according to [7]. Let l_p denote the length of the preceding HDV, the spatial derivative of τ_{hw} yields

$$\begin{aligned} \frac{d\tau_{\text{hw}}}{ds} &= h(s, v, v_p, G, v\tau_{\text{hw}}, u) \\ &= \frac{1}{v}\frac{d}{dt}\left(\frac{s_p - l_p - s}{v}\right) = \frac{(v_p - v) - \tau_{\text{hw}} v \frac{dv}{ds}}{v^2} \end{aligned} \tag{6}$$

where the relationship $\tau_{\text{hw}} v = s_p - l_p - s$ has been used in the final equality. Finally, using (5) in (6) the explicit expression for the model of the time headway gap is attained.

C. Modeling of fuel consumption

In order to produce a fuel efficient control strategy and make up for an intuitive objective function in the MPC controller, the mass of consumed fuel m_f is modeled. This is done by fitting a plane to the engine specific fuel map

TABLE II: Model parameters of the controlled HDV.

Parameter	Notation
Wheel radius	r_w
Final drive ratio	i_f
Gear ratio	i_G
Gear efficiency	η_G
Final drive efficiency	η_f
Wheel inertia	J_w
Engine inertia	J_e
Vehicle frontal area	A_a
Air density	ρ_a
Rolling resistance coefficient	c_r
Aerodynamic drag coefficient	c_D
Vehicle mass	m

for the controlled HDV. The fuel map is a representation of fuel flow $\dot{m}_f$ for different engine torque T_e and engine speed ω_e combinations. By utilizing the fact that engine speed is a gear dependent scaling of the vehicle speed v as [7]:

$$\omega_e = \frac{i_G i_f}{r_w} v \tag{7}$$

the mass of consumed fuel m_f can in spatial coordinates be modeled as

$$\frac{dm_f}{ds} = \xi(v, G, u) = C_{1,G} \frac{i_G i_f}{r_w} + \frac{C_{2,G}}{v} T_e + \frac{C_{3,G}}{v} \tag{8}$$

where $C_{1,G}$, $C_{2,G}$ and $C_{3,G}$ are gear dependent plane constants. In Figure 2 the planes approximating the fuel map for the three highest gears of the HDV are visualized. As can be seen, the highest gear is most fuel efficient for most of the considered operating points. However, the constraints on the net engine torque T_e with respect to velocity varies for different gears which will result in gear shifting strategies during road segments with large inclination.

D. Resulting state space model

Denote the states $x = (v, \tau_{\text{hw}}, m_f)$, the control signals $u = (T_e, F_b)$ and the exogenous inputs $w = (\alpha, v_p)$. The resulting state space model is obtained using (5), (6) and (8) as

$$\frac{dx}{ds} = f(x, u, w, G) \tag{9}$$

Moreover, a stationary working point for each gear G is used $\rho_0(G) = (x_0, u_0(G), w_0)$ in order to produce a linear discrete approximation of (9) for a constant gear number G. With the sampling distance $\Delta s = T_s v_0$, where T_s denotes the sampling time, the discrete piecewise linear approximation of (9) is

$$\bar{x}_{k+1} = F(\rho_0(G))\bar{x}_k + H(\rho_0(G))\bar{u}_k + W(\rho_0(G))\bar{w}_k \tag{10}$$

where the Euler forward method was used in the discretization and $\bar{x}_k = x_k - x_0$, $\bar{u}_k = u_k - u_0$ and $\bar{w}_k = w_k - w_0$. The system (10) is a piecewise affine system with respect to the engaged gear which is a binary variable. Now, the gear shift dynamics is considered.

E. Modeling of gear shift dynamics

The gear shift dynamics in this work refer to the model of transition between two gears. During that time, the engine is disconnected from the wheels and thus no propulsive force can be transmitted, which corresponds to zero torque T_e. In the employed model, we will only work with three possible gears (corresponding to gear 12, 13 and 14 on the real truck). To model the gear shift process we define gear engagement states $q \in \mathbb{R}^3$ and gear selection inputs $\mu \in \{0,1\}^3$. The gear shift control signals μ are thus binary and we only allow one non-zero element at every time instant since multiple gears cannot be engaged simultaneously. The gear states in q are continuous and designed to move between 0 and 1. The gear shift is modeled as a linear first order system with time constant τ_g. With the approximation $v_k \approx v_0$ and the sampling distance $\Delta s = T_s v_0$, the linear discrete model of the gear shift dynamics is ($q_{i,k}$ denotes the value of the ith element at sample k)

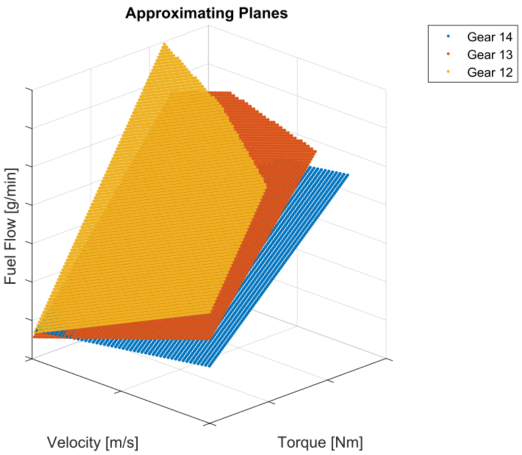

Fig. 2: The planes approximating the fuel map for gear 12 (yellow), 13 (red) and 14 (blue), respectively.

$$q_{i,k+1} = q_{i,k} + \frac{T_s}{\tau_g}(-q_{i,k} + \mu_{i,k}) \tag{11}$$

Note that the dynamics in (11) are independent of the other states. The idea is now to threshold the gear state vector components by some value g_{th} to describe which gear is engaged, i.e., gear i is engaged if $q_i \geq g_{th}$). The threshold g_{th} and the time constant τ_g are adjusted to mimic the real system behavior with respect to the time spent, or rather distance spent, during a gear shift.

III. Velocity profile prediction

In this section, models for the prediction of the preceding vehicle velocity profile are presented. A black-box modeling approach is taken, meaning direct physical relationships are not considered. Instead, generic models with appropriate structure, inputs (here primarily road slope data) and outputs are assumed and then fitted to data [16]. All data considered in this section are positionally discretized at a step length of 10 m. Training, validation and test data are obtained by simulating the HDVs (using a highly accurate nonlinear simulation model developed by Scania CV AB) to be modeled over sampled real world road profiles.

Several models of different HDVs with different power-to-weight ratios and control strategies are modeled and gathered in a model bank. The output of these different models are then corrected online using logged measurements of the preceding vehicle velocity. The model achieving the highest fit after this correction is then selected for use. By tailoring the models' outputs to measurements online in this way, a relatively small amount of models in the model bank is sufficient to cover a broad range of HDVs under different operating conditions. The final output of this system, a prediction of the preceding vehicle's future velocity profile up to the MPC horizon, is then forwarded to the MPC controller as the predicted exogenous input v_p.

A. Artificial neural networks

Artificial neural networks are made up of three types of layers, an input layer, hidden layers, and an output layer. The input layer consists of a vector containing the data to be propagated through the network. The hidden layers consist of one or more neurons. A neuron is essentially a function which takes as input a weighted sum of the outputs of the neurons in the previous layer, add to it a bias, and propagate it through an activation function, which is often chosen to be a monotonically increasing and differentiable function with a range of $(-1,1)$ or $(0,1)$. The output layer works the same way as the hidden layers, and as the final layer it also produces the final output of the network [17], [18].

For an illustration of a this in a closed loop nonlinear autoregressive exogenous input (NARX) neural network (the type used in this work), see Figure 3.

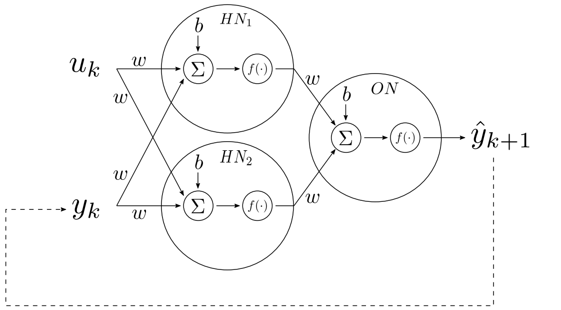

Fig. 3: Illustration of a closed loop (the output in step k is fed back as an input in step $k+1$), sometimes called *recurrent*, NARX neural network with inputs u_k and y_k and as output the prediction $\hat{y}_{k+1}$ [19]. Note that the weights w and biases b are all unique. $f(\cdot)$ denotes the activation function. HN and ON are abbreviations for hidden neuron and output neuron, respectively. The closed loop property of such networks are important in time series prediction problems, since it enables predictions of the output an arbitrary number of steps into the future [20]. Note that u and y are not related to the control signals and outputs in other parts of this work, but merely constitutes generic variables in this example.

These networks are fitted to data by means of tuning the parameters, i.e. the weights and biases in the network, commonly using backpropagation algorithms [18].

B. Velocity prediction model

Let H_p^α and H_f^α denote the past and future horizons of road slope values in discrete steps to be used as model inputs, respectively. Further, let H_p^v denote the horizon in discrete steps for which past outputs (velocities) are used to predict the upcoming one. For an illustration of this, see Figure 4. First, we split the velocity predictions into two parts according to

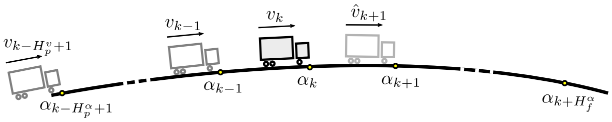

Fig. 4: Illustration of the one step ahead velocity prediction problem and its inputs and output. The inputs are the past and future road slope values and past velocities, and the output is an estimate of the velocity in the next step, $\hat{v}_{k+1}$.

$$\hat{v}_{k+1} = \hat{v}_{k+1}^{\text{cc}} + \hat{v}_{k+1}^{\delta}, \tag{12}$$

where $\hat{v}_{k+1}^{\text{cc}}$ denotes a bias representing an assumed or estimated cruise controller set speed of the preceding vehicle at step $k+1$, and $\hat{v}_{k+1}^{\delta}$ represents the estimated deviation from the cruise controller set speed of the preceding vehicle at step $k+1$. The cruise controller set speed is assumed to remain constant for a horizon, thus

$$\hat{v}_{k+1}^{\text{cc}} = f_{\text{cc}}(\hat{v}_k^{\text{cc}}) = \hat{v}_k^{\text{cc}} \tag{13}$$

and can be initiated with a guess from the road speed limit, or be estimated with methods covered later on. The deviations from set speed are modeled by

$$\hat{v}_{k+1}^{\delta} = f_{\delta}(\alpha_{k-H_p^\alpha+1}, \ldots, \alpha_k, \ldots, \alpha_{k+H_f^\alpha}, \hat{v}_{k-H_p^v+1}^{\delta}, \ldots, \hat{v}_k^{\delta}) \tag{14}$$

where the function $f_\delta(\cdot)$ is a closed loop NARX neural network, which can be used for prediction an arbitrary number of steps into the future, see Figure 3. The function thus constitutes a model for the velocity response in terms of how it increases or decreases given some input data on the road topography and past estimates of velocity deviations.

C. Implementation

Several architectures of varying numbers of layers and neurons have been evaluated. In the case of ACC prediction, adding more than one hidden layer seems to have an almost negligible effect on performance. The final structure used for all of the ACC networks consists of one hidden layer with

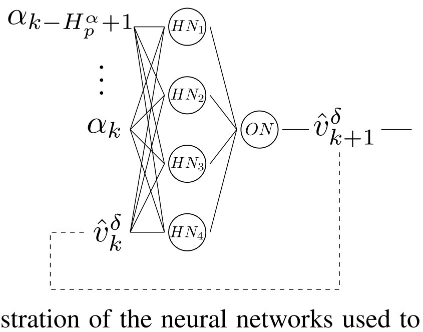

Fig. 5: Illustration of the neural networks used to model the set speed deviations for a vehicle operating an ACC. Note that the network is in closed loop form with respect to the output, $\hat{v}_{k+1}^{\delta}$, meaning that it can be used for prediction of an arbitrary number of steps, since the road slope α is known at every step.

four neurons, with 1 through 30 past road slope values and 1 past set speed deviation used as input, see Figure 5.

The neural networks modeling the ACC set speed deviations are thus functions on the form (14), with $H_p^\alpha = 30$, $H_f^\alpha = 0$ and $H_p^v = 1$.

Finding a structure suitable for LACC prediction is somewhat more challenging due to the lower correlation between road slope and velocity [21]. Empirical results indicate that using more than two hidden layers will not increase performance. The final structure used for all of the LACC networks consists of one hidden layer with ten neurons, with 1 through 30 past road slope values, every fifth future road slope value up to a horizon of 300, and 1 past set speed deviation used as input, see Figure 6. The future road slope inputs are thinned to reduce the model complexity and reduce training times.

The neural networks modeling the LACC set speed deviations are thus functions on the following form

$$\begin{aligned}\hat{v}^{\delta}_{k+1} = f_{\delta}(&\alpha_{k-H_p^\alpha+1},\dots,\alpha_k,\alpha_{k+1},\alpha_{k+6}\dots, \\ &\alpha_{k+H_f^\alpha-5},\alpha_{k+H_f^\alpha},\hat{v}^{\delta}_{k-H_p^v+1},\dots,\hat{v}^{\delta}_{k})\end{aligned} \tag{15}$$

where $H_p^\alpha = 30$, $H_f^\alpha = 300$ and $H_p^v = 1$. Both the ACC and the LACC models were implemented using the Neural Network Toolbox for MATLAB, both developed by MathWorks [22]. The training, validation and test data were obtained by means of simulating the trucks in question, operating their respective cruise controllers at a set speed of 80 km/h, over sampled real world road profiles. The training data amounted to roughly 2000 km of simulated operation for each of the models. The models were trained using the implementation of Bayesian regularization backpropagation found in the Neural Network Toolbox. For details on Bayesian regularization backpropagation, see e.g. [23].

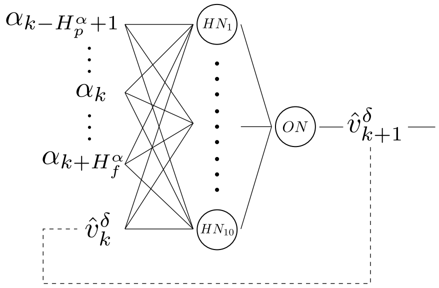

Fig. 6: Illustration of the neural networks used to model the set speed deviations for a vehicle operating an LACC. Note that the network is in closed loop form with respect to the output, $\hat{v}^\delta_{k+1}$, meaning that it can be used for prediction of an arbitrary number of steps, since the road slope α is known at every step.

D. Online correction of predictions

Since all models contained in the model bank cover only a small number of HDVs, and are also all trained for the prediction of HDVs operating their respective cruise controllers at a set speed of 80 km/h, some online adjustments of the model outputs are desirable to account for discrepancies encountered during usage. This is done by letting the models make predictions over a correction window. These predictions are saved, and when driving this window, measurements of the target HDV are taken and logged. The model outputs and measurement data are then used to tune the models to better fit the target HDV in question.

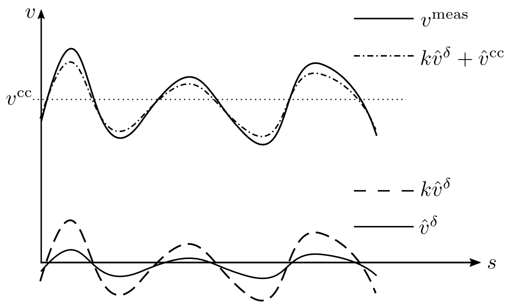

Fig. 7: Illustration of the working principle of the online correction system.

Let H_c denote the horizon of the window on which the correction is based, and n_p the number of predictor models in the model bank. The following predictions of the preceding vehicle velocity response over the horizon are initially made

$$\hat{\boldsymbol{v}}^\delta_{\boldsymbol{1}} = \left[\hat{v}_1^{\delta,1} \dots \hat{v}_{H_c}^{\delta,1}\right]^T \tag{16}$$

$$\vdots$$

$$\hat{\boldsymbol{v}}^\delta_{n_p} = \left[\hat{v}_1^{\delta,n_p} \dots \hat{v}_{H_c}^{\delta,n_p}\right]^T \tag{17}$$

When the horizon is reached, the following measurements of the preceding vehicle have also been collected

$$\boldsymbol{v}^{\text{meas}} = \left[v_1^{\text{meas}} \dots v_{H_c}^{\text{meas}}\right]^T \tag{18}$$

By transforming the predictions according to

$$\boldsymbol{z_1} = \left[z_1^1 \dots z_{H_c}^1\right]^T = \boldsymbol{A_1 x_1} = \begin{bmatrix}\boldsymbol{1} & \hat{\boldsymbol{v}}^\delta_{\boldsymbol{1}}\end{bmatrix}\begin{bmatrix}v_1^{\text{cc}} \\ k_1\end{bmatrix} \tag{19}$$

$$\vdots$$

$$\boldsymbol{z}_{n_p} = \left[z_1^{n_p} \dots z_{H_c}^{n_p}\right]^T = \boldsymbol{A}_{n_p}\boldsymbol{x}_{n_p} = \begin{bmatrix}\boldsymbol{1} & \hat{\boldsymbol{v}}^\delta_{n_p}\end{bmatrix}\begin{bmatrix}v_{n_p}^{\text{cc}} \\ k_{n_p}\end{bmatrix} \tag{20}$$

we want to find an estimate of the preceding vehicle set speed, v^{cc}, and a gain k such that the fit of the transformed predictions $\boldsymbol{z}$ is maximized with respect to the measurements $\boldsymbol{v}^{\text{meas}}$, for each of these transformed predictions, see Figure 7. The problem of maximizing this fit with respect to the set speed v^{cc} and correction gain k can be reduced to the following least squares problem

$$\underset{\boldsymbol{x_i}}{\text{minimize}} \;\; (\boldsymbol{A_i x_i} - \boldsymbol{v}^{\text{meas}})^T(\boldsymbol{A_i x_i} - \boldsymbol{v}^{\text{meas}}) \tag{21}$$

with the analytical solution

$$\boldsymbol{x_i} = (\boldsymbol{A_i^T A_i})^{-1}\boldsymbol{A_i v}^{\text{meas}} \tag{22}$$

The parameters $\boldsymbol{x_i} = [v_i^{\text{cc}} \; k_i]^T$ are then to be used to correct the output of model i for the duration of the next window.

E. Online classification

The preceding vehicle is classified by selecting the predictor in the model bank achieving highest fit (post correction) w.r.t. measurements in the past correction window according to

$$\text{Model to use} = \arg\max_i \; \text{Fit}(\boldsymbol{z_i}, \boldsymbol{v}^{\text{meas}}) \tag{23}$$

IV. MPC CONTROLLER

The task of the controller is to keep the vehicle at a desired velocity and distance to the preceding truck using the limited engine torque, braking and gear selection as inputs, consuming as little fuel as possible, taking known past and future α and predicted future v_p into account. With the three gear shift states appended to the velocity, time headway and mass of consumed fuel states we have a prediction model with six states in the MPC controller. As an initial performance measure, we use an ℓ_1 penalty on absolute tracking error in velocity and headway over a future prediction horizon of N samples, appended with a penalty for consumed fuel.

$$J = \sum_{i=1}^{N-1} |Q_{11}\bar{v}_{k+i}| + |Q_{22}\bar{\tau}_{k+i}| + m_{f,k+N-1} \tag{24}$$

As an alternative, one could replace the absolute values with quadratics. The prediction model is compactly written as $\bar{x}_{k+1} = f_G(\bar{x}_k, \bar{u}_k, \bar{w}_k)$ where G denotes the engaged gear. The dependence w.r.t desired velocity v_0 and headway τ_0 which are used for linearization is omitted for brevity. Due to the discrete dependence on the gear, and the binary gear shift control signal μ, our model is hybrid. The goal now is to reformulate this model to a mixed-integer linear programming representable form. At each sample along the future prediction horizon, we introduce a binary variable $\delta_{j,k+i}$ to indicate if gear j is engaged and a binary variable $\delta_{0,k+i}$ to encode neutral (i.e., between gears with no torque on wheels). Using logical implications, we can enforce the hybrid logic of the model with

$$\delta_{j,k+i} \rightarrow \bar{x}_{k+i+1} = f_j(\bar{x}_{k+i}, \bar{u}_{k+i}, \bar{w}_{k+i}, q_{k+i}), q_{j,k+i} \geq g_{th} \tag{25}$$

$$\delta_{0,k+i} \rightarrow \bar{x}_{k+i+1} = f_0(\bar{x}_{k+i}, \bar{u}_{k+i}, \bar{w}_{k+i}, q_{k+i}), q_{j,k+i} < g_{th} \forall j \tag{26}$$

By adding $\delta_{0,k+i} + \sum_j \delta_{j,k+i} = 1$, the logical model ensures that the correct prediction model is applied when the corresponding gear is engaged, and that only one gear can be engaged. In addition to the gear engagement logics, we also have the gear selection logic which says that μ sums to 1 at all time instances. The logical implications are easily converted to linear constraints using big-M modeling strategies, but the details are omitted, as the employed modeling tool YALMIP [24] will do this step automatically given the logical model.

The resulting mixed-integer linear program (or quadratic program if there are quadratics in the objective) will become increasingly complex as the prediction horizon increases, and to reduce the combinatorial explosion we limit the complexity of the gear shifting sequence in a move blocking fashion. Gear shifts are only allowed every Mth sample, effectively meaning that the control signals μ are kept constant over M samples. Albeit an artificial limitation, it is a reasonable assumption that gear changes are rare. Note though that move blocking does not mean that gear changes are limited in closed-loop, it only means that the future plan the MPC controller derives has to use a reduced number of gear changes. We do however add an additional term to the objective penalizing gear changes from the current gear to avoid excessive gear changing in closed-loop.

Since the model only is an approximation and there are external disturbances, hard constraints on velocity and time headway are relaxed using slack variables which are added with linear penalty to the objective function.

V. RESULTS

This section presents results achieved from simulation experiments where the MPC controller together with the velocity prediction system are used to control a highly accurate nonlinear vehicle simulation model developed by Scania CV AB. In all experiments, the motion of the preceding HDV is simulated using topological data from real world scenarios which have not been used during the training-phase of the velocity prediction system. Results from the set speed deviation predictors and the correction system are first presented. Then, results for the complete MPC framework is presented, i.e., the MPC controller together with the preceding vehicle velocity profile prediction system. In the simulation examples, the prediction horizon $N = 100$, the sampling time $T_s = 1$s and the weights $Q_{11} = 1$ and $Q_{22} = 10$ are used in MPC controller. For the gear shifting sequence, the move-blocking design parameter was set to $M = 50$. The MPC controller is implemented in MATLAB using YALMIP [24] and the simulations have been performed on a standard laptop computer with Intel Core i5 processor at 3.30GHz CPU and 8GB of RAM. The average computation time for each iteration of the control framework was about 250 milliseconds. However, it should be noted that the computational time is increasing exponentially fast if the move blocking parameter is significantly lowered due to the combinatorial explosion in number of possible gear sequences [14].

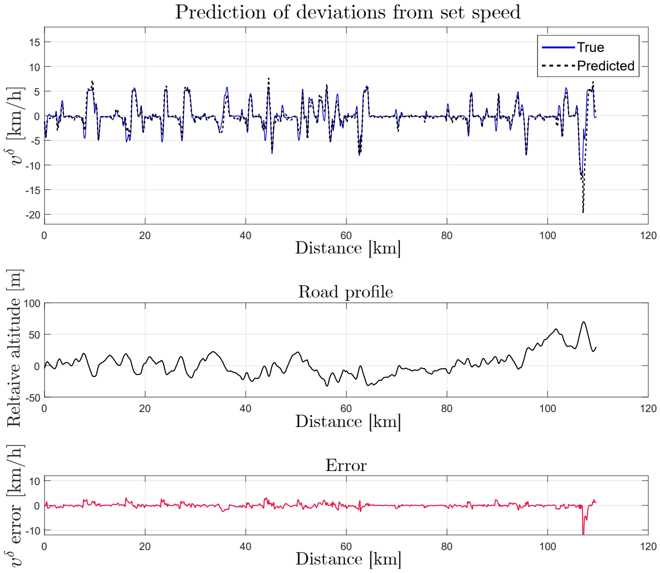

Fig. 8: Evaluation of the 40 tonnes LACC set speed deviation predictor on the road section between Södertälje and Norrköping. Note that the entire series of predictions is made in closed loop. The model generalize well and captures the dynamics well. In particular, the model captures the LACC behavior of slowing down before downhills accurately, as well as the subsequent accelerations in the downhills.

A. Predictor and corrector results

The performance of the set speed deviation predictors are evaluated by predicting set speed deviations on a road section that has not been used during the training of the prediction models. The predictions are compared with true set speed deviation data from the HDVs they model. The 120 kilometer typical long haulage highway between the Swedish cities Södertälje and Norrköping is chosen for model evaluation. Prediction errors of the set speed deviation for the ACC predictors are presented in Table III and for the LACC predictors in Table IV. These models generalize well and the RMSE are as low as 0.05 km/h for ACC and 0.54 km/h for LACC. Set speed deviations for the LACC predictor is visualized in Figure 8.

Results from the corrector system is presented in Figure 9. The top plot displays the past 500 positional steps over which measurements of the preceding truck have been logged. The line marked as raw prediction shows the prediction made at step 0 with no correction gain (but with a correct set speed guess), and the one marked corrected prediction shows the prediction adjusted with the computed correction gain, made at step 500. The computed correction gain and set speed estimate from the correction window are used to adjust new predictions in the upcoming window, shown in the bottom plot. The bottom plot also displays the raw prediction and the true profile for the purpose of comparison. The correction system successfully tunes the model dynamics to better fit an operating condition for which it was not originally trained, in this case the dynamics around the set speed of 90 km/h instead of 80 km/h.

TABLE III: Performance measures of the ACC predictors predicting deviations from the cruise controller set speed between the Swedish cities Södertälje and Norrköping.

Predictor	**Avg. error** [km/h]	**RMSE** [km/h]
10 t ACC	5.2e-04	0.0523
20 t ACC	0.0025	0.1176
30 t ACC	-0.0176	0.2000
40 t ACC	-0.0999	0.3935
50 t ACC	0.0407	0.5596
60 t ACC	-0.1428	0.6152

TABLE IV: Performance measures of the LACC predictors predicting deviations from the cruise controller set speed between the Swedish cities Södertälje and Norrköping.

Predictor	**Avg. error** [km/h]	**RMSE** [km/h]
20 t LACC	-0.0255	0.5367
40 t LACC	-0.0927	1.0275
50 t LACC	0.0618	1.1657

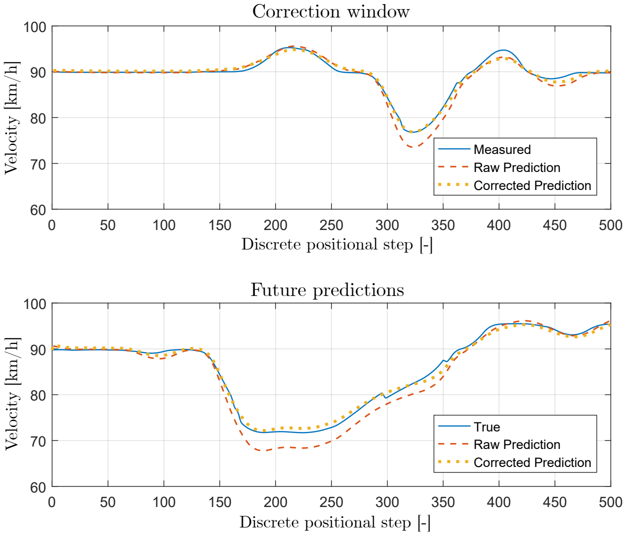

Fig. 9: The top plot shows a comparison between the measured velocity, the raw prediction and the corrected prediction. The bottom plot shows the future outcome with the true profile, the raw prediction and the corrected prediction. Note that despite the discrete positional steps are marked as 0 to 500 in both windows, on a common timeline, the correction window would be 0 to 500 and the future window 501 to 1001.

B. MPC framework results

The complete system consisting of the MPC controller along with online prediction, correction and classification is evaluated on the same road section as the velocity predicting system. The performance of the presented MPC framework is evaluated by measuring the fuel consumption and brake energy consumption compared to an ACC follower truck in an ACC-LACC platoon. Both the MPC controlled follower truck and the ACC controlled follower truck are simulated with a mass of 40 tonnes and solutions for different leading HDVs are compared. Different performance measures for scenarios with different leading trucks are presented in Table V. The fuel savings compared to an ACC controlled follower truck in an ACC-LACC platoon are significant, with a relative reduction of up to 5.9%. The total brake energy spent is also significantly lowered and a reduction of up to 76% is achieved in the simulation examples. This may suggest a stronger tendency in the MPC framework to prefer a look ahead control strategy over platooning for some portions of the road. In Figure 10, relevant measures and signals for two identical MPC controlled trucks on the road section between Södertälje and Norrköping are presented. The red dotted line (labeled predictor) shows simulation results achieved with the presented MPC controller together with online prediction, correction and classification. The blue line (labeled oracle) shows simulation results for a benchmark system where the presented MPC controller is used together with the true velocity profile of the preceding truck as input. As can be seen, the resulting state and control signal trajectories during the two closed-loop simulations are mostly overlapping each other. The initial mismatch, as

TABLE V: Data for follower trucks with a mass of 40 t using the MPC controller with predictor systems. The last column presents reduction in fuel consumption compared to a 40 t follower truck in an ACC-LACC platoon, and the fourth column presents brake energy reduction compared to the same truck.

Lead	**Vel.** [km/h]	τ_{hw} [s]	$\Delta E_{b,ACC40}$	$\Delta m_{f,ACC40}$
20 t ACC	80.05 ± 4.09	2.93 ± 2.26	-70.7%	-5.9%
40 t ACC	79.96 ± 4.31	2.88 ± 2.26	-71.0%	-3.2%
50 t ACC	79.44 ± 4.95	2.75 ± 2.14	-72.7%	-2.8%
20 t LACC	79.94 ± 4.23	2.92 ± 2.25	-76.0%	-5.7%
40 t LACC	79.79 ± 4.67	2.79 ± 2.16	-67.3%	-2.6%
50 t LACC	79.43 ± 5.17	2.74 ± 2.14	-68.6%	-1.9%

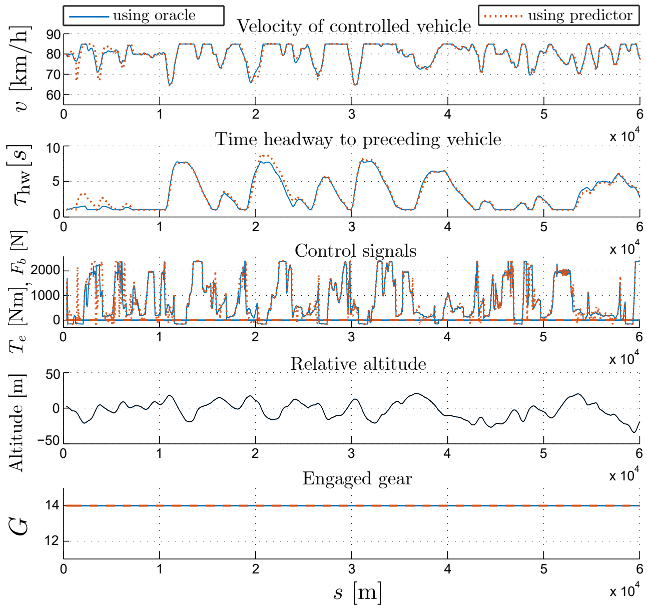

Fig. 10: Visualization of two identical scenarios where one truck is controlled using an oracle and the other using the prediction system. In the control signals subplot, the thin lines represent engine torque and the bold lines represent braking force.

seen in the time headway, is due to the controller using a constant velocity assumption of the preceding vehicle until measurements have been gathered and corrections and a classification have been made. Note that in this road section, no gear changes were made and the reader is referred to the Master's thesis [14] for more simulation examples.

VI. Conclusions and Future Work

A novel MPC framework for longitudinal control of the follower HDV in a platoon consisting of two HDVs where no V2V is available has been presented. In the framework the velocity profile of the preceding HDV is predicted using artificial neural networks. The ANNs have been trained offline using synthetic data sets. Additionally, online classification and online corrector systems are added within the velocity prediction framework. Furthermore, gear shifting and mass of consumed fuel are modeled and utilized in the MPC controller. In order to achieve fuel efficient control of the follower truck, the mass of consumed fuel is added as a final cost in the objective function in the MPC formulation in order to motivate fuel efficient control. The presented MPC framework is tested in simulations where a highly accurate nonlinear vehicle model developed by Scania CV AB is used as simulation model.

The velocity prediction system shows great potential and is able to predict the velocity profile for different HDVs with an average prediction error as low as 0.04 km/h. The simulation results for the complete MPC framework shows that compared to an ACC-LACC platoon, the fuel consumption for the controlled HDV is reduced up to 5.9% and the total consumed brake energy is reduced up to 72.7%.

As future work we would like to investigate if the presented MPC framework can be extended and used for control of longer HDV platoons.

References

[1] ICCT-Europe, "European vehicle market statistics 2015/16," Printed pocketbook, 2015.

[2] "Scania annual report 2010," Scania CV AB, 2010.

[3] E. Hellström, "Look-ahead control of heavy vehicles," Ph.D. dissertation, Linköping University, 2010.

[4] A. Al Alam, A. Gattami, and K. H. Johansson, "An experimental study on the fuel reduction potential of heavy duty vehicle platooning," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 306–311.

[5] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass," *SAE International Journal of Commercial Vehicles*, vol. 7, pp. 626–639, 2014.

[6] V. Turri, "Fuel-efficient and safe heavy-duty vehicle platooning through look-ahead control," Licentiate Thesis, KTH Royal Institute of Technology, 2015.

[7] A. Alam, "Fuel-efficient heavy-duty vehicle platooning," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, 2014.

[8] Q. Deng, "Heavy-duty vehicle platooning modeling and analysis," Licentiate Thesis, KTH Royal Institute of Technology, Stockholm, 2016.

[9] E. Hellström, M. Ivarsson, L. Nielsen, and J. Aslund, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption," *Control Engineering Practice*, vol. 17, pp. 245–254, 2 2009.

[10] C. Kirches, H. G. Bock, J. P. Schlöder, and S. Sager, "Mixed-integer nmpc for predictive cruise control of heavy-duty trucks," in *2013 European Control Conference*, July 2013, pp. 4118–4123.

[11] H. Felixson, "Vehicle ahead property estimation in heavy duty vehicles," Master's thesis, Linköping University, 2014.

[12] W. B. Dunbar and D. S. Caveney, "Distributed receding horizon control of vehicle platoons: Stability and string stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, March 2012.

[13] V. Turri, Y. Kim, J. Guanetti, K. H. Johansson, and F. Borrelli, "A model predictive controller for non-cooperative eco-platooning," in *2017 American Control Conference*, May 2017, pp. 2309–2314.

[14] G. Ling and K. Lindsten, "Model predictive control using neural networks a study on platooning without intervehicular communications," Master's thesis, Linköping University, 2017.

[15] L. Eriksson and L. Nielsen, *Modeling and control of engines and drivelines*. Hoboken: Wiley, 2014.

[16] L. Ljung and T. Glad, *Modeling and Identification of Dynamic Systems*. Lund: Studentlitteratur, 2016.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[18] S. Russell, *Artificial Intelligence : a Modern Approach*. Pearson Education Limited, 2016.

[19] H. Siegelmann, B. Horne, and C. Giles, "Computational capabilities of recurrent narx neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, pp. 208–215, April 1997.

[20] Y. U. Yim and S.-Y. Oh, "Modeling of vehicle dynamics from real vehicle measurements using a neural network with two-stage hybrid learning for accurate long-term prediction," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1076–1084, July 2004.

[21] P. Kupsc, "Preceding vehicle dynamics modeling for fuel efficient control strategies," Master's thesis, KTH Royal Institute of Technology, 2016.

[22] M. Hudson Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox^{TM} User's Guide*. The MathWorks, Inc., 2017.

[23] D. J. C. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, pp. 415–447, 08 1992.

[24] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.