Title: Coordination of motion actuators in heavy vehicles using Model Predictive Control Allocation

Publisher: IEEE

DOI: 10.1109/IVS.2016.7535447

Copyright: © 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Version: Accepted article

To cite this article: SINIGAGLIA, Andrea, et al. Coordination of motion actuators in heavy vehicles using Model Predictive Control Allocation. In: 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2016. p. 590-596.

To link to this article: http://dx.doi.org/10.1109/IVS.2016.7535447

Coordination of Motion Actuators in Heavy Vehicles using Model Predictive Control Allocation

Andrea Sinigaglia[1], Kristoffer Tagesson[1,3], Paolo Falcone[2], Bengt Jacobson[1]

Abstract— **The paper presents a Model Predictive Control Allocation (MPCA) method in order to coordinate the motion actuators of a heavy vehicle. The presented method merges the strong points of two different control theories: Model Predictive Control (MPC) and Control Allocation (CA); MPC explicitly considers the motion actuators dynamics before deciding on a suitable input for the actuators while CA dynamically decides how to use the motion actuators in order to modify the vehicle behaviour. The designed MPCA formulation belongs to the class of Quadratic Programming (QP) problems so that the solution is optimization based, i.e. at every step a quadratic cost function has to be minimized while fulfilling a set of linear constraints. Three scenarios were set up to evaluate the effectiveness of the controller: split-μ braking, split-μ acceleration and brake blending. Split-μ means that the wheels on one side of the vehicle are in contact with a slippery surface (e.g. ice) while the wheels of the other side lay on a normal surface (e.g. dry asphalt). The split-μ scenarios aim to combine three different types of motion actuators, disc brakes, powertrain and rear active steering (RAS), in order to brake/accelerate the vehicle while keeping it on course. The third scenario is a mild braking event on a normal road and its purpose is to combine the use of the engine brake with the disc brakes. Simulation results of the scenarios have shown promising vehicle performance when using MPCA to coordinate the motion actuators. Tests on a real vehicle have then confirmed the expected vehicle behaviour in a slit-μ braking scenario. MPCA has also been compared to a simpler CA formulation, in all scenarios. The performance of the two is comparable in steady state, but MPCA shows advantages in transients, whereas CA is less computationally demanding.**

I. INTRODUCTION

Motion actuators are defined as devices that are able to produce specific forces and moments on the vehicle and consequently affect its motion. The motion actuators considered in this paper are disc brakes, powertrain and rear active steering (RAS), which all use the wheels to generate forces and moments on the vehicle. Because of the high number of wheels, heavy vehicles are in general over-actuated: meaning that there are several ways to combine the motion actuators in order to reach the desired behaviour for the vehicle. Being over-actuated is a suitable characteristic for a vehicle as it increases fault tolerances and gives more freedom on how to use the motion actuators during a manoeuvre.

Control allocation (CA) methods have been extensively exploited to conveniently combine the use of the motion actuators, see e.g. [1–4]. A CA method for an over-actuated vehicle uses a modular architecture for the control where a high-level motion controller computes the resulting forces and moments on the vehicle requested to cope with a specific manoeuvre. Once these forces and moments are known, they are sent to a low-level coordination controller, where CA is implemented, to find a suitable use of the motion actuators so that they produce the requested resulting forces and moments on the vehicle. CA methods usually do not consider the dynamics of the motion actuators, possibly only rate limitations. On the other hand, as we are trying to coordinate motion actuators with different behaviours, it is useful to introduce in the controller an explicit formulation of the actuators dynamics by using a Model Predictive Control (MPC) approach. The resulting control structure, referred to as Model Predictive Control Allocation (MPCA) in the following, has recently been used in other fields of research. Within aerospace [5–7] use an MPCA approach for the inner loop of a re-entry vehicle guidance and control system, while in [8] as part of a missile flight control system. In the automotive area, [9, 10] propose the MPCA strategy to control an engine thermal management system, while [11] use it for the hybrid braking of an electric vehicle with four in-wheel motors. The performance of a vehicle when facing the three proposed scenarios, split-μ braking, split-μ acceleration and brake blending, has never been studied using MPCA. Moreover, no documentation has been found about the implementation of MPCA to improve the dynamics of a heavy vehicle. The objective of this paper is to investigate if MPCA can cope with the three above mentioned scenarios and compare the performance achieved by MPCA with CA. The article is structured as follows. Section II starts with a brief introduction of CA and MPC methods, followed by a description of the designed MPCA together with the solver used during performed simulations and tests. Section III describes in detail the three selected scenarios while section IV shows the results from both simulations and tests. The conclusions, benefits and limitations of the proposed MPCA are stated in section V. Variables and signs are in accordance with ISO 8855:2011 [12] and units are expressed according to SI, unless otherwise specified.

II. MODEL PREDICTIVE CONTROL ALLOCATION

Some convenient vehicle states (e.g. yaw velocity, vehicle sideslip angle, longitudinal and lateral speed) are considered when studying the dynamics of a vehicle. The evolution of these states is modified by the resulting forces and moments that, at each instant, are acting on the vehicle and are here

---

[1] Division of Vehicle Engineering and Autonomous Systems, Chalmers University of Technology, Göteborg, Sweden.

[2] Division of Mechatronics, Chalmers University of Technology, Göteborg, Sweden.

[3] Department of Chassis Strategies and Vehicle Analysis, Volvo Group Truck Technology, Göteborg, Sweden.

Corresponding author: `kristoffer.tagesson@volvo.com`

collocated in a vector, v, called the virtual force input. A common choice, when considering planar vehicle motion, is $v = [F_X \; F_Y \; M_Z]^T$ where F_X and F_Y are respectively the resulting longitudinal and lateral forces on the vehicle, while M_Z is the yaw moment of the vehicle. Fig. 1 shows a typical control architecture for over-actuated systems composed by: (1) a high-level motion controller that finds a suitable v to modify the dynamics of the vehicle in accordance with what the driver desires. Once v has been computed, it is sent to (2) a low-level coordination controller that is where CA or MPCA can be implemented in order to allocate the motion actuators that produce v. Knowing v is the starting point for the methods described below and examples of how to compute the virtual force input are presented in [1, 2].

A. Control Allocation (CA)

Defining $\delta \in \mathbb{R}^m$ as the vector that describes the outputs of the motion actuators (e.g. brake pressure, RAS angle, engine torque, ...), CA finds δ so that $f(\delta) = v \in \mathbb{R}^p$, where m and p are respectively the number of considered motion actuators and resulting forces and moments on the vehicle, f is the function that maps δ into v. If a limited range of operations is considered, the relation between δ and v can be well approximated by a linear model. CA objective is then to find:

$$\{\delta \mid B\delta = v \; ; \; \underline{\delta} \leq \delta \leq \overline{\delta}\} \tag{1}$$

where the *effectiveness matrix* $B \in \mathbb{R}^{p \times m}$ maps the actuators outputs δ_i into the resulting forces and moments on the vehicle v. $\underline{\delta}\,, \overline{\delta}$ are lower and upper limits for the actuators that take into account both, saturation and rate limitations. Saturation limitations are related to the physical capabilities of the motion actuators while the rate limits are set to overcome the problem of the motion actuators dynamics. As the dynamics usually are neglected in CA methods, no distinction is made between the commanded input of an actuator δ_{cmd} and the actual output of the actuator δ. In this paper rate limits have been set so that actuator outputs should be able to follow what is commanded by the controller, at every sample. Hence, $\delta_{cmd} \approx \delta$, which permits actuator dynamics to be neglected in CA.

Because of the constraints a solution for (1) is not guaranteed to exist. Moreover, if a solution exists it might not be unique since the system is overactuated, i.e. $dim(v) < dim(\delta)$. In order to deal with these two problems, (1) is usually rewritten as an optimization problem of the form:

$$\begin{aligned} &\delta = arg\,min\;(\|B\delta - v\|^2_{W_v} + \gamma\,\|\delta - \delta_d\|^2_{W_\delta}) \\ &subject\;to: \;\underline{\delta} \leq \delta \leq \overline{\delta} \end{aligned} \tag{2}$$

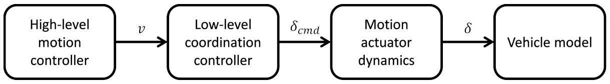

Fig. 1. Control architecture for an over-actuated vehicle. CA assumes $\delta_{cmd} = \delta$, while MPCA considers the dynamics of the actuators. Both controllers are implemented in the low-level coordination controller block.

In (2) the first term is used to force $B\delta$ as close as possible to v. In case $\exists \delta \mid B\delta = v$, the second term, characterised by a small value of the scalar γ, assures a unique solution of the problem. Because of γ, the second term of (2) starts to influence the solution only when $B\delta \approx v$. Here, $\| \cdot \|^2_W$ is a quadratic form $\|a\|^2_W = a^T W a$ and the matrices W_v and W_δ are diagonal weighting matrices. W_v is used to prioritize certain forces or moments in v, while W_δ prioritizes the usage of certain actuators and δ_d is used as a desired set point for the actuator vector.

B. Model Predictive Control (MPC)

Model predictive control aims to find the optimal control input for a system by predicting the possible future states of the system. The method is optimization-based and the predictions are made by using a dynamic model of the system in the constraints. In the common time discrete and linear implementation of MPC, constraints are written on the form:

$$\delta(k+1) = A_\delta \delta(k) + B_\delta \delta_{cmd}(k) \tag{3}$$

Where A_δ and B_δ are the matrices that describe the dynamics of the system, $\delta(k)$ and $\delta_{cmd}(k)$ are respectively the motion actuators outputs and commanded inputs at time instant k. In this paper the constraints illustrated in (3) have been used to account for the dynamics of the different motion actuators. Disc brakes and RAS have been modeled from a real vehicle, sending a step input to each actuator and observing the corresponding response. The powertrain has been modeled as a first order system slower than the disc brakes but faster than RAS.

C. MPCA tailored for intended vehicle

The vehicle model used in simulations is a solo truck 6x2 tag axle with RAS. The vehicle has three axles where the front axle is steerable and mechanically connected to the steering wheel, the second axle is the driven axle while the third steerable axle is controlled by wire. The wheels are numbered so that 1 is the front left wheel, 2 is the front right wheel and so on. The proposed MPCA formulation for the low-level coordination controller is:

$$\begin{aligned} \delta_{cmd}(0) = arg\,min\;(&\sum_{k=1}^{N} \|B\delta(k) - v\|^2_{W_v} + \\ &\gamma \sum_{k=1}^{N} \|C\delta(k) + \delta_e\|^2_{W_\delta}) \end{aligned} \tag{4}$$

$$subject\;to: \delta(k+1) = A_\delta \delta(k) + B_\delta \delta_{cmd}(k) \tag{5}$$

$$\underline{\delta}_{cmd}(k) \leq \delta_{cmd}(k) \leq \overline{\delta}_{cmd}(k) \tag{6}$$

$$\underline{\delta}(k) \leq \delta(k) \leq \overline{\delta}(k) \tag{7}$$

where $k = 1, ..., N$ is the time horizon considered for the predictions of the motion actuators, $\delta_{cmd} \in \mathbb{R}^8$, when $i = 1, ..., 6$, $\delta_{cmd,i}$ is the commanded pressure to the brake in wheel i, $\delta_{cmd,7}$ is the commanded torque at the driven axle and $\delta_{cmd,8}$ is the commanded angle to RAS. δ_i are the corresponding output values of the motion actuators. $\underline{\delta}_{cmd}, \overline{\delta}_{cmd}$

are the lower and upper limits for the commanded inputs of the motion actuators while $\underline{\delta}, \overline{\delta}$ are the lower and upper limits for the actual outputs of the motion actuators. Considering, in the analysed scenarios, vehicle planar dynamics on straight paths, the two forces and moments of interest on the vehicle are the resulting longitudinal force F_X and the yaw moment M_Z, so that $v = [F_X \; M_Z]^T$. The first term of the objective function (4) aims to combine δ_i so that $B\delta(k)$ approaches v as soon as possible over a time horizon of N steps. To define the effectiveness matrix $B \in \mathbb{R}^{2\times 8}$ the following relations between the variables δ_i and their respective forces generated at the interface wheel/ground have been considered, together with the resulting yaw moment produced on the vehicle. For $i = 1, ..., 6$:

$$F_{XT,i} = \frac{k_b}{r_j}\delta_i; \quad M_{Z,i}(COG) = \pm\frac{k_b}{r_j}\frac{w_j}{2}\delta_i \tag{8}$$

where $F_{XT,i}$ is the longitudinal force generated at wheel i, k_b is the relation between brake pressure and moment generated on the wheel, r_j is the wheel dynamic radius at the axle j, and w_j is the track width of the axle j. For the powertrain:

$$F_{XT,7} = \frac{1}{r_2}\delta_7; \quad M_{Z,7}(COG) = 0 \tag{9}$$

where $F_{XT,7}$ is the longitudinal force produced by all the driven wheels. Because the vehicle is equipped with an open differential, the torque at the driven axle is always split $50-50$ and will hence never generate yaw moment on the vehicle.

$$F_{XT,8} = 0; \quad M_{Z,8}(COG) = -(C_{\alpha,5} + C_{\alpha,6})l_r\delta_8 \tag{10}$$

where $F_{XT,8}$ is the longitudinal force produced by turning the wheels of the tag axle, $C_{\alpha,5}$ and $C_{\alpha,6}$ are respectively the cornering stiffness of the wheel 5 and 6, l_r is the distance between the centre of gravity of the vehicle and the third axle. For all the equations, $M_{Z,i}(COG)$ is the yaw moment produced at the centre of gravity of the vehicle by δ_i. The second term of (4) starts to influence the result of the minimization only when $B\delta(k) \approx v \; \forall k$ and it has been defined as follows:

$$C = diag(\frac{k_b}{r_1}, \frac{k_b}{r_1}, \frac{k_b}{r_2}, \frac{k_b}{r_2}, \frac{k_b}{r_3}, \frac{k_b}{r_3}, 1, 1) \tag{11}$$

$$\delta_e = [0\; 0\; 0.5\frac{\delta_7(0)}{r_2}\; 0.5\frac{\delta_7(0)}{r_2}\; 0\; 0\; 0\; 0]^T \tag{12}$$

$$\begin{aligned} W_\delta = diag(&\frac{1}{\mu_{x,1}F_{ZT,1}}, \frac{1}{\mu_{x,2}F_{ZT,2}}, \frac{1}{\mu_{x,3}F_{ZT,3}}, \\ &\frac{1}{\mu_{x,4}F_{ZT,4}}, \frac{1}{\mu_{x,5}F_{ZT,5}}, \frac{1}{\mu_{x,6}F_{ZT,6}}, 0, 1) \end{aligned} \tag{13}$$

where $\mu_{x,i}$ and $F_{ZT,i}$ are respectively the maximum longitudinal force coefficient and tyre normal force on wheel i. This, as explained in section III-C, to prioritize the use of the engine brake ($W_\delta(7,7) = 0$) while distributing the braking force on each wheel accordingly to their maximum available force without slipping ($W_\delta(i,i) = \frac{1}{\mu_{x,i}F_{ZT,i}}$, $i = 1, ..., 6$). MPCA has three types of constraints: that of dynamics (5), commanded input (6) and actual output (7) of the motion actuators. The model of the motion actuators is a first order system with different time constants, τ_b for the brakes, τ_p for the powertrain and τ_{RAS} for the RAS. (5) is the discrete version of these models. The constraints in (6) deal with saturation limits and confine the maximum commanded input to the motion actuators. The constraints in (7) have been used to limit the amount of force generated by the motion actuators on the interface wheel/ground and prevent wheel slip. In order to prevent the wheels from slipping, the friction ellipse of every wheel has been taken into account. The wheels of the first and third axle can produce negative longitudinal force and lateral forces, if steered. The constraints shown in Fig. 2 have been considered in order to approximate the friction ellipse of these wheels with linear constraints. Although the low-level controller does not control the steering wheel, the lateral forces produced when using the steering wheel have been taken into account in the constraints of the first axle wheels. As a result, if the driver needs to steer, the pressure in the brakes will ease off to avoid wheel slip and make the vehicle controllable. The motion actuators of the second axle can produce both positive and negative longitudinal force but not lateral forces, the combined force produced by the motion actuators on each one of the second axle wheels has been limited to $[-\mu_{x,i}F_{ZT,i}, \; \mu_{x,i}F_{ZT,i}]$.

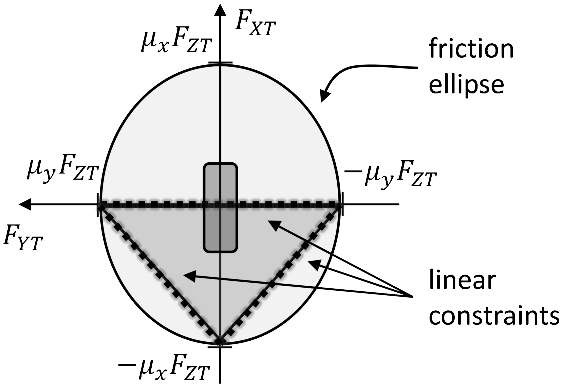

Fig. 2. Linear approximation of the friction ellipse

D. Solver

The solver used to deal with MPCA is CVXGEN [13]. CVXGEN transforms the MPCA problem into a standard QP form and uses an algorithm based on the primal-dual interior-point method with Mehrotra's predictor-corrector to solve the QP problem at each step. One of the objectives of this paper is to compare MPCA with CA so both algorithms have been generated using CVXGEN. To write the CA formulation, the objective function (4) without time horizon has been considered, moreover the constraints in (5) have been supplemented with rate constraints on δ. The other constraints of CA are the same of MPCA with $\delta_{cmd} = \delta$. In order to run the two controllers in real time, during the in-vehicle tests, the two algorithms have been implemented in a rapid prototyping control unit, a MicroAutoBox II. Due to the higher number of variables, the MPCA algorithm was in the order of 10 times slower than the CA algorithm.

III. SCENARIOS

Three scenarios have been considered in this paper to test the performances of MPCA and compare them with CA: split-μ braking, split-μ acceleration and brake blending. A split-μ road is an asymmetric road that is causing a high friction coefficient on one side and low friction coefficient on the other side.

A. Split-μ braking

In this scenario the vehicle brakes on a split-μ road. This is a bit tricky because of the trade-off between an acceptable braking distance and induced vehicle yaw disturbance. The idea in this scenario is illustrated in Fig. 3 where, in the left figure, the brakes on the high friction coefficient side are used to produce a considerable amount of braking force that leads to an undesired yaw disturbance on the vehicle; then, in the right figure, the introduction of RAS counteracts the yaw moment generated by the brakes keeping the vehicle stable without compromising on its stopping distance. The key point of this scenario is to coordinate RAS and brakes in order to build up the two opposite yaw moments at the same time.

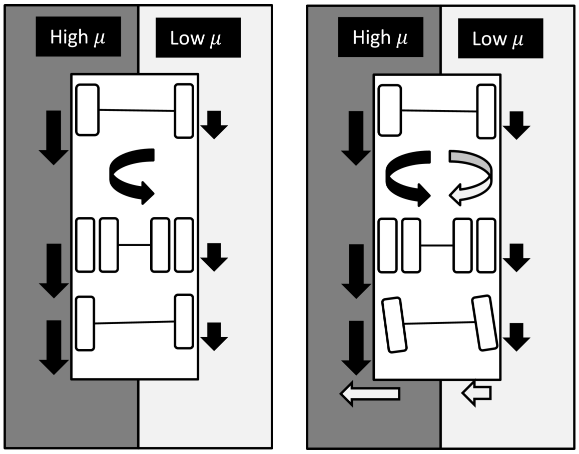

Fig. 3. Strategy to brake on a split-μ road

B. Split-μ acceleration

In this scenario the vehicle accelerate on a split-μ road from standstill. Due to the open differential, the maximum torque at the driven axle is always limited by 2 times what is allowed on the low friction side without causing wheel slip. This limitation often results in insufficient traction forces. Fig. 4 shows how the designed controller overcomes the problem by automatically braking the low friction side wheel so that the wheel on the opposite side can generate a higher traction force to move the vehicle. RAS is introduced again to eliminate the tendency of the vehicle to deviated form a straight path caused by the asymmetric traction forces.

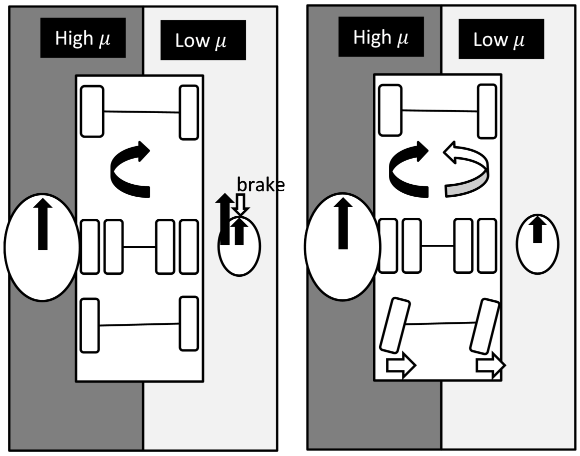

Fig. 4. Strategy to accelerate on a split-μ road

C. Brake blending

The brake blending scenario is here defined as a braking event on a normal road, where the use of the disc brakes is to be combined with the engine brake. The engine brake, in fact, has a slower response than the disc brakes but it is preferable to use the engine brake as much as possible because it does not have the typical problems present on disc brakes: fading and wear. During a mild braking event, when none of the constraints are saturated, there are several ways to achieve $B\delta(k) \approx v \; \forall k$ and so the second term of (4) has an influence on the solution of the allocation problem that is exploited to optimize the use of the motion actuators. In this situation, RAS is not used ($\delta_8(k) = 0 \; \forall k$) and the first six elements of $C\delta(k) + \delta_e$ are the braking forces produced by the actuators on each wheel. The cost of using the engine brake is zero but the force it produces on the third and fourth wheel is taken into account by using δ_e. This is done to prioritize the use of the engine brake but, at the same time, to consider the combined forces produced by the engine brake and disc brakes on the second axle. Due to the costs set by $W_\delta(i,i),\; i = 1, ..., 6$ the disc brakes are used so that each wheel brakes proportionally to the maximum available braking force on the wheel. This approach to distribute the braking force is particularly convenient because it ensures that no wheel starts slipping before all the other wheels reach their peak value $\mu_{x,i} F_{ZT,i}$.

IV. RESULTS

The three scenarios have been implemented in simulations both with MPCA and CA. Due to the relative potential showed during simulations and because of time constraints, only the split-μ braking scenario has been tested in a real vehicle. In all the simulations and tests, avoiding undesired yaw moment on the vehicle has always been prioritized by means of the W_v matrix. The opposite would be to prioritise stopping distance only, which has a trivial solution. A deeper analysis of what states to ultimately prioritise can be found in [4]. As the potential of MPCA is to predict the behaviour of the actuators and their influence on the vehicle,

TABLE I
PARAMETERS DURING SIMULATIONS

Parameters	Value	description
m	$22760\,kg$	vehicle mass
$F_{Z,1}$	$62519\,N$	static vertical load axle 1
$F_{Z,2}$	$107174\,N$	static vertical load axle 2
$F_{Z,3}$	$53582\,N$	static vertical load axle 3
$[r_1\ r_2\ r_3]$	$[0.53\ 0.53\ 0.54]\,m$	dynamic wheel radius on axle 1-2-3
$[w_1\ w_2\ w_3]$	$[2.05\ 1.85\ 2.05]\,m$	track width of axle 1-2-3
$[l_{12}\ l_{13}]$	$[4.8\ 6.17]\,m$	distance axle 1-2 and distance axle 1-3
$[\tau_b\ \tau_p\ \tau_{RAS}]$	$[0.1\ 0.3\ 0.4]\,s$	motion actuators time constants
$[K_P\ K_I\ K_D]$	$[-1.3\ -0.1\ 0]\,s$	PID parameters
k_b	$-1470.6\,\frac{Nm}{bar}$	relation between wheel torque and brake pressure
$W_v(1,1)$	$0.1\,\frac{1}{N^2}$	weighting matrix
$W_v(2,2)$	$100\,\frac{1}{(Nm)^2}$	weighting matrix
γ	0.01	scalar for the prioritization
N	10	steps in the minimization
T	$0.05\,s$	sampling time used in predictive actuator model
T_s	$0.01\,s$	controller sample time

TABLE II
PARAMETERS DURING TESTS

Parameters	Value	description
m	$17640\,kg$	vehicle mass
$F_{Z,1}$	$57290\,N$	static vertical load axle 1
$F_{Z,2}$	$49050\,N$	static vertical load axle 2
$F_{Z,3}$	$49050\,N$	static vertical load axle 3
$F_{Z,4}$	$17658\,N$	static vertical load axle 4
$[\frac{k_b}{r_{1,2,3}}\ \frac{k_b}{r_4}]$	$[-3600\ -2000]\,\frac{N}{bar}$	relation between brake pressure and braking force on axle 1-2-3 and on the last axle
$[w_1\ w_2\ w_3\ w_4]$	$[2.05\ 1.85\ 1.85\ 2.05]\,m$	track width of the axle 1-2-3-4
$[l_{12}\ l_{13}\ l_{14}]$	$[3.2\ 4.57\ 5.94]\,m$	distance axle 1-2, distance axle 1-3, distance axle 1-4
$[\tau_b\ \tau_{RAS}]$	$[0.1\ 0.4]\,s$	motion actuators time constants
$W_v(1,1)$	$0.1\,\frac{1}{N^2}$	weighting matrix
$W_v(2,2)$	$100\,\frac{1}{(Nm)^2}$	weighting matrix
γ	0.01	scalar for the prioritization
N	10	steps in the minimization
T	$0.05\,s$	sampling time used in predictive actuator model
T_s	$0.02\,s$	controller sample time

no states of the vehicle such as yaw velocity (ω_Z) or vehicle sideslip angle (β) have been fed back to any of the controllers. This feed-forward nature of the controllers permits better transparency in showing limitations and benefits of the predictions. The vehicle model used during simulations was a non-linear model developed by Volvo Trucks. The vehicle model includes all the important features needed to simulate the dynamics of a heavy vehicle (suspensions, body compliances, magic formula with relaxation for the tyres model, ...). The vehicle used in the real vehicle tests was a Volvo FMX 8x4 tag axle, that means the vehicle had 4 axles with second and third axle appointed to driven axles and RAS collocated on the last axle. This vehicle is different from that being used in simulation. The purpose with this is to show the flexibility in using MPCA and CA, meaning that the low-level controller easily can be adapted to different vehicle configurations. Simulations were also run with a 8x4 vehicle model prior to real vehicle tests in order to validate the controller. These results are however not included in here. The parameters used during simulations and tests can be found respectively in Table I and Table II.

A. Split-μ braking tests

The tests involving the truck have been run on a handling and braking track, where a surface of wet basalt has been used to get low friction conditions. To assure repeatability, 5 split-μ braking events were performed with each controller, starting from $v_0 = 50\,km/h$. The friction coefficients calculated for the high and low conditions were $\mu_H = 0.55$ and $\mu_L = 0.08$, the static vertical loads $F_{Z,i}$ were calculated with a specific scale for vehicles. The following figures are representative of the typical behaviour of vehicle and driver during the manoeuvre. Fig. 5 shows how RAS behaved during the initial seconds of the braking event with MPCA and CA. Due to the rate limits necessary to get $\delta_{cmd} \approx \delta$, CA requires more time to turn the wheels of the tag axle and the brakes have to wait longer before they can brake asymmetrically without producing a yaw moment on the vehicle. Fig. 6 illustrates how this limitation makes MPCA build up the deceleration for the vehicle faster. With this difference in decelerations, the vehicle equipped with MPCA travels about $1\,m$ shorter during the initial transition of the braking event. Apart from that, both controllers have shown comparable and satisfying performance regarding the effort required by the driver to maintain the vehicle on a straight line. This is illustrated in Fig. 7 where the used steering wheel angle is shown. It can be noted that the driver never had to correct the direction of the vehicle using more than 18° of steering wheel angle.

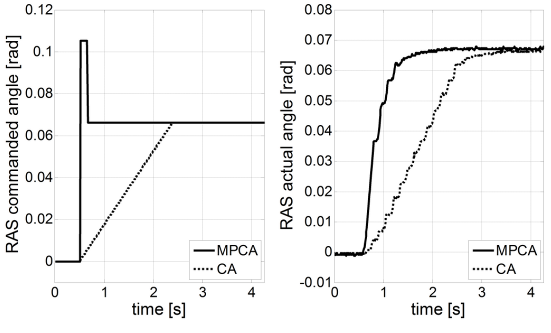

Fig. 5. Split-μ braking tests: RAS commanded angle (left) and RAS actual angle (right) when using MPCA and CA.

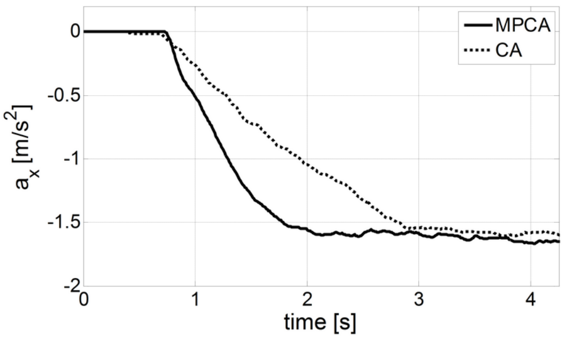

Fig. 6. Split-μ braking tests: vehicle deceleration with MPCA and CA.

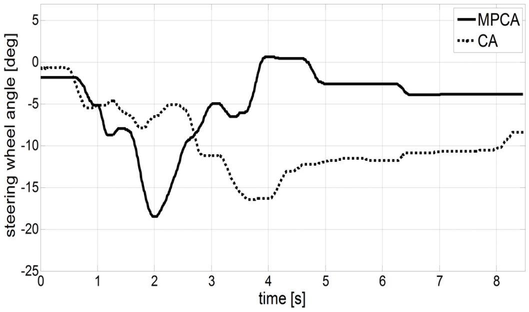

Fig. 7. Split-μ braking tests: steering wheel angle with MPCA and CA.

B. Split-μ braking simulations

The simulations, run before the real tests, have been essential for the general design of the controller. In this scenario, $v_0 = 50\,km/h$, $\mu_H = 0.7$ (dry asphalt) and $\mu_L = 0.1$ (icy road). During the braking event, an external driver actuating on the steering wheel has been implemented as a smooth PID that takes care of following a straight path. The input of the PID was the lateral deviation of the vehicle, while the output was the steering wheel angle. Fig. 8 shows the lateral position of the vehicle with MPCA and CA. The vehicle starts braking at $1\,s$ and the maximum lateral deviation, reached after about $2.3\,s$ with MPCA and $3.4\,s$ with CA, is less than $0.15\,m$ in both cases. During the manoeuvre, the maximum steering wheel angle used by the driver model was $15°$. As already observed in the results from real testing, Fig. 9 shows that high deceleration is reached faster when using MPCA compared to CA. In particular, when both decelerations reach their steady-state values, the vehicle equipped with MPCA has travelled $1\,m$ shorter than the vehicle equipped with CA.

C. Split-μ acceleration simulations

In the split-μ acceleration scenario, $v_0 = 0\,km/h$, $\mu_H = 0.7$ and $\mu_L = 0.1$ when the driver decides to accelerate in order to move the vehicle. In this scenario, the simulations showed that MPCA and CA exhibit almost identical performance and the actuators dynamics showed not to be of primary importance.

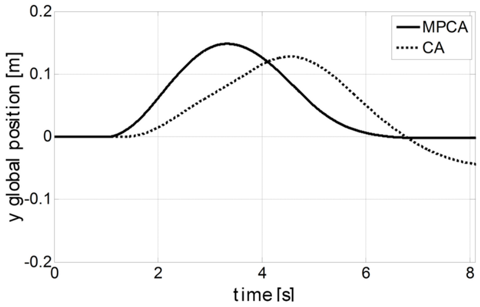

Fig. 8. Split-μ braking simulations: vehicle deviation with MPCA and CA.

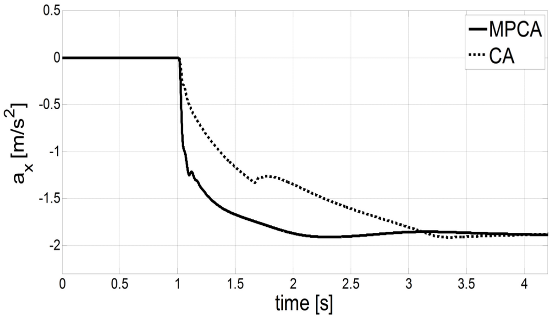

Fig. 9. Split-μ braking simulations: vehicle deceleration during the initial transient when using MPCA and CA.

D. Brake blending simulations

In this scenario, $\mu = 0.7$ for all the wheels and the vehicle starts braking from an initial speed $v_0 = 50\,km/h$. Fig. 10 shows the sum of the forces generated by all the disc brakes, the force generated by the engine brake and the total longitudinal force acting on the vehicle when using MPCA and CA during a modest braking. It can be noted how MPCA is able to reach the steady state value of total longitudinal force faster than CA. Moreover, MPCA has a smoother behaviour when the pressure on the brakes starts to be released and the engine brake torque ramps up.

During a mild braking event, when not only the engine brake is used in steady-state conditions, it has been verified that the braking force on each axle is distributed according to the axle load in both set-ups. Table III shows the values of the vehicle when it is braking in steady-state conditions using MPCA and it is clear that all the axles are using the same amount of available friction.

$$\kappa_j = \frac{F_{X,j}}{F_{Z,j}} = 0.12 \;\; j = 1, 2, 3 \qquad (14)$$

where $F_{X,j}$ and $F_{Z,j}$ are respectively the resulting longitudinal force and normal load on the axle j. F_X and F_Z are respectively the resulting longitudinal force and normal load on the vehicle.

TABLE III

BRAKING FORCE DISTRIBUTION

	$\frac{F_{Z,j}}{F_Z}$	$\frac{F_{X,j}}{F_X}$
Axle 1	0.318	0.319
Axle 2	0.461	0.461
Axle 3	0.220	0.219

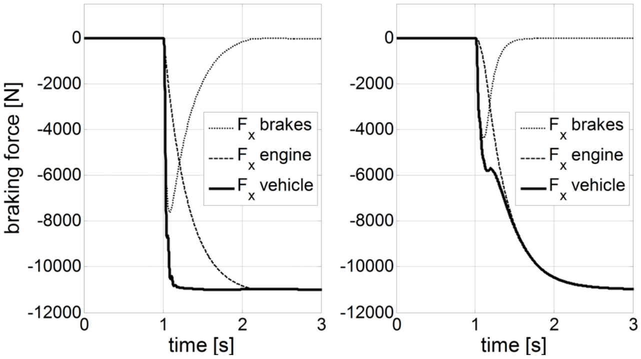

Fig. 10. Combination of disc brakes and engine brake when using MPCA (left) and CA (right)

V. CONCLUSIONS

The designed MPCA controller has proved to be able to cope with the three different scenarios. Compared to CA, MPCA presents a faster response and an improvement of the vehicle performances in transients. In the split-μ braking scenario this responsiveness results in a shorter time to reach the desired deceleration for the vehicle. The difference between the two controllers is appreciable when the motion actuators are intensely exploited. Moreover, when the assumption $\delta_{cmd} \approx \delta$ is wrong, CA does not perform as good as MPCA. This phenomena has especially been observed during the tuning of the rate limit for RAS in the split-μ braking simulations. If the rate limit is too high the brakes on the high friction side produce a yaw moment on the vehicle too early and requires the driver to steer more and more in order to maintain the vehicle on the course. On the other hand, if there is no need to deeply exploit the motion actuators, set conservative rate limits in CA facilitates the handling of the vehicle as forces and moments are produced smoother.

In the brake blending scenario, MPCA is able to fully use the disc brakes before the engine torque ramps up, resulting in a faster achievement of the requested longitudinal force on the vehicle and a gentle combination of the two different types of actuators, while in the split-μ acceleration scenario the performances of the two controllers are indiscernible.

The cost of explicitly considering the motion actuators dynamics in MPCA is a higher computational effort demanded to the designated processor. The computational time to solve the MPCA problem during the tests was, on average, of $10\,ms$, compared with the average $0.7\,ms$ to solve the CA problem. Although it is difficult to compare the performance of a MicroAutoBox II with a commercial ECU, MPCA computational time indicates good chances of achieving a feasible implementation of the method in a standard ECU within suitable sample times $(50 - 100\,Hz)$. As the designed MPCA has a time horizon of $0.5\,s$, 10 steps in the objective function (4) with the models (5) discretized at $0.05\,s$, it would be interesting to further study if a change of these values leads to a faster MPCA without deteriorating its performances. Future work would also include the extension of the method to new vehicle configurations with trailers and semi-trailers, apart from new test cases that consider a wider set of manoeuvres, e.g. steady-state curves. Finally, the implementation of a real-time estimator for the tyre-road friction coefficient would probably be the major contribution that helps understanding the potentialities of the controller in a day-to-day environment.

REFERENCES

[1] K. Tagesson, P. Sundström, L. Laine, and N. Dela, "Real-time performance of control allocation for actuator coordination in heavy vehicles," in *Proc. IEEE Intelligent Vehicles Symposium*, Xi'an, China, June 2009, pp. 685–690.

[2] J. Eklöv, "Real-time implementation of a vehicle motion coordination for a single unit truck," Master's thesis, Chalmers University of Technology, Göteborg, Sweden, 2013.

[3] K. Uhlen, P. Nyman, J. Eklöv, L. Laine, M. S. Kati, and J. Fredriksson, "Coordination of actuators for an a-double heavy vehicle combination using control allocation," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, Qingdao, China, Oct. 2014, pp. 641–648.

[4] K. Tagesson, L. Laine, and B. Jacobson, "Combining coordination of motion actuators with driver steering interaction," *Traffic Injury Prevention*, vol. 16, pp. 18–24, June 2015.

[5] Y. Luo, A. Serrani, S. Yurkovich, M. Oppenheimer, and D. Doman, "Model predictive dynamic control allocation with actuator dynamics," in *Proc. IEEE American Control Conference*, Boston, USA, June 2004, pp. 1695–1700.

[6] ——, "Dynamic control allocation with asymptotic tracking of time-varying control input commands," in *Proc. IEEE American Control Conference*, June 2005, pp. 2098–2103.

[7] ——, "Model-predictive dynamic control allocation scheme for reentry vehicles," *Journal of Guidance, Control and Dynamics*, vol. 30, pp. 100–113, Jan. 2007.

[8] M. Hanger, T. A. Johansen, G. Mykland, and A. Skullestad, "Dynamic model predictive control allocation using cvxgen," in *Proc. IEEE International Conference on Control and Automation*, Santiago, Dec. 2011, pp. 417–422.

[9] C. Vermilion, J. Sun, and K. Butts, "Model predictive control allocation for overactuated systems - stability and performance," in *Proc. IEEE Decision and Control*, New Orleans, USA, Dec. 2007, pp. 1251–1256.

[10] ——, "Predictive control allocation for a thermal management system based on an inner loop reference model design, analysis, and experimental results," in *Proc. IEEE Control Systems Technology*, July 2011, pp. 772–781.

[11] C. Satzger, R. Castro, and T. Bunte, "A model predictive control allocation approach to hybrid braking of electric vehicles," in *Proc. IEEE Intelligent Vehicle Symposium*, Deaborn, USA, June 2014, pp. 286–292.

[12] ISO 8855:2011, "Road vehicles - vehicle dynamics and road-holding ability - vocabulary," International Organization for Standardization, Dec. 2011.

[13] J. Mattingley and S. Boyd, "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, pp. 1–27, Oct. 2011.