



Volume 58 Number 7 July 2020
ISSN 0042-3114
International Journal of Vehicle Mechanics and Mobility
Official Organ of the
Editorial Board
International Association for Vehicle System Dynamics
Motoo Shimada
Shinichiro Horie
Satoshi Iwaki
Masaru Fukui



Vehicle System Dynamics

International Journal of Vehicle Mechanics and Mobility

ISSN: 0042-3114 (Print) 1744-5159 (Online) Journal homepage: <https://www.tandfonline.com/loi/nvsd20>

Real-time control for at-limit handling driving on a predefined path

T. Novi, A. Liniger, R. Capitani & C. Annicchiarico

To cite this article: T. Novi, A. Liniger, R. Capitani & C. Annicchiarico (2020) Real-time control for at-limit handling driving on a predefined path, *Vehicle System Dynamics*, 58:7, 1007-1036, DOI: [10.1080/00423114.2019.1605081](https://doi.org/10.1080/00423114.2019.1605081)

To link to this article: <https://doi.org/10.1080/00423114.2019.1605081>



Published online: 15 Apr 2019.



Submit your article to this journal



Article views: 493



View related articles



View Crossmark data



Citing articles: 2 View citing articles



Real-time control for at-limit handling driving on a predefined path

T. Novi^a, A. Liniger^b, R. Capitani^a and C. Annicchiarico^c

^aDepartment of Industrial Engineering, Università degli Studi di Firenze, Florence, Italy; ^bDepartment of Information Technology and Electrical Engineering, Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland; ^cMeccanica 42 S.r.l, Sesto Fiorentino, Italy

ABSTRACT

Autonomous driving at at-limit handling represents a big challenge due to the nonlinear behaviour of a vehicle. In this paper, a hierarchical control scheme composed of two Nonlinear Model Predictive Controls (NMPCs) for path following is proposed. This structure allows for real-time feasibility without having to simplify excessively the vehicle model. The higher level NMPC operates on a long prediction horizon with a point-mass model which constraints are given by a tyre dependant ‘g-g diagram’. The output of the higher level is a velocity profile which is used as terminal constraint by the lower level NMPC. This operates on a short horizon and is based on a seven degrees of freedom vehicle model with full Pacejka Magic Formula tyre formulation on all tyres, load transfers and Limited Slip Differential. Because of the terminal set computed by the higher level, it is possible to use a short horizon and exploit the vehicle performance in real-time. For both controllers, the full Nonlinear Optimisation Problem is solved at each step. The algorithm is tested in a Model-in-the-Loop environment where a validated Vi-Grade vehicle model is controlled in co-simulation. The results show the effectiveness and feasibility of real-time control with complex vehicle models.

ARTICLE HISTORY

Received 21 August 2018
Revised 12 February 2019
Accepted 30 March 2019

KEYWORDS

Autonomous driving; path following; limit handling; nonlinear model predictive control

1. Introduction

Driving at at-limit handling and exploiting vehicle performance has always represented a big challenge in motorsport [1–4]. In such conditions, the tyres are operating in their nonlinear range and the vehicle is generally unstable, as even a small variation in the contact between tyres and road can induce a rapid variation of grip [5]. Additionally, the feedback that a driver gets from the vehicle is limited due the human’s inability to process such feedback in the short amount of available time. At-limit handling is important also in every day driving where critical manoeuvres are often necessary, i.e. sudden obstacle avoidance manoeuvres [6].

Besides this, Advanced Driver Assistance Systems (ADAS) have increased in numbers and complexity. Nowadays, all car manufacturers equip their vehicles with these kind of devices which are now becoming mandatory, e.g. automatic emergency braking. In Europe this has been dictated by the desired reduction of number of fatalities that the European Commission has set for the upcoming years [7]. The growth of ADAS has led to large interest in Autonomous Driving (AD), for which the Society of Automotive Engineering (SAE) have come up with different levels of AD [8]. With a robotic controller driving the vehicle, the maximum performance that can be obtained could be enhanced since all commands can be actuated in a faster, more reliable and repeatable way.

The rest of paper is organised as follows. In section two, a detailed literature review is done and the contribution of this paper is presented. Section three describes modelling, composed of the vehicle model of the low-level controller and point-mass dynamics of the high-level controller. In Section 4, the hierarchical controller scheme will be discussed in detail and the mathematical framework used for both levels will be described. Finally, in Section 5, results obtained with the proposed algorithm tested on a commercial software will be discussed.

2. State of the art

In this section the literature is reviewed. First, lap time optimisation methods are analysed. These algorithms find the optimal control sequence which minimises time. The inputs are computed offline, thus, cannot be applied for AD. However, they represent the basics for controlling an autonomous vehicle at at-limit handling. Second, algorithms which control autonomous cars are analysed, with a focus on those which exploit vehicle performance. Finally, the contributions of this paper are presented.

2.1. Lap time optimisation

The first attempts to find the sequence of inputs that maximise vehicle performance were done with offline simulations for motorsport applications. The goal was to obtain the minimum lap time control input sequence and the optimal trajectory. The first lap time simulation was done by Mercedes Benz in the 1930s [9]. In the past decades, many different researchers have studied this topic in various ways. One approach is to simulate the Quasi-Steady-State (QSS) conditions. Brayshaw et al. [10] used the QSS method on a pre-defined path. The algorithm starts by calculating the vehicle's acceleration limits from a series of constrained nonlinear optimisation problems (NLPs). Then, peaks in the curvature data are identified as an apex of a corner and the maximum possible steady-state speed for each curve is found. The maximum acceleration and deceleration between apex pairs is then calculated and, finally, speed profiles are obtained together with the input sequence. Using the same method, Tremlett et al. [11] studied the influence of limited slip differential (LSD) on lap time simulation.

A more realistic approach is the one based on transient vehicle behaviour and optimal control techniques. This approach can be divided in two main categories, direct and indirect methods. The former have the advantage that they do not require the adjoint equations which become more complicated to derive as the vehicle model and boundary constraints increase in complexity. However, the latter are more accurate and there is not a risk of

incurring into a local minimum. Direct methods were first applied by Casanova et al. [12] who used direct multiple shooting applied to a seven degrees of freedom nonlinear vehicle model solved with Sequential Quadratic Programming (SQP). A very similar approach was used by Kelly and Sharp [13,14] who additionally implemented a thermodynamic tyre model. Tremlett and Limebeer [15] also apply a study on tyre usage in terms of temperature and wear with these algorithms. Perantoni and Limebeer [16] apply direct lap time optimisation to study a set of vehicle parameters on a formula type vehicle. One of the issues when trying to minimise lap time with optimal control is the objective function formulation. In fact, it is non-trivial if the best optimisation strategy is to minimise time or maximise velocity. This problem is addressed by means of direct optimisation, by Velenis and Tsiotras [17].

To the authors knowledge, indirect methods were first applied by Tavernini et al. to analyse optimal handbrake cornering [18] and the effect of road surface and car transmission layout [19]. Tremlett et al. [20] also use this method to study optimal control of motorsport differentials while Rucco et al. [21,22] improve the optimisation algorithm by including a derivation of suitable transverse coordinates that allow splitting of the dynamics into longitudinal and lateral and propose a continuation method for the intermediate trajectories. Lot et al. [23] use indirect methods to study lap time optimisation of a go-kart and use an independent variable transformation from time to space. Dal Bianco et al. [24] also consider a suspension model to a formula type vehicle to analyse how suspensions influence lap time.

All of these papers require a long time to solve the optimisation problem; to improve the computation time and move towards real-time implementation Velenis et al. developed a semi-analytical model that allows generation of the optimal velocity profile by applying Pontryagin's Maximum Principle. This was done first on a point-mass material model [25] and then extended to a half-car model [26]. To apply this in real-time, a receding horizon strategy was also applied [27,28].

2.2. Towards autonomous driving

For AD it is crucial for the controller to be executed in real-time, which to guarantee reasonable performance means between one and one hundred milliseconds, depending on the exact task. This is linked to the dynamics which are controlled. Generally, the lower the lever of the controller the higher the speed should be. Hence, one millisecond update rate can be referred to controllers such as stability controllers where the dynamics are very fast whereas one hundred milliseconds update rate can be referred to navigation tasks where the dynamics are much slower since the decision-making process changes relatively slowly, based on the estimated traffic en-route. This requirement renders most of the previously discussed approaches not applicable. However, over the last decades many different control techniques have been developed for AD. In the methods presented here, controllers that focus on at-limit handling are discussed. These controllers can be divided into two groups, first the ones that separate longitudinal and lateral dynamics, and second the ones that integrate all inputs into one controller. Within the first category is the controller of Zhang et al. [29], who developed a lateral controller only by means of sliding mode using a first order sliding surface to control lateral velocity and yaw rate. More recently, Klomp et al. [30] worked on a pure lateral controller for at-limit driving using a preview path curvature and

using a feedforward and feedback steering control calculated with a linear bicycle model. The most interesting approach that falls in this category is the one of Kritayakirana and Gerdes [31,32], who developed a controller considering clothoid curves as trajectory and a quasi-static linearised bicycle model. A feedforward steering and longitudinal input are calculated based on the understeering gradient [33] and the ellipse adherence of the tyres. Feedback controllers are then added to assure at-limit handling driving and vehicle stability. Specifically: a lanekeeping feedback with lookahead for tracking purposes, a yaw damping feedback to minimise yaw oscillation and a slip circle feedback which is used to control rear tyre slip so that rear axle saturation is avoided. The same authors tried using the centre of percussion [34] for the feedforward and feedback steering control to increase controller robustness to external disturbances coming from the rear axle.

In the other category are the controllers that combine lateral and longitudinal behaviour in one algorithm. This is mostly done with MPC. Prokop [35] are within the first to have explored this technique and showed how MPC is very similar to the way that human drivers plan their trajectories when driving. However, the interest of this paper is on tracking rather than path planning. Despite being used for lateral control only, some of the first papers on MPC in AD for tracking purposes are the ones of Borrelli et al. [36] and Keviczky et al. [37]. Also Kraus et al. [38] and Cui et al. [39] use a similar approach and at the same time estimate the vehicle states, however, while operating far from tyre saturation. The approach of Borrelli et al. was extended by Falcone et al. [40] to control all inputs of the vehicle. The MPC was solved with a Linear Time Varying (LTV) method to decrease the computational complexity given by the full NLP, although a very simplified vehicle model is used in the formulation presented.

Focusing on the control of the vehicle at its limits, the LTV-MPC was used also by Timings and Cole [41,42] for lateral control only with the goal, thus objective function of the optimisation, of maximising progression along the track.

To obtain the real optimal control input sequence and exploit the vehicle's performance, it is necessary to consider high fidelity vehicle models, properly define an objective function with which time can be minimised, and solve the NLP without linearising while maintaining real-time feasibility. Also, one of the critical aspects is the horizon length of the MPC. Since a full scale vehicle driving around a track requires long braking distance, especially at high speed, to complete an entire lap long horizons are generally required for the robotic controller.

Four different approaches have been developed in recent years, all of which address some of the presented issues.

The first method proposed by Liniger and Lygeros [43] uses a hierarchical control structure, where in the upper level the progress optimal trajectory is computed using a tree search algorithm and a simplified vehicle model. In the lower level the previously computed trajectory is tracked using an MPC, which uses a more accurate vehicle model with nonlinear tyre forces. To deal with the issue of the horizon length, viability theory is used to compute a terminal constraint that guarantees safety of all trajectories remaining in that set. This terminal constraint allows to run shorter horizon and speed up the tree search. The same authors extended the method in [44] to allow for game theoretic decision making of two cars racing head-to-head. However, the viability theory approach used in these papers requires the terminal constraint to be pre-computed and this operation requires a few hours, thus it is not possible to use in real-time.

The second is based on an iterative MPC in which the autonomous driver learns lap after lap as for Rosolia et al. [45,46]. This approach is however computationally very expensive and requires the car to drive around the circuit many times before performing at its limits.

Another approach is to use the Model Predictive Contouring Control (MPCC) formulation as suggested by Liniger et al. [47]. With this method, the controller's objective is a trade-off between the progress along the track and the contouring error. In this paper, the Quadratic Programming (QP) approximation is solved by means of an interior point method, which allows fast sampling times and long horizons.

The final approach which can be found in the literature is to transform the state-space into space dependant; this method has the great advantage of having an explicit formulation of time, which can be used as an objective function. On the other hand, one of the disadvantages is that the horizon is space dependant, thus, for higher velocities, the same lookahead as for low velocities is used. This approach was first used by Gao et al. [48]. An extension of this but with a more complete vehicle model was done by Frasch et al. [49] who include also load transfer and wheel dynamics. These controllers maintain real-time feasibility, however, a manoeuvre with a very short horizon is simulated, e.g. an obstacle avoidance manoeuvre with approximately 25 m lookahead. Also, these authors solve the QP approximation obtained by linearising the NLP. This method is also known as Real Time Iteration (RTI) and allows solution of the optimisation very quickly but most of the time resulting in a suboptimal solution due to only solving one QP, instead of solving the SQP approach to convergence. Verschueren et al. [50,51] use the same approach, however focusing on an autonomous racing application and using simpler models to facilitate real-time implementation. Finally, Anderson et al. [52] also transform the problem into space dependant and integrate a genetic algorithm to adjust the costs of the optimisation objective function between minimum time and maximum exit velocity. The main disadvantage of this approach is the high computational burden to solve the genetic algorithm.

The main issues with all the discussed papers are that none of them are using a high fidelity model and that the used prediction horizons are not long enough for autonomous racing on a race track. This leads mainly to two shortcomings, first, the controller cannot achieve the best possible performance the vehicle could deliver since in this range of a vehicle's performance even little changes can make a big difference. Second, to accurately predict critical decision points, long horizons are needed, this is very important, for example, for braking points.

2.3. Contribution

In this paper, a robotic controller for AD that can tackle the previously discussed shortcomings of existing methods is proposed. A novel hierarchical control scheme composed by two NMPCs is introduced.

The contributions here are threefold; first, a novel hierarchical structure is proposed, wherein the upper level computes a velocity profile given a simple point-mass model with acceleration constraints representing the tyre limits. Due to the simplicity of this problem, it can be solved for long prediction horizons (over several hundred metres). The velocity profile is then used as a terminal constraint in the lower level, which can run on short prediction horizons and thereby retain real-time feasibility. By solving both problems in a receding horizon fashion, this method allows for maximal flexibility against

disturbances and it would even allow updating the path online. This is the main advantage compared to computing the velocity profile with a more sophisticated model offline. Additionally, it is possible with the proposed algorithm to run the controller in less controlled environments.

The second contribution is the low-level controller. Even though the controller is similar to the one found in [49], the approach presented uses a higher fidelity model using the full Pacejka Magic Formula 6.1 [53] with all combined effects. Additionally, an LSD formulation is included in the dynamics and the torque limits of the engine-gearbox drivetrain are realistically approximated. Also, this approach does solve the full NLP and not only approximates it using an RTI method.

Finally, the performance of the controller is verified using the commercial software Vi-Grade and a validated vehicle model in a Model-in-the-Loop simulation. The simulations show that the method is indeed able to control a car at-limit handling and achieves expert human driver-like performance.

3. Modelling

In this section, all aspects concerning the vehicle models used in the control scheme will be discussed. Specifically, the high fidelity vehicle model used for the low-level control is described, which consists of planar dynamics, vertical dynamics, wheel dynamics and tyre forces. Furthermore, the point-mass dynamics and acceleration limits used in the high-level controller are discussed.

3.1. Vehicle model

The vehicle model used for the low-level controller is a seven degrees of freedom model which describes the planar dynamics of the vehicle, neglecting heave, pitch and roll motions. To properly consider vertical force tyre sensitivities and eliminate algebraic loops, a dynamic formulation of both longitudinal and lateral load transfers are considered. Additionally, the variation of vertical load due to aerodynamic effects is present in the model. Wheel dynamics are properly modelled including the effect of LSD since at at-limit handling, the presence of a LSD heavily influences the vehicle dynamics [54] especially during combined accelerations. The tyres are modelled with a MF6.1 formulation comprising of all combined effects. However, aligning and overturning moments generated by the tyres are neglected.

The engine torque is modelled considering maximum and minimum torque at the wheel for each gear and using the envelope curve as a torque limit over the entire range of wheel speeds. This is possible since an additional gear controller which shifts gear at pre-determined wheel speeds is implemented. Note that this type of controller is based on a discrete state machine which also uses the time from the previous gear change as a control input, this assures no oscillation between gears. This is the standard gear controller used in the commercial software Vi-grade.

The input vector u is composed of engine torque T_e , braking pedal percentage B_p and steering wheel angle δ_w

$$u(t) = (T_e, B_p, \delta_w). \quad (1)$$

The 12-dimensional state vector is composed of positions, yaw angle, lateral and longitudinal velocities, yaw rate, wheel speeds and vertical load transfer. Since each state is discussed in detail further on, here the state vector is only defined and the reader is referred to the following paragraphs for the exact formulation,

$$\mathbf{x}(t) = (X, Y, \psi, v_x, v_y, \dot{\psi}, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}, \Delta F_{zx}, \Delta F_{zy}) . \quad (2)$$

3.1.1. Planar dynamics

The four-wheel classic seven degrees of freedom model describes longitudinal position X , lateral position Y , and yaw angle ψ of a vehicle with respect to a fixed reference system as well as longitudinal velocity v_x , lateral velocity v_y , and yaw rate $\dot{\psi}$ with respect to a vehicle reference system with origin in its centre of gravity (C_g) and vertical axis pointing upwards. The vehicle model described is illustrated in Figure 1. The equations of motion of positions and yaw angle with respect to the fixed reference system are described as follows,

$$\dot{X} = v_x \cos(\psi) - v_y \sin(\psi), \quad (3)$$

$$\dot{Y} = v_y \cos(\psi) + v_x \sin(\psi), \quad (4)$$

$$\dot{\psi} = d\psi/dt. \quad (5)$$

To formulate the equation of motion for the remaining states, the following subscripts are introduced to refer to individual wheels: $(*)_{as}$ where $a \in \{f, r\}$ denotes front and rear axis and $s \in \{l, r\}$ denotes the left and right side, respectively. The front-left and front-right steering angles, respectively, δ_{fl} and δ_{fr} are calculated with Ackermann geometry

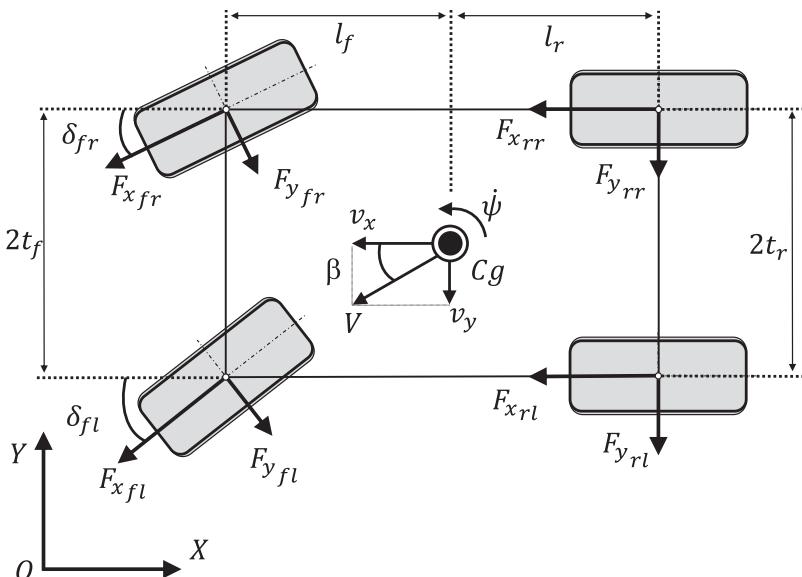


Figure 1. Vehicle model scheme.

as follows:

$$\delta_{fl} = \frac{2\delta(l_f + l_r)}{2(l_f + l_r) - \delta t_f}, \quad (6)$$

$$\delta_{fr} = \frac{2\delta(l_f + l_r)}{2(l_f + l_r) + \delta t_f}. \quad (7)$$

Where δ is the steering wheel angle at the front axis which is obtained by considering the steering ratio given by the pinion-rack engagement. Even though Ackermann geometry is considered in the NMPC model, to simplify the notation in the following equations Ackermann steering is neglected and $\delta_{fr} = \delta_{fl}$. Considering the front and rear wheelbase l_f and l_r , front and rear track t_f and t_r , vehicle mass m and yaw inertia J_{zz} , longitudinal tyre forces $F_{x_{as}}$ and lateral tyre forces $F_{y_{as}}$, air density ρ , front vehicle surface S , vehicle drag coefficient C_x and neglecting static toe, the planar dynamics can be written as

$$m\dot{v}_x = F_x - mv_y\dot{\psi}, \quad (8)$$

$$m\dot{v}_y = F_y + mv_x\dot{\psi}, \quad (9)$$

$$J_{zz}\ddot{\psi} = M_z, \quad (10)$$

where the total forces F_x and F_y and moment M_z acting on the vehicle are described as

$$F_x = \cos(\delta)(F_{x_{fl}} + F_{x_{fr}}) - \sin(\delta)(F_{y_{fl}} + F_{y_{fr}}) + F_{x_{rl}} + F_{x_{rr}} - \rho S C_x v_x^2 / 2, \quad (11)$$

$$F_y = \cos(\delta)(F_{y_{fl}} + F_{y_{fr}}) + \sin(\delta)(F_{x_{fl}} + F_{x_{fr}}) + F_{y_{rl}} + F_{y_{rr}}, \quad (12)$$

$$\begin{aligned} M_z = & \cos(\delta)(F_{x_{fr}} - F_{x_{fl}})t_f + \sin(\delta)(F_{x_{fl}} + F_{x_{fr}})l_f \\ & + \cos(\delta)(F_{y_{fl}} + F_{y_{fr}})l_f + \sin(\delta)(F_{y_{fl}} - F_{y_{rl}})t_f \\ & + (F_{x_{rr}} - F_{x_{rl}})t_r - (F_{y_{rl}} + F_{y_{rr}})l_r. \end{aligned} \quad (13)$$

3.1.2. Wheel dynamics

The forces acting on the vehicle are strongly influenced by the wheel's dynamics. Specifically: engine, gearbox, differential and wheels are considered in this model, comprehensive of inertia, transmission ratio and torque distribution. A visual scheme of the layout is shown in Figure 2.

In this model the following effects are neglected: the clutch engagement (the clutch is considered to be one with the gearbox) and the losses in the transmission (unitary efficiency). This allows the modelling of the input torque to the differential T_d as a function of engine torque, $T_d = T_e / (\tau_g \tau_d)$, where τ_g and τ_d are the transmission ratios of the gearbox, and differential respectively. Since the controller will be operating mainly at at-limit handling, it is important for the controller to predict combined adherence for each tyre to maximise performance. For this reason, the model should correctly predict how slip ratio and slip angle evolve as a consequence of the control input. Therefore, wheel speed ω_{as} of each wheel needs to be correctly modelled. In this paper a Rear Wheel Drive vehicle is considered, thus, the front wheels are only affected by braking torques, whereas the rear wheels are also influenced by engine torque. Braking torques of each wheel are computed based on front and rear brake transfer functions from brake percentage to brake torque k_f^b and

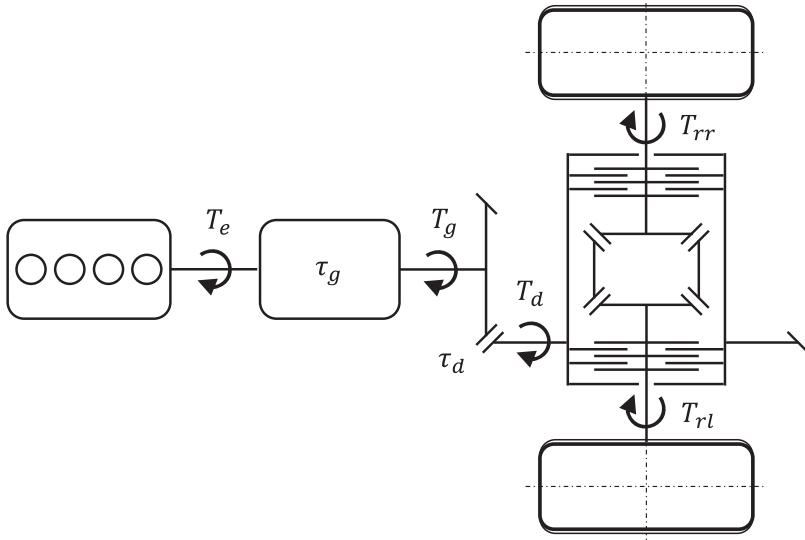


Figure 2. Transmission layout.

k_r^b (i.e. depending on master cylinder, brake disc radius, friction coefficient and number of brake pistons), front to rear brake biases b_b , differential locking percentage k_l^d , and wheel rolling resistance M_{yas} . To correctly model the torque of the drivetrain at each rear wheel it is also important to consider the LSD effect since at at-limit handling locking of the differential occurs and torque is transferred from the faster wheel to the slower one. Since a LSD model causes the wheel dynamics to be discontinuous, which in turn would prohibit the use of continuous optimisation solvers, two simplifications are necessary. First, preload is neglected and second the hyperbolic tangent function is used to approximate the sign function normally present in LSD models. Based on the above-described torques the wheel dynamics can be written as,

$$J_{rotfl}\dot{\omega}_{fl} = -(B_p k_f^b b_b) - F_{xfl}R_{fl} - M_{yfl}, \quad (14)$$

$$J_{rotfr}\dot{\omega}_{fr} = -(B_p k_f^b b_b) - F_{xfr}R_{fr} - M_{yfr}, \quad (15)$$

$$J_{rotrl}\dot{\omega}_{rl} = 1/2 T_d [1 - k_l^d \tanh(\omega_{rl} - \omega_{rr})] - B_p(1 - b_b)k_r^b - F_{xrl}R_{rl} - M_{yrl}, \quad (16)$$

$$J_{rotrr}\dot{\omega}_{rr} = 1/2 T_d [1 + k_l^d \tanh(\omega_{rl} - \omega_{rr})] - B_p(1 - b_b)k_r^b - F_{xrr}R_{rr} - M_{yrr}, \quad (17)$$

where J_{rot_*} is the inertia of the respective wheel including the brake rotor and drive shaft in the case of the rear wheels.

3.1.3. Vertical dynamics

Vertical dynamics ought to be considered in the model since tyres have a nonlinear characteristic with respect to vertical forces which vary due to planar dynamics, i.e. load transfer. However, load transfers are again depending on tyre forces and thus, vertical forces; this causes an algebraic loop. One possible way to resolve this issue is to introduce a dynamical system describing vertical loads. More precisely, following [55] a dynamical system for the

longitudinal and lateral weight transfers is formulated. Given these weight transfers, vertical load acting on each tyre $F_{z_{as}}$ can be expressed as the sum between static vertical load $F_{z0_{as}}$, longitudinal and lateral weight transfers ΔF_{zx} and ΔF_{zy} and downforce $F_{l_{as}}$. Hence, the total vertical force acting on each tyre is expressed as

$$F_{z_{as}} = F_{z0_{as}} \pm \Delta F_{zx} \pm \Delta F_{zy} t_a / t_{avg} + F_{l_{as}}, \quad (18)$$

where the sign in front of ΔF_{zx} is negative for the front wheels and positive for the rear wheels and the sign in front of ΔF_{zy} is positive for the right wheels and negative for the left wheels. To account for front to rear lateral load transfer distribution, the lateral load transfer is multiplied by t_a , which is the front track for the front wheels' vertical load calculation and rear track for the rear wheels' vertical load calculation, and divided by t_{avg} , which is the average between front and rear track. The static vertical load for each tyre can be found with simple geometrical consideration and is found as follows,

$$F_{z0_{fl}} = F_{z0_{fr}} = \frac{mgl_r}{2(l_f + l_r)}, \quad F_{z0_{rl}} = F_{z0_{rr}} = \frac{mgl_f}{2(l_f + l_r)}. \quad (19)$$

The aerodynamic load acting on each tyre can be expressed considering the vehicle's lift coefficient C_z and is given by $F_{l_{as}} = (\rho S C_z / 8) v_x^2$. Finally, dynamic load transfers are expressed as a first-order delay system. These equations are characterised by a longitudinal time constant τ_x and lateral time constant τ_y , which implicitly represent equivalent pitch and roll suspension stiffness, and centre of gravity's height from the ground h

$$\begin{aligned} \Delta \dot{F}_{zx} &= \frac{1}{\tau_x} \left(\frac{h}{2(l_f + l_r)} F_x - \Delta F_{zx} \right), \\ \Delta \dot{F}_{zy} &= \frac{1}{\tau_y} \left(\frac{h}{2(t_f + t_r)} F_y - \Delta F_{zy} \right). \end{aligned} \quad (20)$$

3.1.4. Actuator dynamics

We consider a simple actuator dynamics model, where the inputs cannot be changed directly but the rate of change can be controlled. This corresponds to a simple integrator dynamics. This also allows to limit the rate of change of the real inputs and optimise for smooth input trajectories. However, this results in a larger state vector \hat{x} which also contains the (real) inputs, and the input \hat{u} to the system become the rate of the inputs,

$$\hat{u}(t) = (\dot{T}_e, \dot{B}_p, \dot{\delta}_w), \quad (21)$$

$$\hat{x}(t) = (X, Y, \psi, v_x, v_y, \dot{\psi}, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}, \Delta F_{zx}, \Delta F_{zy}, T_e, B_p, \delta_w). \quad (22)$$

Note that similarly it would also be possible to include more complex actuator dynamics, this could be especially interesting for steering angle and engine torque, but for computational reasons this simple integrator model is used.

3.1.5. Tyre forces

Tyre forces are modelled by means of Pacejka MF6.1 tyre model [56]. In this formulation, large camber angles and pressure changes are considered in addition to a better rolling resistance $M_{y_{as}}$ description. Overturning $M_{x_{as}}$ and aligning torques $M_{z_{as}}$ are neglected due

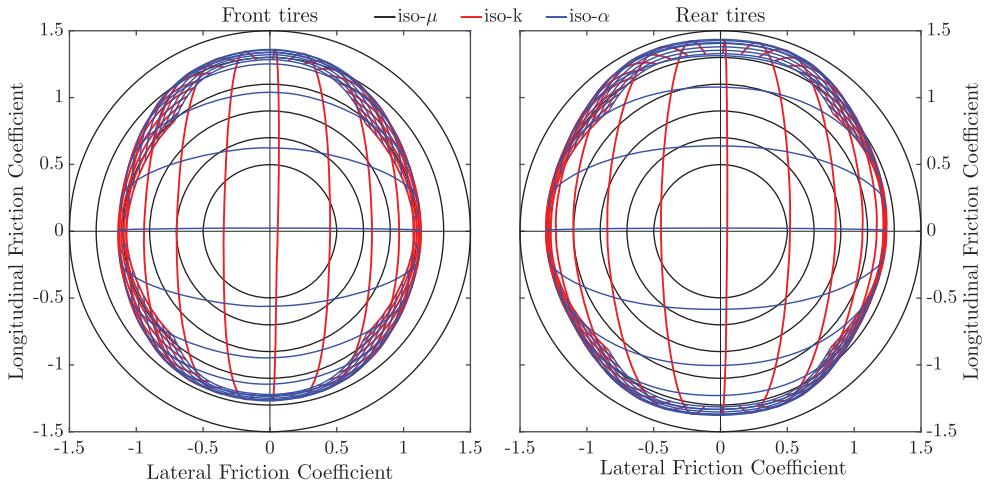


Figure 3. Tyre combined forces.

to the computational burden. Particularly the latter are important for vehicle handling purposes. However, the contribution which these give to the model isn't enough to justify the increase in computation time (approximately three times higher due to the need to calculate the Jacobian and Hessian for the optimiser). Concerning forces, the entire combined force formulation was implemented for both longitudinal and lateral behaviour. This is especially important since tyres represent the only component which exchanges forces with the ground. A classic slip ratio σ_{as} and slip angle α_{as} formulation is used [57] which considers velocity at the wheel centre. The resulting adherence ellipse for the front and rear tyres for a single vertical load is shown in Figure 3.

The exact tyre force can be seen in the aforementioned reference, however, tyre force results as a function of the following quantities:

$$F_{x_{as}} = f(\sigma_{as}, \alpha_{as}, F_{z_{as}}, v_x, v_y, \dot{\psi}), \quad F_{y_{as}} = g(\sigma_{as}, \alpha_{as}, F_{z_{as}}, v_x, v_y, \dot{\psi}). \quad (23)$$

3.2. State-space transformation

The above-described model can be written in state space form $\dot{\hat{x}}(t) = f(\hat{x}(t), \hat{u}(t))$. With the model written in this form, time is the independent variable. However, since the goal of the control system is to minimise how much time the vehicle takes to complete a lap, one popular approach is to transform the model from a time-dependent to a track-dependant (spatial) formulation. Additionally, this transformation allows for a natural formulation of obstacles and general road bounds under varying vehicle speed.

To obtain this transformation, the reference path C must be transformed by describing it with a curvilinear abscissa approach. Thus, it can be expressed as a function of its curvature k and the parametrisation of the curve by its arc-length s . With this approach, the cartesian position and angular coordinates X , Y , and ψ can be replaced by the longitudinal position on the reference line s , the lateral error with respect to it n and the heading angle error with respect to it α . A visual scheme of this transformation can be seen in Figure 4.

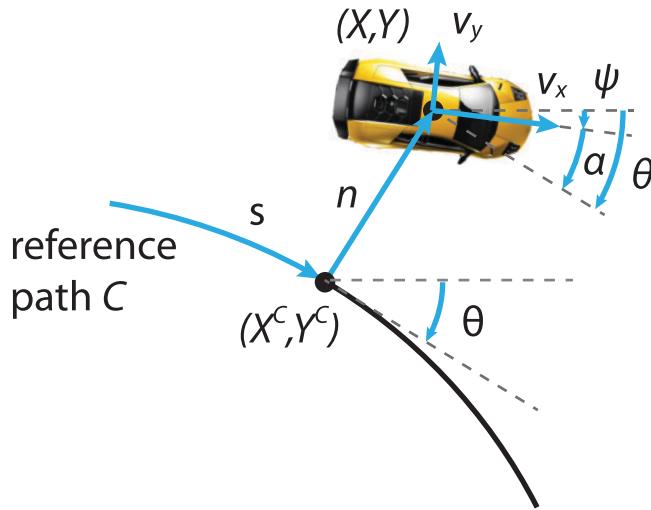


Figure 4. State space transformation.

The equation of motion of the vehicle's global position in the curvilinear abscissa formulation can be written as,

$$\dot{s} = \frac{v_x \cos(\alpha) - v_y \sin(\alpha)}{1 - nk(s)}, \quad (24a)$$

$$\dot{n} = v_x \sin(\alpha) + v_y \cos(\alpha), \quad (24b)$$

$$\dot{\alpha} = \dot{\psi} - \frac{v_x \cos(\alpha) - v_y \sin(\alpha)}{1 - nk(s)} k(s). \quad (24c)$$

The system of equations is valid under the assumption that the vehicle always stays at a lateral distance n that is smaller than the distance of the local centre of curvature of the road, i.e. $n < 1/k(s)$ and that the vehicle's total velocity is greater than zero $\|v_x + v_y\| > 0$. Based on this state transformation the system can now be transformed into the spatial form, where the independent variable is no longer time but s the position along the reference path,

$$\dot{\tilde{x}}(t) = \frac{d\tilde{x}}{dt} = \frac{d\tilde{x}}{ds} \frac{ds}{dt} = \dot{s}\tilde{x}'(s) \Rightarrow \tilde{x}'(s) = \frac{f(\tilde{x}(t), \hat{u}(t))}{\dot{s}} = \tilde{f}(\tilde{x}(s), \hat{u}(s)). \quad (25)$$

Where $\tilde{x}'(s)$ denotes the derivative of $\tilde{x}(s)$ with respect to s , and for the transformation to be well defined it needs to hold that $\dot{s} > 0$, which is the case if the absolute velocity is positive. Note that this transformation renders the s state redundant, thus the state $\tilde{x}(s)$ is given by,

$$\tilde{x}(s) = (n, \alpha, v_x, v_y, \dot{\psi}, \omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}, \Delta F_{zx}, \Delta F_{zy}, T_e, B_p, \delta_w). \quad (26)$$

3.3. Point-mass dynamics

The high-level controller is based on a simple point-mass model as shown in Figure 5.

The model represents a mass which position is always on the reference path, thus, the curvature is always known. Knowing the curvature at each point of the trajectory and the acceleration limits of the mass, it is possible to calculate a velocity profile.

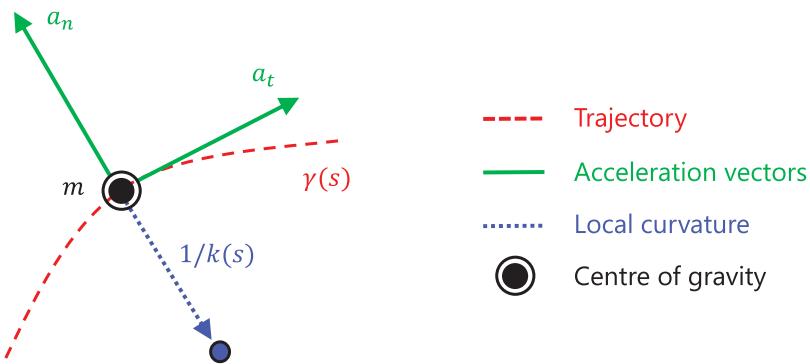


Figure 5. Point-mass dynamics model.

More precisely, the velocity of a point-mass along the reference path can be described by the following model,

$$\frac{dv_t}{dt} = a_t = \frac{f_t}{m} - \frac{\rho S C_x}{2m} v_t^2. \quad (27)$$

Where m , ρ , S , and C_x are as defined in Section 3.1, a_t is the tangential acceleration and f_t is the tangential force applied to the point-mass and, therefore, the input to the point-mass model. Furthermore, since the tangential velocity of the point-mass is known, the centripetal acceleration a_n is given by $a_n = v_t^2 k(s)$. Thus, both tangential force and centripetal acceleration are known which allows constraining the accelerations within the limits of the vehicle, mainly the tyres and engine limits which are discussed in the next section.

However, since the curvature is needed to compute the centripetal acceleration, it is beneficial to again transfer the system into the spatial form, where s is the independent variable and thus $k(s)$ is fix. Moreover, for the given point-mass dynamics where the lateral velocity v_y and deviation from the path n are zero, it can be seen that (24a) simplifies to $\dot{s} = v_t$. Thus, the state transformation is significantly simpler than in the general case described in the previous section. Additionally, similar to Section 3.1, an integrator actuator dynamics is introduced, mainly to be able to minimise (indirectly) the longitudinal jerk. Thus, the dynamics of the point-mass model in the spatial form are,

$$\begin{aligned} v_t(s)' &= \frac{1}{v_t} \left(\frac{f_t}{m} - \frac{\rho S C_x}{2m} v_t^2 \right), \\ f_t(s)' &= \frac{j_t}{v_t}. \end{aligned} \quad (28)$$

The dynamics can be written in the following state space model, $q(s)' = g(q(s), w(s))$, where $q = (v_t, f_t)$ is the state vector and $w = j_t$ is the input.

3.3.1. Acceleration limits

The acceleration limits of the vehicle can be represented with the g-g diagram. This represents the vehicle's performance limits measured at the centre of gravity under steady-state behaviour, hence can be used as a constraint for the point-mass model. Since it is important for the high-level controller to not underestimate the potential performance of the

vehicle, a slightly overoptimistic approach to compute the g-g limits based on tyre models is proposed. This is achieved by neglecting load transfer and thereby vertical force tyre sensitivity. Note that the acceleration limits here described are used as a constraint only for the point-mass model and high-level controller. Because of the structure of the hierarchical scheme, it is intended for the point-mass model to slightly overestimate vehicle performance when lateral dynamics are most important and short prediction horizons are sufficient to follow the path. In such conditions, the low-level controller can be free to make decisions (without any constraints getting in the way) and exploit vehicle performance. On the other hand, when the dynamics are dominated by longitudinal dynamics (i.e. braking), the high-level controller should correctly predict vehicle performance since the prediction horizon required is too long for the low-level controller, thus additional acceleration limits which include longitudinal weight transfer are included. Our method runs a sweep of slip angles and slip ratios at a fixed longitudinal velocity and calculates the correspondent longitudinal and lateral tyre forces. A matrix of tyre loads for every combined slip condition and each wheel is then obtained. From each tyre force matrix, alpha-shapes [58] are used to calculate the polygon which connects the external boundary of each tyre ellipse adherence, as shown on the left in Figure 6. The various tyre force boundaries are then summed and divided by the total vehicle mass to find the acceleration limits of the vehicle. At this point, another boundary polygon, which can be expected to be similar to an ellipse, is obtained, as shown on the right in Figure 6. This represents the acceleration limits of the vehicle. With this method, the g-g diagram is only dependant on tyre performance, which is the theoretical maximum performance that a vehicle can obtain. Finally, due to the need for an analytical formulation, a least squares approach is used to calculate the best fit ellipse. This ellipse equation can be used in the MPC scheme. However, the vertical load tyre sensitivity cannot be neglected for all effects, mainly when considering downforce and hard braking. Therefore, the vertical load caused by aerodynamic forces is considered since it can have

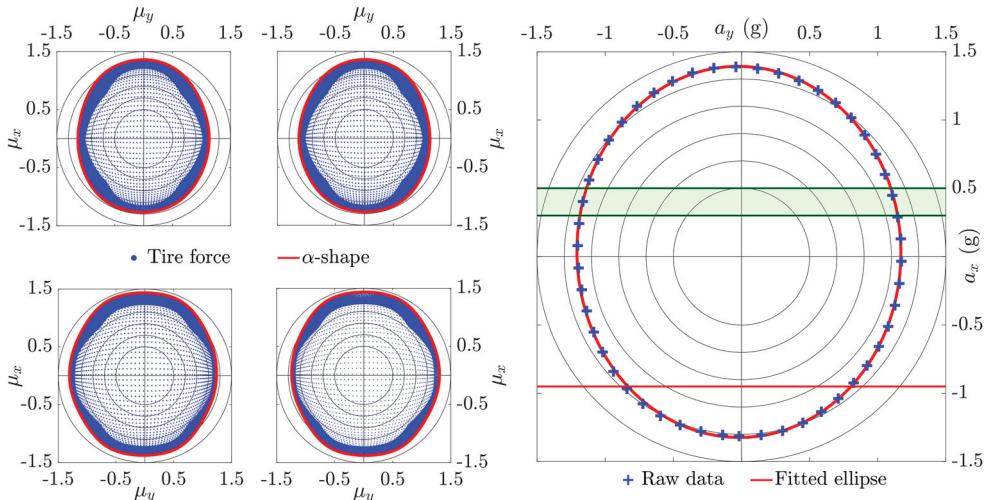


Figure 6. Acceleration limits. On the left the tyre forces and *alpha*-shapes are shown, on the right the resulting g-g diagram consisting of the tyre limits (raw and fitted shape).

a great effect on vehicles with high downforce. Furthermore, the change in downforce is slow and can be approximated as purely dependant on the longitudinal velocity of the car. Thus, the g-g diagram, including downforce effects, can be computed by solving the above described method for different longitudinal velocities and let the g-g diagram also depend on longitudinal velocity. In the case of braking, considering load tyre sensitivity is especially important due to the great load transfer obtained during braking manoeuvres and to the criticality of such manoeuvres; these effects have to be considered to obtain a feasible velocity profile for the lower level. However, compared to downforce it is not necessary to change the complete g-g diagram, but the limit can be considered by bounding the maximum longitudinal deceleration. This constraint is calculated with an iterative process. This consists of initialising the maximum longitudinal deceleration as the one calculated from the vehicle's adherence ellipse. From this value, longitudinal stationary weight transfer is calculated and, therefore, the new vertical tyre loads. With the new loads, it is then possible to calculate the new vehicle's maximum deceleration. This process is repeated until the difference between the starting vertical load and the one given by the load transfer caused by the braking effort is smaller than a threshold. In this work, 500N is used as a threshold. Note that due to the low downforce of the car considered these two effects can be considered separately, in case of a high-downforce car, the deceleration limit would also depend on the longitudinal velocity.

Finally, the longitudinal accelerations are limited by engine power, which does depend on longitudinal velocity and is considered by the power curve of the engine-gearbox unit. The resulting g-g diagram is mainly dominated by the ellipsoidal shaped tyre g-g limits as shown in Figure 6. This shape is cut with two bounds on the longitudinal acceleration, in negative direction caused by the tyre limits under braking (see red horizontal line in Figure 6) and positive direction by the engine power, which is additionally depending on velocity (see green shaded area in Figure 6). Thus at low velocities the car can accelerate more than at higher velocities. The resulting constraint is summarised in the set $\mathcal{A}(v_t)$ where v_t highlights that the constraint does depend on the longitudinal velocity. Note that other methods that compute the g-g diagram exist, some even include load transfer, but in this work a slightly overestimating g-g diagram is explicitly aimed for. With the procedure described in this section the acceleration limits are dependant on vehicle velocity. Specifically, the g-g diagram is calculated on a sweep of longitudinal velocities as is the longitudinal acceleration constraint since it is dependant on engine power. However, the LSD model is not included in this constraints as the acceleration limits constraint here presented are not used in the low-level control and, as already mentioned, purposely overestimate the vehicle performance especially when the dynamics are dominated by lateral dynamics.

4. Control scheme

In this section the proposed hierarchical control scheme will be described. In particular, first an overview of the control scheme will be discussed with details on how the high-level and low-level controllers interact with each other. Then, the high-level controller will be described in detail and its mathematical framework will be shown. Finally, the low-level controller will be described and details about its formulation will be given.

4.1. Hierarchical control scheme

The proposed hierarchical control scheme is designed to overcome one of the major issues with MPC controllers for autonomous driving applications. If the goal is to follow a reference path, as in the proposed controller, one has to use a long prediction horizon to adapt longitudinal velocity before curves to stay within the limits of handling in a racing setting and to guarantee passenger comfort or due to speeding limits in an AD setting. In the setting studied in this paper, where at-limit handling on a racing circuit is of interest, the horizon needs to be long since the car can drive over 200 km/h on straights, thus the horizon needs to be several hundred metres to correctly predict the braking point. On the other hand, one would like to use detailed vehicle models with fast actuation times to precisely predict the vehicle's motion, thus controlling the car also at at-limit handling. However, when combining a detailed vehicle model as discussed in Section 3.1 with a prediction horizon of several hundred metres, the NMPC problem takes at least several minutes to solve. Therefore, the problem is split into a two level hierarchical controller, where the high-level NMPC controller uses a long horizon and a simple model (point-mass model discussed in Section 3.3) to compute a velocity profile for the next 250 m. The low-level NMPC on the other hand uses a detailed vehicle model (vehicle model discussed in Section 3.1), and a short prediction horizon to render the problem real-time feasible. The velocity profile from the higher level controller is then used as a terminal velocity constraint in the lower level. This couples the long term velocity profile from the higher level controller to the short term precise decision making of the lower level NMPC. For example the terminal velocity allows the low-level NMPC to start braking at the correct braking point without even 'seeing' the next curve. The concept of the hierarchical controller is illustrated in Figure 7 together with a block diagram in Figure 8. Where the velocity profile found by the high-level controller is plotted on a typical slow curve after a straight line and the point along the trajectory at which this velocity is used as a terminal constraint for the low-level control is shown.

The advantage of this hierarchical approach is that the high-level problem, even with a prediction horizon of 250 m, can be solved within less than a millisecond on a thread of

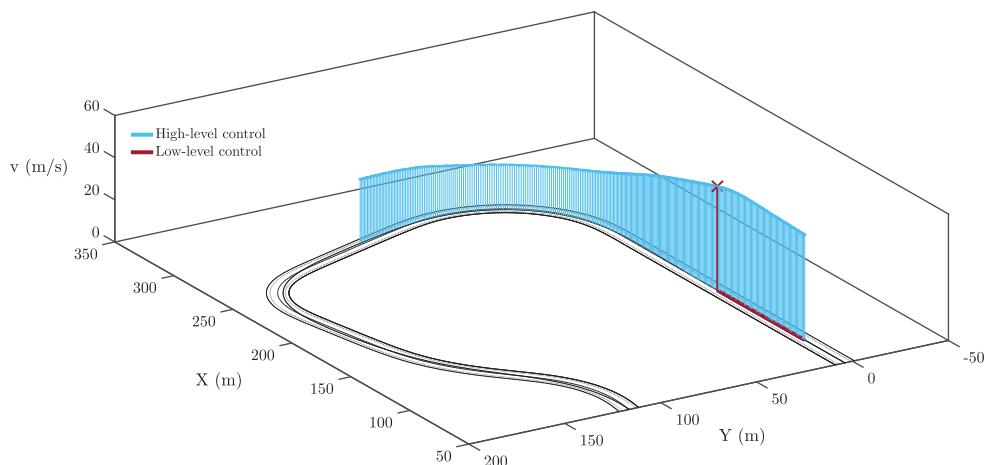


Figure 7. Hierarchical control scheme.

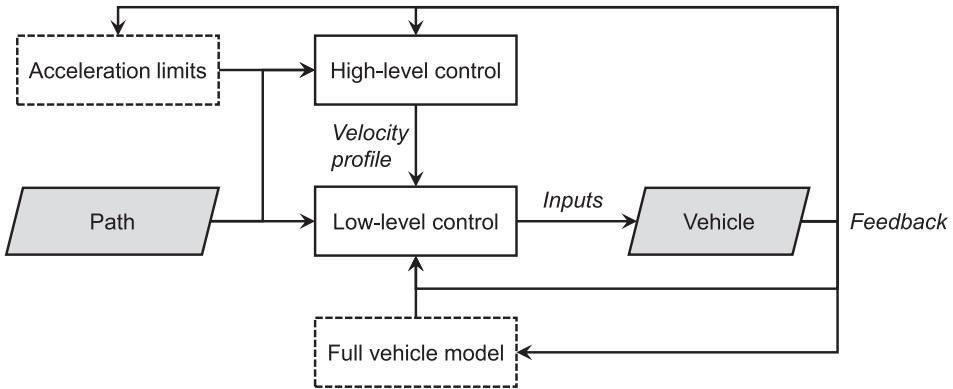


Figure 8. Hierarchical control scheme – block diagram.

a Intel i7-6700HQ CPU at 2.60 GHz. At the same time the horizon of the low-level problem can be reduced to below 30 m which allows solving the problem in approximately one hundred milliseconds, thus rendering the whole hierarchical controller real-time feasible. Finally, the higher level only determines a terminal constraint, thus the low-level NMPC is free to optimise its own trajectory, except the terminal velocity. This allows the vehicle really to act at at-limit handling and as a consequence, in certain areas of the circuit, the low-level controller achieves a higher speed profile than the generally optimistic high-level controller.

4.2. High-level controller

The high-level NMPC is designed to find the time optimal trajectory of a point-mass particle following a reference path. The model of the point-mass particle $g(q, w)$ is given by (29), where the state q is composed of tangential velocity v_t and tangential force f_t and the input w is the rate of change of tangential force j_t . Note that centripetal acceleration is given by $a_n = v_t^2 k(s)$. The accelerations of the point-mass particle are constrained to lay within $\mathcal{A}(v_t)$, which is described in Section 3.3.1. Where the acceleration limits come from tyre adherence limits, braking limits and limited engine power. The minimum time objective is straight forward to achieve since the model is formulated in the spatial form with distance as the independent variable. Thus the minimum time objective has the following transformation,

$$\min \int_{s_0}^{s_f} t \, d\xi = \min \int_{s_0}^{s_f} \frac{1}{\dot{s}(\xi)} \, d\xi, \quad (29)$$

where s_0 and s_f are the start and finish position, \dot{s} is defined in (24a) and ξ is the curvilinear abscissa integration variable. Since n , α , and v_y are zero for the point-mass model $\dot{s} = v_t$. Thus $1/\dot{s}(\xi)$ further simplifies to $1/v_t(\xi)$. Finally, the dynamics and the cost function are discretised to transform the problem into a format such that an NLP solver can solve the problem. The dynamics are discretised using a fourth-order Runge–Kutta method, whereas the cost integral is approximated with a Riemann integral. Note that this approach is often

referred to a multiple shooting.

$$\begin{aligned}
\min_{\mathbf{q}, \mathbf{w}} \quad & \sum_0^{N_h} \frac{1}{v_{t,k}} + w_k^T R w_k \\
\text{s.t.} \quad & q_0 = q(0) \\
& q_{k+1} = g_d(q_k, w_k) \\
& \left[\frac{f_{t,k}}{m}, v_{t,k}^2 k(s_k) \right] \in \mathcal{A}(v_t) \\
& v_{t,k} > 0 \\
& w_k \in \mathcal{W}.
\end{aligned} \tag{30}$$

Where $\mathbf{q} = [q_0, \dots, q_{N_h}]$ is the state trajectory and $\mathbf{w} = [w_0, \dots, w_{N_h-1}]$ the input trajectory. N_h is the horizon length of the high-level controller. Furthermore, is $q(0)$ the measured state, $g_d(q_k, w_k)$ the discretised version of the dynamics, $v_{t,k} > 0$ is imposed to ensure that the spatial transformation is well defined, and \mathcal{W} represents upper and lower bounds on the input w .

4.3. Low-level controller

Similar to the high-level controller the goal of the low-level controller is also to follow a path in a time optimal way. However, compared to the high-level controller, the model consists of a seven degrees of freedom model, a full MF6.1 tyre model, a LSD differential and tyre vertical load variations as described in Section 3.1. Furthermore, the car can deviate from the reference path, thus constraints on lateral distance to the path are introduced to guarantee that the car stays inside that track. Finally, the terminal velocity constraint is introduced based on the optimal solution of the high-level controller. Thus the discrete time MPC problem can be formulated as follows,

$$\begin{aligned}
\min_{\tilde{\mathbf{x}}, \hat{\mathbf{u}}} \quad & \sum_{k=0}^{N_l} j_{\text{time},k} + j_{\text{dyn},k} + j_{u,k} + j_{\hat{u},k} + j_{\text{slack},k} \\
\text{s.t.} \quad & \tilde{x}_0 = \tilde{x}(0) \\
& \tilde{x}_{k+1} = \tilde{f}_d(\tilde{x}_k, \bar{u}_k) \\
& n_k \geq \underline{n}_k - \rho_k, \quad n_k \leq \bar{n}_k + \rho_k, \quad \rho \geq 0 \\
& \tilde{x}_k \in \tilde{\mathcal{X}} \\
& \hat{u}_k \in \hat{\mathcal{U}} \\
& v_{x,N_l} \leq v_{x,T}.
\end{aligned} \tag{31}$$

Where $\tilde{\mathbf{x}} = [\tilde{x}_0, \dots, \tilde{x}_{N_l}]$ is the state trajectory and $\hat{\mathbf{u}} = [\hat{u}_0, \dots, \hat{u}_{N_l-1}]$ the input trajectory. N_l is the horizon length of the low-level controller. The cost is built up of several components, first the time minimisation $j_{\text{time},k} = 1/\dot{s}_k$, where \dot{s} is defined in (24a). Second, regularisation costs related to the dynamics of the car, penalising deviations from the path, angle

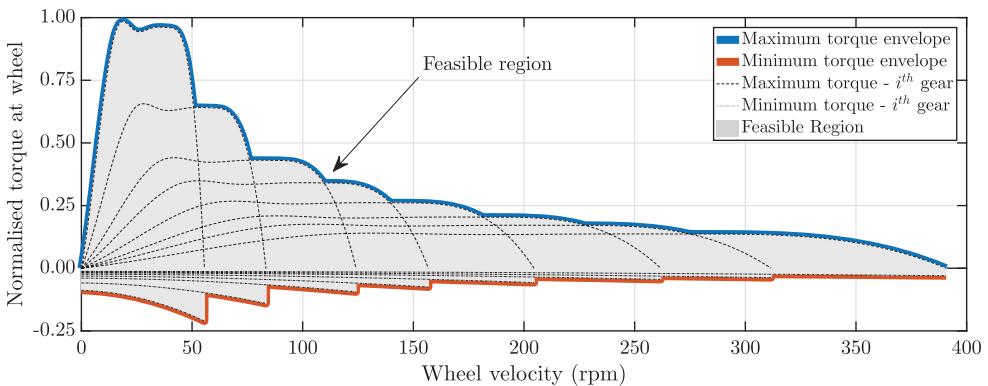


Figure 9. Torque constraints.

deviations from the path and side slip angles, $j_{\text{dyn},k} = d_n n_k^2 + d_\alpha \alpha_k^2 + d_\beta (\tan^{-1}(v_y/v_x))^2$, where d_n , d_α , and d_β are weights. Third, $j_{u,k} = u_k^T R u_k$ and $j_{\hat{u},k} = \hat{u}_k^T R_\Delta \hat{u}_k$ penalise deviations from zero in the real inputs as well as the input rates, with R and R_Δ being positive definite weight matrices. Finally, $j_{\text{slack},k} = \gamma \rho^2$ is the penalisation of the slack multipliers related to the soft constraint implementation of the track boundary constraints. Furthermore, $\tilde{x}(0)$ is the measured state, $\tilde{f}_d(\tilde{x}_k, \bar{u}_k)$ the discretised vehicle dynamics described in Section 3.1, where an implicit Runge-Kutta method is used to integrate the dynamics. The border constraints are implemented as bounds \underline{n} and \bar{n} on the orthogonal deviation from the path n , to avoid feasibility issues the constraints are implemented as soft constraints. $\tilde{\mathcal{X}}$ and $\hat{\mathcal{U}}$ represent bounds on all the states and inputs, explicitly on the real inputs but also on the derivatives of the inputs. Note that the bounds on the drivetrain torque does change depending on wheel speed of the rear axle, which approximates the changing torque bounds due to the engine speed and gear. A figure of the wheel torque envelopes considering each each gear is shown in Figure 9 with the feasible region considered by the controller. Note that also the maximum and minimum torque for each gear are illustrated.

This approach assumes that the gear in which achieves the highest torques is always selected, which is a reasonable assumption for automatic racing and helps to avoid integer variables due to the gear. Finally, the longitudinal velocity at the end of the horizon is limited by $v_{x,T}$ the terminal constraint form the higher level. Note that the low-level controller is not allowed to drive faster than the terminal velocity constraint only at the end of the horizon while the minimum time objective is concerned about achieving the best possible velocity. In the above description it was mentioned that an implicit integrator is used, this is necessary since the wheel dynamic equations are stiff, especially under braking for low speed. To further increase the robustness with respect to integration errors the discretisation distance is chosen to be adaptive. In the spatial formulation the discretisation distance is fixed, thus when the car is driving slower, the ‘discretisation time’ is getting longer. Hence, integration errors are getting larger exactly when the dynamics are most challenging. To counter act this effect, the discretisation distance is adapted based on the minimal velocity of the last optimal solution. This results in a behaviour similar to discrete time systems where the horizon length in distance depends on the velocity. Additionally to the robustness of the solution, with the proposed method the horizon grows when the velocity is higher and is shorter when the velocity is lower. This is closer to what a human does since

for low velocities a very long horizon is not required. With this method, the advantages of both space and time formulations are maintained and the algorithm is proved to be very robust to critical situations such as wheel locking.

5. Results

In this section the results of the proposed algorithm will be shown and discussed. The simulations are run on a previously validated Vi-Grade model. The controller has been implemented in MATLAB on an off-the-shelf laptop with an Intel i7-6700HQ CPU at 2.60 GHz. To solve the NMPC problems the FORCES Pro NLP solver is used [59], which is an NLP solver tailored for MPC problems exploiting the given sparsity pattern, more information about the solver can be found in the paper of Zanelli et al. [60]. The low-level controller solves the optimisation at a mean frequency of 10Hz and feeds a new input (input rate) at the same frequency. The real input is given at a frequency of 100Hz considering constant rate. The high-level control runs at a mean frequency of 200 Hz, however, the input is fed to the model at the same frequency as the lower level (10 Hz for the rate and 100 Hz for the real input). The vehicle model runs at 1kHz. The horizon length of the high-level controller is $N_h = 250$ with a discretisation distance of 1m which results in a prediction horizon of 250 m while the low-level has a horizon length of $N_l = 30$ with a discretisation distance of 1 m at 30 m/s which results in a prediction horizon of 30 m.

The simulations that will be shown in this section are those of a sport saloon driving around Hockenheim circuit. First the outputs of the high-level controller will be shown. Finally, the results of the low-level controller of the hierarchical scheme will be discussed. Note that the two controllers are tied together due to the feedback that the high-level control gets from the low-level control and the terminal constraint that the high-level feeds to the low-level. However, for clarity of exposure, the results will be initially analysed separately. The results of the high-level control consider only the values corresponding to the terminal curvilinear abscissa of the low-level control.

5.1. Trajectory

In the previous sections a known reference path is assumed, in the introduction several methods to generate a ideal path to minimise lap time were discussed. However, in this paper a rather simple approach to find a good reference path is used by applying the minimum curvature trajectory approach of Braghin et al. [61]. The authors refer to this paper for the complete formulation. Furthermore, it is important to point out that the exact path is not so much of interest since the goal is to design a control scheme capable of optimising the vehicle performance on any given path.

The circuit used in this paper is Hockenheimring in its short configuration. In Figure 10 the results of the minimum curvature algorithm on this circuit are shown.

5.2. High-level controller results

The goal of the high-level controller is to supply the terminal velocity constraint to the low-level. It is important to not underestimate vehicle's performance. As mentioned in the

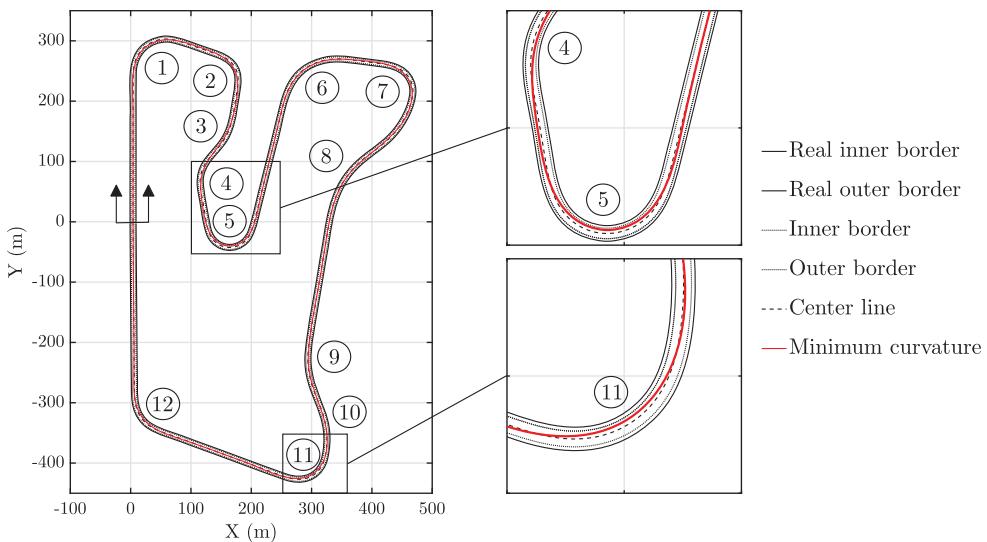


Figure 10. Minimum curvature trajectory.

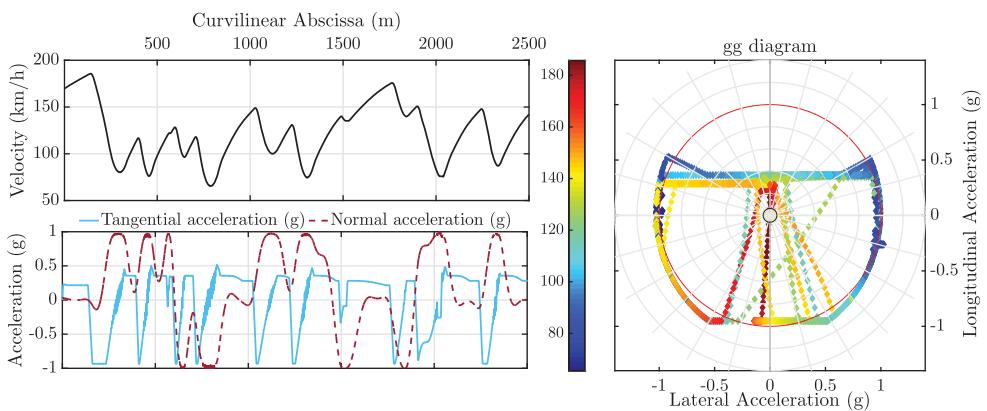


Figure 11. High level controller velocity profile.

previous section, vehicle's lateral behaviour can easily be predicted by the low-level controller even with a short horizon since the track's curvature is large and smooth. On a straight line instead, it is the high-level controller that has to dictate the terminal velocity and, more importantly, the braking point. In fact, to correctly predict when to start braking it is important to have very long horizons. The velocity profile that the high-level controller calculates along the horizon should have abrupt changes in acceleration (e.g. hard braking) since this way the low-level controller will have a realistic braking profile as a reference.

Figure 11 shows the velocity profile calculated by the high-level controller by running the high-level controller in a receding horizon fashion with the point-mass model as a simulation model.

The top figure shows the velocity profile while the lower figure shows both the tangential and normal acceleration profiles. The velocity profile is very smooth and shows the typical

high gradients given by a braking manoeuvre executed by a human. Additionally, the longitudinal velocity has a quadratic profile due to aerodynamic forces. With regards to the acceleration profiles, both normal and tangential accelerations tend to the g-g constraints given to the controller. Since positive longitudinal acceleration is bounded with velocity, maximum longitudinal acceleration varies along the circuit. The oscillatory profiles of the accelerations are due to the states being in a small region near the bounds, leading the solver to oscillate around a feasible region. Since acceleration constraints are modelled as hard constraints, when the point-mass finds itself near to the bounds, the solver tends to push the vehicle to the limit while maintaining feasible results, resulting in an oscillatory profile. However, as shown in the velocity profile, these high frequency oscillations are damped in the velocity domain. Therefore, it can be concluded that these high frequency fluctuations do not influence the purpose of the controller, which is to give a terminal constraint to the lower level. The resulting g-g diagram obtained by the high-level controller is also shown in Figure 11.

Note that the high-level controller acts at the limits of the theoretical g-g diagram (acceleration constraints). In fact, in the acceleration space, the resulting acceleration profile generates a shape which is similar to the adherence ellipse shown in Figure 6. Additionally, the maximum negative deceleration (braking) is saturated to the value used as a constraint and found with the previously described iterative process (independent of velocity). On the other hand, the maximum positive acceleration tends to follow different horizontal lines at different speeds. This is a consequence of a velocity dependant torque constraint in the controller. For low speeds and high lateral accelerations, which correspond to low speed corners exits, the point-mass' acceleration limits are given by the tyre constraints rather than torque constraints. For this reason the maximum longitudinal acceleration at high lateral acceleration and low velocity is higher than in other areas of the state space.

5.3. Low-level controller results

The lower level controller is the one responsible for the real performance of the vehicle. Assuming that the higher level didn't underestimate the acceleration limits, the terminal constraint feeds the braking point to the low-level. Thus, it is interesting to analyse the difference in the velocity profiles obtained by the low-level controller and the high-level controller. Note that since the high-level receives its initial condition as a feedback from the low-level, the velocity in the prediction horizon of the low-level is illustrated. The low-level has a short horizon and solves the optimisation problem slowly while the high-level has a long horizon and solves the problem with high frequencies. The comparison of the velocity profiles can be seen in Figure 12.

Note that the high-level controller results are based on point-mass dynamics on a given trajectory, thus, absolute velocity is equal to tangential velocity. On the other hand, the low-level controller is based on rigid body dynamics, thus, absolute velocity comprises also lateral velocity. However, in this plot longitudinal velocity only is shown since it's the main contribution in terms of velocity for low sideslip angles.

It can be seen how terminal velocity and real velocity are very similar but, as expected, the real velocity is lower than the terminal constraint almost everywhere. However, for certain values of curvilinear abscissa, the real velocity is higher than the terminal one. This result shows one of the advantages of using only a terminal set for the controller instead

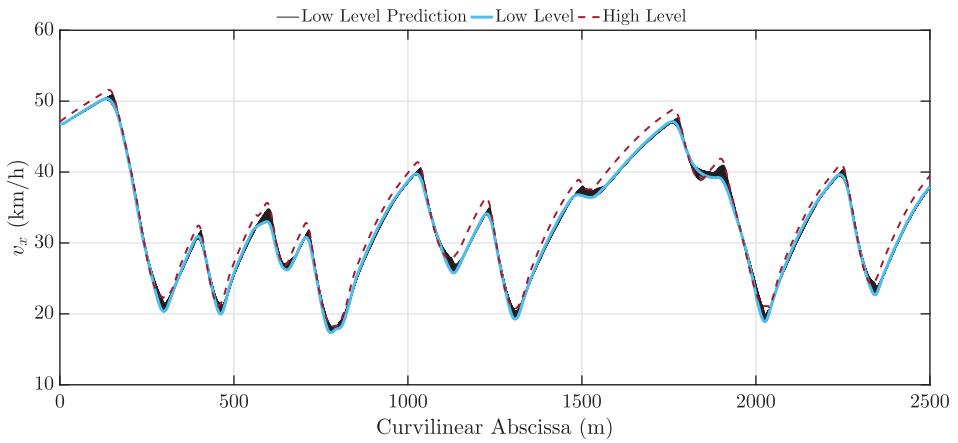


Figure 12. Velocity profiles comparison.

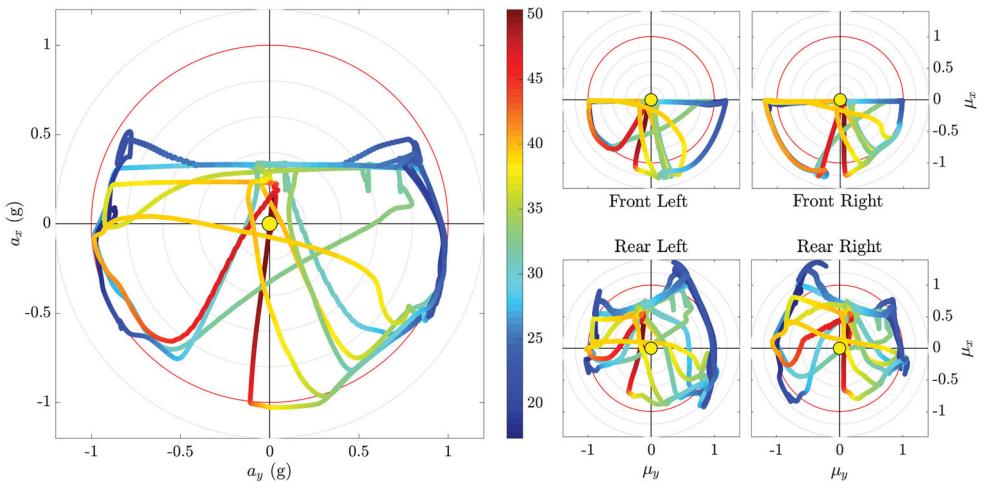


Figure 13. Low level controller g-g diagram.

of the profile along the entire horizon. Additionally, the velocity profile generated by the high-level controller is far from realistic given the model's simplification. Note that in the abscissa plots the proposed algorithm proves to be effective since the real velocity follows the terminal velocity where abrupt braking is required. However, when the vehicle is performing a curve, real velocity is much lower than the terminal constraint. In this condition, the low-level control, as expected, performs less than the high-level control due to lateral load transfer and vertical tyre sensitivity. The need of a weight transfer formulation in the low-level NMPC model is highlighted here. Longitudinal positive accelerations are very similar for both controllers, as can be seen from the gradients of the velocity profiles. To fully evaluate the performance of the controller, the g-g diagram is shown in Figure 13.

In this figure both the acceleration diagram and the adherence ellipse of the four tyres are shown. Note that these results are coming from a commercial software vehicle model that is controlled by the proposed scheme. Analysing the acceleration diagram, it can be

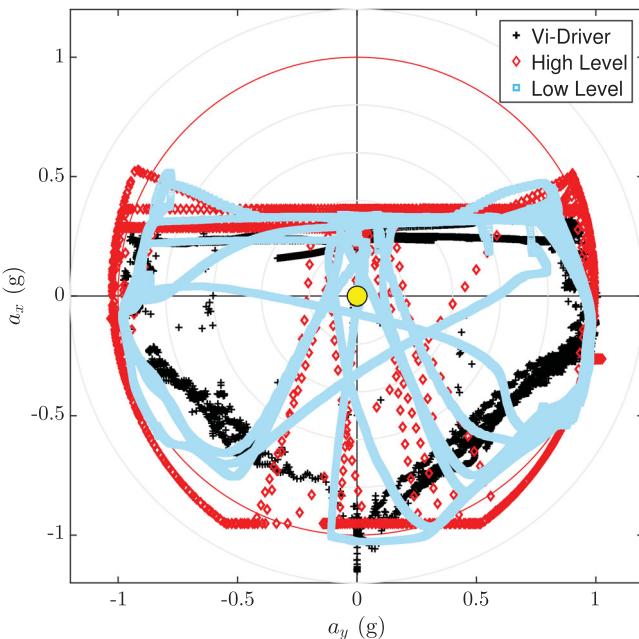


Figure 14. Comparison of g-g diagram.

noted how the low-level controller has a very similar profile to the acceleration diagram of the point-mass model. The acceleration diagram shows how the controller is able to exploit the combined behaviour of the vehicle where normally a lot of performance is lost. In these conditions the tyres are operating in their nonlinear range and every little variation can make the vehicle lose control or performance. On the other hand, a robotic controller based on a predictive control with a high fidelity model can maximise the vehicle's performance. In the positive longitudinal behaviour, it can be observed how for low speed and high lateral acceleration, once again, the peak acceleration is given by the engine torque rather than the tyres. Finally, since the analysed vehicle has negative downforce (as most commercial vehicles), the peak lateral acceleration decreases with speed. Positive lateral accelerations in this circuit are more frequent due to the circuit orientation.

In this figure the tyre forces are also shown. The acceleration diagram is a direct result of these forces as previously explained. The advantages of considering a combined tyre formulation, weight transfer and, more importantly, LSD formulation can be seen in the rear tyre force plots where the maximum force is obtained in combined slip conditions. Contrarily, the front tyres show a more symmetric behaviour.

The results of the proposed controller, in terms of acceleration diagrams, can be compared to Vi-Driver, the standard driver used in commercial software Vi-Grade. This driver is not a real-time driver and relies on backstepping when a simulation does not converge. This is very advantageous for Vi-Driver since it does not have real-time constraints in the solver. However, it is interesting to see how the results compare. The comparison is showed in Figure 14.

Both the high-level controller and the low-level controller outperform the commercial driver. While for the high-level this was expected, the same cannot be said for the low-level.

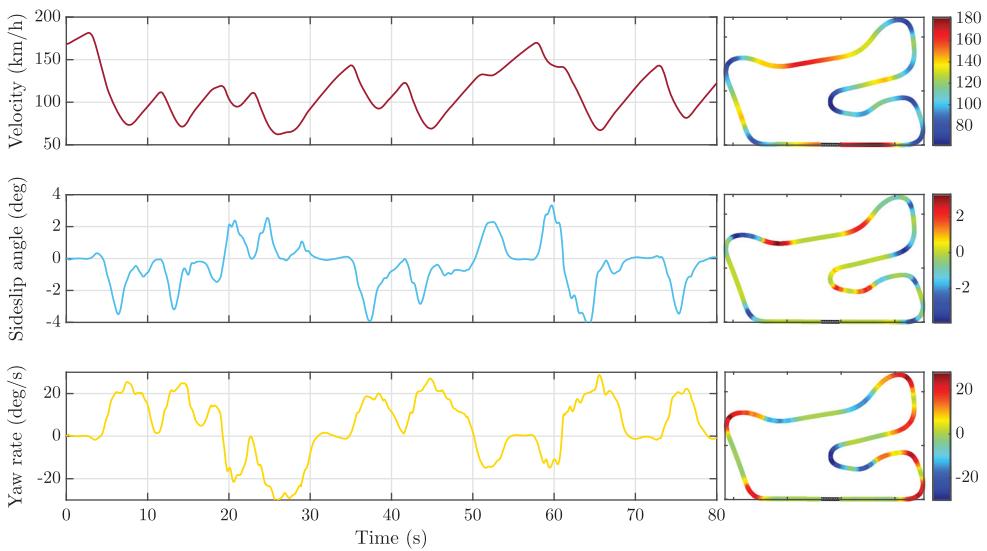


Figure 15. Velocity, sideslip angle and yaw rate of the proposed algorithm.

Note that the same vehicle, trajectory and solver for the vehicle model are used in all simulations. The combined behaviour of the proposed algorithm is where the performance is most improved, showing the importance of considering the complete MF model, comprehensive of combined formulation. Also, the low-level controller performs very similarly to the higher level in terms of combined behaviour except for hard braking. This is due to the fact that under hard braking, wheel dynamics become critical for the solver since wheel under rotation can easily occur. Overall, it can be concluded that the proposed method shows promising results when compared to state of the art controllers.

Besides longitudinal velocity and accelerations, it is important to analyse other significant states of the vehicle, such as sideslip angle and yaw rate. In Figure 15 both of these quantities are plotted together with longitudinal velocity. These are plotted versus time and with a colourmap along the circuit.

Sideslip angle gives an idea of the lateral slipping of the vehicle. Since tyre slip angle depends on yaw rate and sideslip angle, smaller sideslip angle implicates a greater possibility to yaw. However, due to nonlinear behaviour of the dynamics and to the non-neutral behaviour of the vehicle, the optimiser could converge to a solution where drifting around a curve is optimal for time. This would make the vehicle unstable and would not be a realistic driving scenario. With the proposed controller, a cost term in the objective function of the low-level controller ensures a bounded sideslip angle, enhancing stability [62], while at the same time allowing the vehicle to slightly drift when required. The maximum absolute value of sideslip angle obtained around the circuit corresponds in this case to the maximum open-loop sideslip angle of the vehicle. The highest values of sideslip angle occurs when an inversion of yaw rate sign is required since, in such conditions, a pendulum effect is generated and the front and rear slip angles are in counter phase so a great yaw moment is produced. Note that the vehicle analysed is characterised by a positive stability margin [63]. Finally, considering yaw rate, the absolute values obtained are not excessive since the

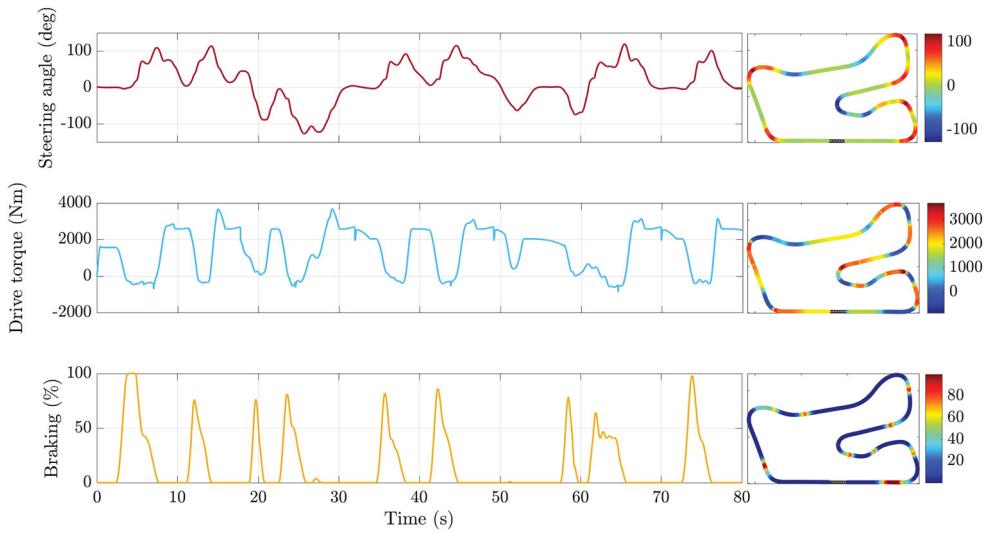


Figure 16. Inputs the proposed algorithm.

vehicle is stable. The important thing to note is that the controller is capable of inverting yaw rate sign without losing control and performing agile manoeuvres.

Finally, the vehicle's inputs are analysed, namely steering angle, drive torque and braking percentage. As previously described, the controller calculates rates of these quantities. Since the solver calculates a solution every 10 Hz, this guarantees a smooth input sequence. In fact, an input rate is calculated every 10 Hz, however, the inputs are fed to the Vi-Grade model at 1 kHz. Thus, during each optimisation time window, one hundred different inputs are fed. This is done with an interpolation in the optimal input sequence which the NLP solvers returns (input rate). Furthermore, due to the used input dynamics the real inputs have first order hold characteristic, with the gradient being the inputs defined in (23). The results of the optimisation can be seen in Figure 16 where, for the reasons explained, the input sequence is very smooth. Note that in the drive torque curve, there are various peaks given by gear changes. The cost in the objective function on steering angle was kept very high to guarantee a smooth steering action, however, as shown in the figure, high frequency corrections still occur when necessary. The most critical input to calculate for the optimiser is brake percentage. This is due to the already mentioned stiff wheel dynamics. For this reason, only seldomly does brake percentage reach its maximum value.

6. Conclusion

In this paper a hierarchical control scheme is proposed to exploit an autonomous vehicle's performance. The proposed robotic controller is composed of two levels; a high-level NMPC based on a point-mass particle travelling along the reference path whose constraints are given by a tyre based g-g diagram, a maximum torque transmittable via the powertrain and a maximum braking force. Due to the simplified dynamics, this controller is capable of finding a feasible solution in real-time with long horizons. The purpose of this level of the controller is to calculate a velocity profile that can correctly predict the braking point

for the lower level. In fact, the low-level controller is an NMPC based on a seven degrees freedom vehicle model with vertical dynamics, LSD model and with Pacejka MF6.1 tyre model including combined force effects. Due to the complexity of the model, the controller is capable of taking the vehicle to its limits while solving the optimisation problem in real-time only with a very short prediction horizon. However, this level of the controller uses the velocity profile predicted by the higher level to obtain a terminal velocity constraint which then governs the braking of the vehicle. This makes the controller robust and capable of completing a lap of a circuit in a time optimal manner. Both controllers are transformed into space dependant dynamics.

The proposed algorithm shows how two of the main issues that can be found in the pre-existing control algorithms in literature can be overcome with a hierarchical type scheme as the one presented here. Firstly, the vehicle dynamics at at-limit handling can be correctly predicted thanks to a complex vehicle model, thus exploiting vehicle performance. Second, solving a time optimal autonomous driving problem in real-time by separating the task in long and short term decisions and couple them efficiently using constraints.

Finally, the proposed control scheme can be applied to any trajectory since the terminal constraint is computed in a receding horizon manner. This also allows to use the method in the less controlled AD environment on public roads.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

- [1] Allen RW, Myers TT, Rosenthal TJ. The effect of tire characteristics on vehicle handling and stability. SAE Technical Paper; 2000.
- [2] Fujioka T, Kimura T. Numerical simulation of minimum-time cornering behavior. JSAE Rev. 1992;13(1):44–51.
- [3] Allen RW, Rosenthal TJ, Klyde DH, et al. Computer simulation analysis of light vehicle lateral/directional dynamic stability. SAE Trans. 1999;108:482–497.
- [4] Clover CL, Bernard JE. The influence of lateral load transfer distribution on directional response. Warrendale (PA): SAE International; 1993. (SAE Technical Paper 930763).
- [5] Voser C, Hindiyeh RY, Gerdes JC. Analysis and control of high sideslip manoeuvres. Veh Syst Dyn. 2010 Dec;48(sup1):317–336.
- [6] Durali M, Javid GA, Kasaiezadeh A. Collision avoidance maneuver for an autonomous vehicle. In: 9th IEEE International Workshop on Advanced Motion Control, 2006 Mar; 2006. p. 249–254.
- [7] Adminaita D, Calinescu T, Jost G, et al. Ranking EU progress on road safety. European Transport Safety Council; 2018.
- [8] Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE International; 2018.
- [9] Milliken WF, Milliken DL. Race car vehicle dynamics. Warrendale (PA): SAE International; 1995.
- [10] Brayshaw DL, Harrison MF. A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location. Proc Inst Mech Eng Part D J Automob Eng. 2005;219(6):725–739.
- [11] Tremlett A, Assadian F, Purdy D, et al. Quasi-steady-state linearisation of the racing vehicle acceleration envelope: a limited slip differential example. Veh Syst Dyn. 2014;52(11):1416–1442.

- [12] Casanova D, Sharp R, Symonds P. Minimum time manoeuvring: the significance of yaw inertia. *Veh Syst Dyn.* **2000**;34(2):77–115.
- [13] Kelly DP, Sharp RS. Time-optimal control of the race car: a numerical method to emulate the ideal driver. *Veh Syst Dyn.* **2010**;48(12):1461–1474.
- [14] Kelly DP, Sharp RS. Time-optimal control of the race car: influence of a thermodynamic tyre model. *Veh Syst Dyn.* **2012**;50(4):641–662.
- [15] Tremlett AJ, Limebeer DJN. Optimal tyre usage for a formula one car. *Veh Syst Dyn.* **2016**;54(10):1448–1473.
- [16] Perantoni G, Limebeer DJ. Optimal control for a formula one car with variable parameters. *Veh Syst Dyn.* **2014**;52(5):653–678.
- [17] Velenis E, Tsotras P. Minimum time vs maximum exit velocity path optimization during cornering. In: 2005 IEEE international symposium on industrial electronics; 2005. p. 355–360.
- [18] Tavernini D, Velenis E, Lot R, et al. On the optimality of handbrake cornering. In: IEEE 52nd Annual Conference on Decision and Control (CDC), 2013. IEEE; 2013. p. 2233–2238.
- [19] Tavernini D, Massaro M, Velenis E, et al. Minimum time cornering: the effect of road surface and car transmission layout. *Veh Syst Dyn.* **2013**;51(10):1533–1547.
- [20] Tremlett A, Massaro M, Purdy D, et al. Optimal control of motorsport differentials. *Veh Syst Dyn.* **2015**;53(12):1772–1794.
- [21] Rucco A, Notarstefano G, Hauser J. Computing minimum lap-time trajectories for a single-track car with load transfer. In: 2012 IEEE 51st annual conference on decision and control (CDC). IEEE; 2012. p. 6321–6326.
- [22] Rucco A, Notarstefano G, Hauser J. An efficient minimum-time trajectory generation strategy for two-track car vehicles. *IEEE Trans Control Syst Technol.* **2015**;23(4):1505–1519.
- [23] Lot R, Dal Bianco N. Lap time optimisation of a racing go-kart. *Veh Syst Dyn.* **2016**;54(2):210–230.
- [24] Dal Bianco N, Roberto L, Marco G. Minimum time optimal control simulation of a GP2 race car. *Proc Inst Mech Eng Part D J Automob Eng.* **2017**.
- [25] Velenis E, Tsotras P. Optimal velocity profile generation for given acceleration limits: theoretical analysis. *System.* **2005**;2:5.
- [26] Velenis E, Tsotras P. Optimal velocity profile generation for given acceleration limits; the half-car model case. In: 2005 IEEE International Symposium on Industrial Electronics; 2005. p. 355–360.
- [27] Velenis E, Tsotras P. Minimum-time travel for a vehicle with acceleration limits: theoretical analysis and receding-horizon implementation. *J Optim Theory Appl.* **2008**;138(2):275–296.
- [28] Velenis E, Tsotras P. Optimal velocity profile generation for given acceleration limits; Receding horizon implementation. In: Proceedings of the 2005 American Control Conference, 2005. IEEE; 2005. p. 2147–2152.
- [29] Zhang J, Xu S, Rachid A. Sliding mode lateral motion control for automatic steering of vehicles. In: Control Conference (ECC), 2001 European; Sep.; 2001. p. 3636–3641.
- [30] Klomp M, Olsson K, Sandberg C. Non-linear steering control for limit handling conditions using preview path curvature. *Int J Veh Auton Syst.* **2014**;12(3):266–283.
- [31] Kritayakirana K, Gerdes JC. Autonomous cornering at the limits: maximizing a g-g diagram by using feedforward trail-braking and throttle-on-exit. *IFAC Proc Vol.* **2010**;43(7):548–553.
- [32] Kritayakirana K, Gerdes JC. Autonomous vehicle control at the limits of handling. *Int J Veh Auton Syst.* **2012**;10(4):271–296.
- [33] Guiggiani M. The science of vehicle dynamics: handling, braking, and ride of road and race cars. Dordrecht: Springer; 2014; oCLC: 873455612.
- [34] Kritayakirana K, Gerdes JC. Using the centre of percussion to design a steering controller for an autonomous race car. *Veh Syst Dyn.* **2012**;50(sup1):33–51.
- [35] Prokop G. Modeling human vehicle driving by model predictive online optimization. *Veh Syst Dyn.* **2001**;35(1):19–53.
- [36] Borrelli F, Falcone P, Keviczky T, et al. MPC-based approach to active steering for autonomous vehicle systems. *Int J Veh Auton Syst.* **2005**;3(2–4):265–291.

- [37] Keviczky T, Falcone P, Borrelli F, et al. Predictive control approach to autonomous vehicle steering. In: American Control Conference, 2006. IEEE; 2006. p. 6–pp.
- [38] Kraus T, Ferreau H, Kayacan E, et al. Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. *Comput Electron Agric*. **2013**;98:25–33.
- [39] Cui Q, Ding R, Zhou B, et al. Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering. *Proc Inst Mech Eng Part D J Automob Eng*. **2017**.
- [40] Falcone P, Tufo M, Borrelli F, et al. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In: 2007 46th IEEE Conference on Decision and Control. IEEE; 2007. p. 2980–2985.
- [41] Timings JP, Cole DJ. Minimum manoeuvre time of a nonlinear vehicle at constant forward speed using convex optimisation. In: Proceedings of The 10th International Symposium on Advanced Vehicle Control, Loughborough, UK; 2010.
- [42] Timings JP, Cole DJ. Efficient minimum manoeuvre time optimisation of an oversteering vehicle at constant forward speed. In: American Control Conference (ACC), 2011. IEEE; 2011. p. 5267–5272.
- [43] Liniger A, Lygeros J. Real-time control for autonomous racing based on viability theory. *IEEE Trans Control Syst Technol*. **2019 Mar**;27(2):464–478.
- [44] Liniger A, Lygeros J. A non-cooperative game approach to autonomous racing. arXiv preprint arXiv:171203913; 2017.
- [45] Rosolia U, Borrelli F. Learning model predictive control for iterative tasks. arXiv preprint arXiv:160901387; 2016.
- [46] Rosolia U, Carvalho A, Borrelli F. Autonomous racing using learning model predictive control. In: American Control Conference (ACC), 2017. IEEE; 2017. p. 5115–5120.
- [47] Liniger A, Domahidi A, Morari M. Optimization-based autonomous racing of 1:43 scale RC cars: optimization-based autonomous racing. *Optim Control Appl Methods*. **2015**;36(5):628–647.
- [48] Gao Y, Gray A, Frasch JV, et al. Spatial predictive control for agile semi-autonomous ground vehicles. In: Proceedings of the 11th international symposium on advanced vehicle control; 2012.
- [49] Frasch JV, Gray A, Zanon M, et al. An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. In: 2013 European Control Conference (ECC). IEEE; 2013. p. 4136–4141.
- [50] Verschueren R, De Bruyne S, Zanon M, et al. Towards time-optimal race car driving using nonlinear MPC in real-time. In: 2014 IEEE 53rd annual conference on decision and control (CDC). IEEE; 2014. p. 2505–2510.
- [51] Verschueren R, Zanon M, Quirynen R, et al. Time-optimal race car driving using an online exact hessian based nonlinear MPC algorithm. Jun. IEEE; 2016.
- [52] Anderson JR, Ayalew B, Weiskircher T. Modeling a professional driver in ultra-high performance maneuvers with a hybrid cost MPC. In: American control conference (ACC), 2016. IEEE; 2016. p. 1981–1986.
- [53] Pacejka HB. Tyre and vehicle dynamics. Reprint ed. Amsterdam: Elsevier Butterworth-Heinemann; 2004; oCLC: 249713869.
- [54] Tesi A, Vinattieri F, Capitani R, et al. Development of an e-LSD control strategy considering the evolution of the friction torque with the wear depth. *SAE Int J Eng*. **2016**;9(3).
- [55] Zanon M, Frasch JV, Diehl M. Nonlinear moving horizon estimation for combined state and friction coefficient estimation in autonomous driving. In: 2013 European control conference (ECC). IEEE; 2013. p. 4130–4135.
- [56] Besselink IJ, Schmeitz AJ, Pacejka HB. An improved magic formula/swift tyre model that can handle inflation pressure changes. *Veh Syst Dyn*. **2010**;48(sup1):337–352.
- [57] Guiggiani M. Dinamica del veicolo. S.l.: CittStudi; 2007; oCLC: 800028905.
- [58] Edelsbrunner H. Alpha shapesa survey. *Tessellations Sci*. **2010**;27:1–25.
- [59] Domahidi A, Jerez J. FORCES Professional. embotech GmbH; 2014 Jul.
- [60] Zanelli A, Domahidi A, Jerez J, et al. FORCES NLP: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *Int J Control*; 2017.

- [61] Braghin F, Cheli F, Melzi S, et al. Race driver model. *Comput Struct*. **2008**;86(13–14):1503–1516.
- [62] Novi T, Capitani R, Annicchiarico C. An integrated artificial neural network unscented kalman filter vehicle sideslip angle estimation based on inertial measurement unit measurements. *Proc Inst Mech Eng Part D J Automob Eng*. p. 0954407018790646.
- [63] Novi T, Liniger A, Capitani R, et al. The influence of autonomous driving on passive vehicle dynamics. *SAE Int J Veh Dyn Stab NVH*. **2018**;2:285–295.