American Society of Mechanical Engineers

ASME Accepted Manuscript Repository

Institutional Repository Cover Sheet

First   _Last_

ASME Paper Title: Hierarchical Control for Trajectory Generation and Tracking Via Active Front Steering

Authors: Joseph S. Hellewell, Atanas A. Popov, Gary E. Burnett

*VOR (version of record)

Hierarchical control for trajectory generation and tracking via active front steering

Joseph S. Hellewell[*]
PhD student
Email: joseph.hellewell1
@nottingham.ac.uk

Atanas A. Popov
Professor of Engineering Dynamics
Email: atanas.popov@nottingham.ac.uk

Gary E. Burnett
Professor of Transport Human Factors
Email: gary.burnett@nottingham.ac.uk

Faculty of Engineering
University of Nottingham
Nottingham, UK

A new hierarchical model predictive controller for autonomous vehicle steering control is presented. The controller generates a path of shortest distance by determining lateral coordinates on a longitudinal grid, while respecting road bounds. This path is then parameterized by arc length before being optimized to restrict the normal acceleration values along the trajectory's arc length. The optimized trajectory is then tracked using a nonlinear model predictive control scheme using a bicycle plant model to calculate an optimal steering angle for the tires.

The proposed controller is evaluated in simulation during a double-lane-change maneuver, where it generates and tracks a reference trajectory while observing the road boundaries and acceleration limits. Its performance is compared to a controller without path optimization, along with another that uses a smooth, predetermined, reference path instead of creating its own initial reference.

It is shown that the proposed controller improves the tracking compared to a controller without path optimization, with a four-times reduction in average lateral tracking error. The average lateral acceleration is also reduced by 6%. The controller also maintains the tracking performance of a controller that uses a smooth reference path, while showing a much greater flexibility due to its ability to create its own initial reference path rather than having to follow a predetermined trajectory.

1 Introduction

Autonomous driving technology has become an increasingly popular research topic [1], in both academic and industrial environments. Efforts have primarily focused on environment detection, path planning and vehicle control. For control of the lateral response of the vehicle, research has focused both on conventional methods of lateral control such as Active Front Steering (AFS), either as a driver aid [2], or as an autonomous implementation, along with new techniques made possible using additional actuators. Direct Yaw-moment Control (DYC) [3] via either differential braking or active torque distribution to manipulate the longitudinal force of individual tires, is one such example of a control technique that is able to achieve more control authority by utilizing longitudinal tire forces, in a manner that is not achievable by human drivers.

To successfully implement autonomous vehicle behavior such as AFS, a control methodology which can handle complex, time-varying constraints and is adaptable to varying system parameters and passenger preferences in the problem definition is required. Model Predictive Control (MPC) is one such method [4, 5], as the use of optimization allows constraints to be actively considered in the design of the controller. MPC has many applications in automotive contexts, see [6] for examples. It has been shown to be effective at controlling Multi-Agent Systems [7], highlighting its utility in autonomous driving, where a fleet of vehicles communicate with one another to control and optimize their collective behavior.

In particular, nonlinear MPC [8, 9] allows for the use of higher-fidelity plant models, that give a more accurate portrayal of vehicle behavior. It allows for the inclusion of nonlinear constraints into the control problem, allowing for the diverse types of constraints present in autonomous driving, such as acceleration and road constraints, to be formulated explicitly into the problem formulation. This provides for the intuitive inclusion and modification of constraints, allowing

---

[*]Corresponding author.

1

for the control designer to change the dynamic performance to suit the personal preferences of passengers or specific behaviors of the autonomous vehicle.

It has always been a concern however, that due to the nature of MPC solving an optimization routine online and in-the-loop, that it is computationally expensive and prohibits its application to controllers needing operating frequencies of one cycle per second (Hz) and greater. From a computational complexity point of view, the value of the preview horizon is particularly important [10], so it is beneficial to use as small a value as possible, yet the preview horizon has a significant effect on both controller performance and stability.

A way of reducing the preview horizon while maintaining both stability and performance is therefore vital. Hierarchical multilayer control [11] is such a method and has been used in several scenarios related to vehicle dynamics and autonomous driving functionality [12, 13]. The upper level controls the slower dynamics of the system in consideration; specifying a trajectory for the lower levels to follow by using a large value for the preview horizon. The lower level uses a higher frequency to control the faster acting dynamics within the system allowing it to calculate the optimal control action over a shorter timescale.

Utilizing different prediction horizon lengths provides some stability to the control problem. The high-level control has an awareness of potential events long before they come into consideration by the low-level controller. This allows for the upper level reference generator to provide a revised trajectory, which mitigates difficult scenarios, such as sudden step changes to road bounds, for the lower level to track. This is done before the lower level controller, which incorporates dynamics that are more complex, must optimize the control variables to traverse a difficult scenario.

There are several reasons why a hierarchical framework is an attractive control methodology for complex tasks such as path planning and active steering control; it allows for the logical splitting of the control task into separate, discrete modules. Additionally, the hierarchical control framework is analogous to those that exist to model human driver behavior [14, 15]. The ability to split the different control tasks, so that they can be addressed separately, is also beneficial as it simplifies the control approach.

In [12] a two-tier control framework is presented consisting of path optimization, where a reference trajectory for the vehicle's lateral displacement and yaw angle, generated via trigonometric functions, is optimized to ensure that the paths normal acceleration does not exceed a specified limit. This optimal path is then passed to a nonlinear model predictive controller, which calculates an optimal steering angle to track the target lateral displacement and yaw angle. This controller was evaluated on a simulation vehicle in a low friction environment during a double-lane-change maneuver. The controller did not constrain the lateral displacement, which limited its application to highway environments. It also required a pre-determined reference trajectory to be provided, which limits it to pre-existing scenarios.

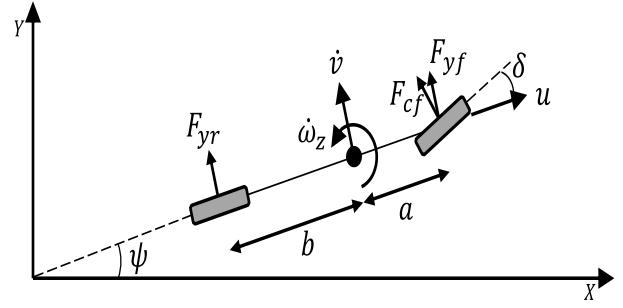In this paper, a Hierarchical Model Predictive Control

Fig. 1. The vehicle model shown in the global XY coordinate frame

(HMPC) framework is presented that extends the model presented in [12] with two key aspects of functionality, during constant speed scenarios. A third control module is added, termed the Path Generation Module (PGM), at the start of the optimization scheme, to generate an initial path without the need to specify one via the composition of trigonometric functions. The path is then parameterized by arc length so that it can be addressed in the time domain.

This combination of path generation, path optimization and vehicle control was chosen as it is analogous to the model proposed by Michon [16, 17] for a driver's model of the driving task. In this model, driving is split into high-level strategic planning, more immediate tactical driving and the low-level control of the vehicle.

Road bounds are also included in the control problem to ensure that the vehicle is still within the safe sections of the road. In this paper, it is assumed that the upper and lower lateral bounds are supplied to and utilized by each layer of the hierarchical controller, after being detected and determined via an appropriate method, such as via LIDAR or a computer vision system (see, for example, [18, 19]).

The second stage is the Path Optimization Module (POM) then constrains the normal acceleration of the trajectory, via path curvature, as in [12]. To accommodate the nature of the PGM, the POM calculates a yaw angle for a single steered axle vehicle to track. When calculating the yaw angle, it penalizes excessively large values of yaw angle which helps with stability in certain scenarios, such as highway driving.

The final stage of the controller uses a nonlinear vehicle model to track this reference trajectory by manipulating the steer angle of the front tires. Evaluation of the controller is completed using a simulation model of an autonomous vehicle completing a double-lane-change maneuver in MATLAB [20] at two longitudinal speeds.

The paper is structured as follows. In Section 2 the vehicle models used by the controller and for the simulated vehicle are presented. Section 3 details the controller methodology and implementation used in this paper. Finally, Section 4 describes the driving maneuver used to evaluate the controller performance, along with the simulation results.

2 Vehicle model

A five state nonlinear bicycle model is utilized in the simulations. The lateral and yaw acceleration of the vehicle is modelled along with the lateral and longitudinal displacement in a global reference frame for coordinate tracking. It is assumed for the analysis that the vehicle is travelling at a constant longitudinal speed u. Because of this, the normal force acting upon the tires is modelled as the vehicle's static weight distribution. Due to the tire model's nonlinear dependence on normal load, the vehicle model assumes two tires at each axle, each with half the axle load acting upon it.

Therefore, the state vector of the system is $z = (v, \omega_z, \psi, X, Y)^T$. The lateral velocity of the vehicle in its local reference frame is denoted as v, while ψ is the yaw angle of the vehicle defined as the rotation angle between the local and global longitudinal axes. Its derivative is ω_z, the vehicle's yaw rate. X and Y are the global coordinates of the vehicle's center of mass. The system of differential equations that describe the vehicle's motion is denoted by:

$$\dot{z} = f(z, u) \qquad (1)$$

where u is the control input vector, which for this implementation is simply the steer angle of the front tires, δ. An emboldened function denotes that it returns a vector value; otherwise, the function returns a scalar value.

The lateral and yaw acceleration of the vehicle body is given by the following equations:

$$\dot{v} = \frac{2}{m}\left(F_{yf} + F_{yr}\right) - \omega_z u$$
$$\dot{\omega}_z = \frac{2}{I}\left(aF_{yf} - bF_{yr}\right) \qquad (2)$$

The force terms in (2), calculated for a single tire on an axle, are given by referring the forces calculated by the Magic Tire Formula (MTF) [21] to the body-fixed coordinate frame. For the front tire the terms are referred from the tires coordinate frame to the body fixed coordinate frame, see Figure 1,

$$F_{yf} = F_{cf}\cos\delta, \qquad (3)$$

where the tire-fixed cornering term, F_{cf}, is obtained by the MTF. As there is no steer angle acting upon the rear tires, $F_{yr} = F_{cr}$. The MTF requires the calculation of the lateral slip angle, α, which is given for the front and rear tires as:

$$\tan\alpha_f = \frac{(v+a\omega_z)\cos\delta - u\sin\delta}{(v+a\omega_z)\sin\delta + u\cos\delta}, \qquad \tan\alpha_r = \frac{v - b\omega_z}{u}.$$
$$(4)$$

The global longitudinal and lateral coordinates are computed

Table 1. Parameter values used by the vehicle models and Pacejka Magic Tire Formula

Parameter	Value	Units
m	2050	kg
I	3344	kg·m^2
a	1.1	m
b	1.4	m
σ	0.3	m
MTF Parameters		
B	-11.5	-
C	1.35	-
E	-0.85	-

by the integration of the following equations:

$$\dot{X} = u\cos\psi - v\sin\psi, \quad \dot{Y} = u\sin\psi + v\cos\psi \qquad (5)$$

This vehicle model outlined above is used in Section 3.4 as the prediction model, where the integration in time is performed using the forward Euler method with a fixed time-step. The parameter values used in the model are shown in Table 1.

2.1 Tire transience

The tire forces calculated via the MTF are steady-state approximations that assume a tire's slip angle is instantly reached. If the steer angle of the tire varies rapidly, such as in an active steering application using a piecewise constant input, then this assumption is not representative of the behavior. A first order tire transience is added to the vehicle model to characterize the lag in the tire force:

$$\dot{\alpha} = \frac{u}{\sigma}(\alpha_s - \alpha) \qquad (6)$$

where α_s is the static tire slip, as calculated via (4) and α is now the apparent slip angle, used to calculate the lateral tire force in the MTF. σ is the relaxation length of the tire. This augmented system of equations, $f_{tt}(z_{tt}, u)$, is used as the simulation model, with the state vector z extended to include the apparent slip angle for both the front and rear axles.

3 Control methodology

The proposed hierarchical structure splits the control task into three separate modules composed of: path generation, path optimization and vehicle control modules (see the schematic in Figure 2). The first module generates a static path as a first approximation for the vehicle to track.
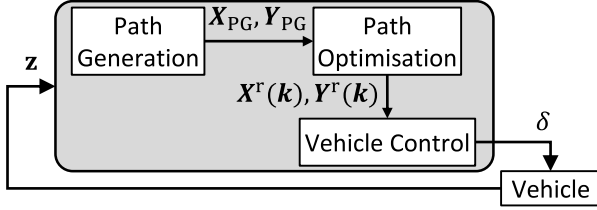
3

Fig. 2. The hierarchical framework used by the controller, with the interaction with the vehicle plant shown. PGM, POM, VCM denote the Path Generation, Path Optimization and Vehicle Control Modules, respectively

This path is then taken by the trajectory optimization routine in which the path is altered to ensure that specified lateral acceleration constraints are not exceeded. The final module is responsible for vehicle control, where it tracks the optimised reference path and tracks them by optimising the steer angle of the vehicle's front axle.

The system is designed with the assumption that each control module has a different call rate, allowing for the path generation module to have a longer computation time due to the much larger preview horizon than that used by the path planner and trajectory optimization routines. For the current analysis, the computational delay is omitted to simplify the analysis, and because this delay is highly dependent on the hardware on which the controller is implemented.

There are existing techniques, however, to study the effect on performance and stability of a time delay [22] and how to account for it [8]. The lack of a computational delay allows the presented controller to react more quickly to changes in the environment, making it possible to take corrective action more quickly as well as reducing the prediction model error. For the results presented, this assumption has negligible impact as there are no sudden changes in the simulated environment.

3.1 Hierarchical structure

The individual modules of the controller have different control frequencies. Therefore the hierarchical control framework, which is responsible for coordinating its modules, must call the individual controllers at different time steps. Call ratios are defined, which determine how many times a subordinate module is called for a single call to its parent module. As there are three controllers in three separate tiers of the hierarchy, there are two call ratios, ρ_1 which determines how many times the path optimization module is called for single call to the path planning module, and ρ_2 determines how many times the vehicle controller is called for every trajectory optimization call:

$$\rho_1 = \frac{\Delta T_{pg}}{\Delta T_{po}}, \qquad \rho_2 = \frac{\Delta T_{po}}{\Delta T_{vc}} \qquad (7)$$

where ΔT is the call period, with the subscripts 'pg', 'po', 'vc', denoting the Path Generation, Path Optimization and Vehicle Control modules, respectively.

3.1.1 Choosing the preview horizons

One key design choice, critical for both the performance and stability of the hierarchical system, is the preview horizon value chosen for each of the three control modules [8]. For both performance and stability, values as large as possible are desirable. However, increasing the preview horizon increases the computational complexity dramatically [10].

Therefore, there is a compromise to be made when choosing the preview horizon and no analytical method for determining it. Instead, rules-of-thumb are employed along with manual tuning of values based upon a given scenario [23].

Several conditions must be met that guide the choice of preview horizons for the hierarchical framework. The most important one is to ensure that the reference trajectory provided to a subordinate controller is at least as long as that of the parent controller:

$$T_{pg} \geq T_{po} \geq T_{vc}, \qquad (8)$$

where T_{pg}, T_{po}, T_{vc} represent how far the path generation, path optimization and vehicle control modules predict into the future respectively. The next factor to consider is that the PGM works in the spatial domain, so one can write the preview horizon constraint between the path generation and optimization modules as:

$$s_{pg} \geq s_{po}, \qquad (9)$$

where these values denote the final displacement values over a single given prediction. The problem is that the arc length generated by the PGM is not fixed, as the lateral coordinates are the optimization variables. It is assumed that for the driving scenarios considered, the average lateral step size, $\Delta \bar{Y}$, chosen by the POM is much smaller than the fixed longitudinal step size, ΔX ($\Delta \tilde{Y} \ll \Delta X$). A value, $\varepsilon > 0$, is given so that when using the following expression:

$$s_{pg} = N_{pg} \sqrt{\Delta X^2 + \varepsilon} \qquad (10)$$

for an approximation of the arc length of the PGM path, Eq. (10) holds. ΔX is chosen to be the distance travelled by the vehicle over the fixed time-step Δt, used by both the POM and VCM:

$$\Delta X = u \Delta t, \qquad (11)$$

so that the PGM is capable of reacting to different road environments and the displacement values s_{pg}, s_{po} are both func-

4

tions of the vehicles longitudinal speed, as well as ensuring, for sensible values of ε, Eq. (9) holds.

For the relationship between the POM and VCM the constraint $N_{po} \geq N_{vc}$ ensures that the second inequality in Eq. (8) holds, as they both utilize the same fixed time step. The values are then chosen to provide stability to the system.

3.2 Geometric path generation

The PGM creates the initial reference path for the POM by generating a set of N_{pg} Y-coordinate values, denoted $\boldsymbol{Y}_{pg}$, based upon a longitudinal grid with equal step size, ΔX, creating a set of N_{pg} Cartesian coordinate pairs $(\boldsymbol{X}_{pg}, \boldsymbol{Y}_{pg})$. At each X-coordinate the road limits are defined by an upper and lower bound resulting in two vectors, $\boldsymbol{Y}_{ub}$ and $\boldsymbol{Y}_{lb}$, that represent bound constraints on the absolute lateral displacement. The controller must therefore determine a coordinate between these limits at each X-coordinate.

The PGM is designed to generate a spatial path which minimizes the total distance travelled between the initial and final X-coordinates. This assumes that if a vehicle was travelling along a path of low or zero curvature, which as an example, is true for motorway driving, then travelling in a straight line is the optimal behavior, and a lane change is only made to avoid a slower moving vehicle ahead.

To accomplish this, the optimization routine calculates N_{pg} lateral steps $\Delta\boldsymbol{Y}$, which represent the lateral step between adjacent Y-coordinates. The absolute lateral position, $\boldsymbol{Y}_{pg}$, is then calculated componentwise as:

$$Y_{pg,i+1} = Y_{pg,i} + \Delta Y_i, \qquad \forall i = 1, \ldots, N_{pg}, \quad Y_0 = \tilde{Y}_0 \quad (12)$$

Where $\tilde{Y}_0$ is the initial measured lateral position of the vehicle at the start of the optimization problem. In this paper, the value is taken from the state vector $\boldsymbol{z}$ of the simulation model.

The bound constraints included in the optimization routine have a safety margin, Y_{sm}, included when determining the optimization bounds to account for vehicle width, which reduces the apparent feasible space available to the path planner.

The resulting optimization problem, which finds the optimal reference path by choosing the vector of lateral step-sizes $\Delta\mathbf{Y}$, at time $t = i$ can be written as:

$$\underset{\Delta\mathbf{Y}}{\text{minimise}} \quad \sum_{i=t+1}^{t+N_{pg}} \Delta Y_i^2 \qquad (13a)$$

$$\text{subject to} \qquad Y_{pg,i} = Y_{pg,i-1} + \Delta Y_i, \qquad (13b)$$

$$Y_{lb,i} + Y_{sm} \leq Y_{pg,i} \leq Y_{ub,i} - Y_{sm} \qquad (13c)$$

The optimal initial reference path generated by the PGM is then calculated by Eq. (12), using the vector $\Delta\boldsymbol{Y}^*$ that represent the optimal solution to the optimization problem. This problem is of Quadratic Programming (QP) form with inequality constraints, for which there are fast solution methods for both large (sparse) and small optimization problems [24].

3.2.1 Path parameterization

In the POM, the $\boldsymbol{X}_{pg}$ and $\boldsymbol{Y}_{pg}$ variables that make up the geometric path are addressed separately. In order to do this, the PGM module parameterizes the path coordinates so that they are functions of the path's arc length.

The continuous path that is assumed to connect coordinate pairs, $\mathbf{p}$, is approximated via straight-line segments, with $\mathbf{p}_i$ denoting the (X, Y) coordinates at the i-th time-step. From that, the displacement between the points $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$, denoted Δs_i is given by:

$$\Delta s_i = \sqrt{\Delta X^2 + \Delta Y_i^2}, \qquad \forall i = 1, \ldots, N_{pg} \qquad (14)$$

where ΔX is the fixed step-size utilized in the PGM module and ΔY_i is the lateral step-size between $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ which can be taken from the solution of the optimization problem defined in Eq. (13). The vector containing the absolute displacement values from the initial point of the path up to the point at the preview horizon can be calculated elementwise as:

$$s_i = s_{i-1} + \Delta s_i, \qquad \forall i = 1, \ldots, N_{pg}, \quad s_0 = 0 \qquad (15)$$

This set of displacement values, which are monotonically increasing, can now be used to parametrize the $\boldsymbol{X}_{pg}$ and $\boldsymbol{Y}_{pg}$ coordinate pairs, which are denoted $\boldsymbol{X}_{pg}(\boldsymbol{s})$ and $\boldsymbol{Y}_{pg}(\boldsymbol{s})$.

So that the PGM coordinates can be tracked in the time domain, a time-based interpolation vector, denoted $\hat{\boldsymbol{s}}$, is required. As this paper considers constant longitudinal speed scenarios and, assuming that the lateral velocity is of a much smaller magnitude, this interpolation vector are calculated elementwise as:

$$\hat{s}_i = \hat{s}_{i-1} + u\Delta t, \qquad \forall i = 1, \ldots, N_{po}, \quad \hat{s}_0 = 0 \qquad (16)$$

where Δt is the time-step of the VCM. The resulting vector allows for the interpolation of the vector $\boldsymbol{Y}_{pg}$ at any time-step up until N_{po} using $\hat{\boldsymbol{s}}$ as the interpolation vector. Linear interpolation is then used to create the initial reference trajectory for the path optimization.

3.3 Path optimization

The POM attempts to match the parametrized $\boldsymbol{X}_{pg}(\boldsymbol{s})$ and $\boldsymbol{Y}_{pg}(\boldsymbol{s})$, while considering kinematic and dynamic factors affecting the reference path generation problem. It does this by assuming that the vehicle is a point mass travelling along the reference spatial path. The path coordinates from the PGM are modified so that the acceleration component perpendicular to the optimized path does not exceed specified limits.
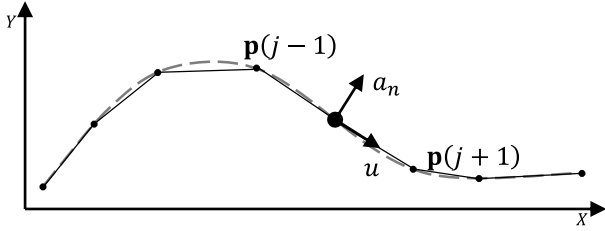
Fig. 3. The point mass model used in the path optimization scheme. The solid straight lines show the assumptions used to calculate path length and curvature of the assumed continuous path (dashed line). The filled dots are the location of a reference path (X, Y) pair at a single time-step

Along with a reference displacement, the path optimization module also determines a yaw reference profile for the vehicle control module to track. Following [25], where the optimal yaw angle is tangential to the reference path, the yaw reference is calculated as:

$$\psi^r = \arctan\left(\frac{\Delta Y}{\Delta X}\right) \tag{17}$$

Where ΔX and ΔY are the finite differences of the X and Y values. A penalty to the yaw magnitude is included in the POM to encourage the module to return a path that is smooth and does not have excessively large deviations. This can occur due to the PGM returning a path that consists of straight lines between changes in road bounds.

The normal acceleration is calculated as the centripetal acceleration of a point mass travelling along a path with curvature, κ, at a tangential velocity, u (see Figure 3). An approximation of the path curvature is calculated using finite difference approximations for the first and second derivatives of the X and Y coordinates with respect to time. The curvature at the jth time-step is given by:

$$\kappa_j = \frac{\Delta X_j \Delta^2 Y_j - \Delta Y_j \Delta^2 X_j}{\left(\Delta X_j{}^2 + \Delta Y_j{}^2\right)^{\frac{3}{2}}} \tag{18}$$

where $\Delta X, \Delta Y$ and $\Delta^2 X, \Delta^2 Y$ denote the first and second backward finite differences for the $X(t)$ and $Y(t)$ coordinates, which are calculated by backward finite differences. Using the backwards finite difference method, it is possible to account for the data points that are lost via the finite difference method, with data from the previous time-step.

As path curvature is the inverse of radius of curvature, the vehicle's lateral acceleration is given as the centripetal acceleration of the point mass travelling along the arc length:

$$a_{n,j} = \kappa_j \cdot u^2 \tag{19}$$

So far, the model outlined for the POM is not constrained by

inertia effects and so is free to vary the path displacement as quickly as possible so long as inertia and displacement constraints are obeyed. This can result in a path that is not realizable by the VCM controller. To restrict the optimal path to a set that is more easily tracked by the VCM, the step-size in lateral acceleration is constrained in the POM problem formulation via limiting the step-size between normal acceleration values calculated at each time-step.

The optimization problem solved at $t = j$ can be formulated as:

$$\begin{aligned} \underset{\boldsymbol{Z}_{\text{po}}}{\text{minimise}} \quad & J = \sum_{j=t+1}^{t+N_{\text{po}}} \left\| \boldsymbol{z}_j - \boldsymbol{z}_j^r \right\|_{\boldsymbol{Q}_{\text{po}}}^2 \end{aligned} \tag{20a}$$

$$\text{subject to} \quad u^2 \Delta t^2 = (X_{\text{po,j}} - X_{po,j-1})^2 + (Y_{\text{po,j}} - Y_{po,j-1})^2, \tag{20b}$$

$$\hat{\boldsymbol{Y}}_{\text{lb}}(\boldsymbol{X}_{\text{po}}) \leq \boldsymbol{Y}_{\text{po}} \leq \hat{\boldsymbol{Y}}_{\text{ub}}(\boldsymbol{X}_{\text{po}}), \tag{20c}$$

$$-\bar{a}_{\text{n}} \leq \boldsymbol{a}_{\text{n}} \leq \bar{a}_{\text{n}}, \tag{20d}$$

$$-\overline{\Delta a}_{\text{n}} \leq a_{\text{n,j}} - a_{\text{n,j}-1} \leq \overline{\Delta a}_{\text{n}}, \tag{20e}$$

$$(X_{k-2}, X_{k-1}, X_k) = (\bar{X}_{k-2}, \bar{X}_{k-1}, \bar{X}_k), \tag{20f}$$

$$(Y_{k-2}, Y_{k-1}, Y_k) = (\bar{Y}_{k-2}, \bar{Y}_{k-1}, \bar{Y}_k) \tag{20g}$$

Where $\boldsymbol{Z}_{\text{po}} = \left(\boldsymbol{z}_{\text{po,t+1}}, \ldots, \boldsymbol{z}_{\text{po,t+N}_{\text{po}}}\right)$ is the sequence of reference trajectories generated by the POM to optimize the cost function Eq. (20), with $\boldsymbol{z}_{\text{po,j}}^T = (X_j, Y_j, \psi_j)$ and $\boldsymbol{z}_j^r$ are the corresponding reference values provided by the PGM. $\|\boldsymbol{x}\|_{\boldsymbol{M}}^2$, as seen in Eq. (20), denotes the square of the weighted matrix norm $\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}$. Equation (20b) constrains the re-planned coordinates to ensure that the distance between two adjacent coordinates is equal to the distance travelled by the point mass model when moving at the fixed longitudinal speed, u, over the interval of a single time-step, Δt.

To calculate the lower and upper bound terms in Eq. (20c), linear interpolation is used. The optimized coordinates $\boldsymbol{X}_{\text{po}}$ are used as the interpolation vector to interpolate the upper and lower bounds based upon the original fixed $\boldsymbol{X}_{\text{pg}}$ grid generated by the path planning module.

Equations (20f) and (20g) specify the additional data variables that must be provided to the optimization routine for the calculation of path curvature. The variables with overbars denote that they are taken from the vehicle plant itself.

3.4 Vehicle control

The task of the Vehicle Control Module (VCM) is to calculate an optimal control sequence for the vehicle plant when called at time k. The VCM attempts to track the reference coordinates supplied to it by manipulating the internal plant model's steering angle, up to preview horizon N_{vc}. The plant model that is utilized is taken as the vehicle model in Section 2. The continuous time model is discretized using the forward Euler method:

$$\begin{aligned} \boldsymbol{z}_{k+1} &= \boldsymbol{z}_k + \Delta t \cdot \dot{\boldsymbol{z}}_k \\ \dot{\boldsymbol{z}}_k &= \boldsymbol{f}(\boldsymbol{z}_k, \boldsymbol{u}_k) \end{aligned} \tag{21}$$

6

The lateral acceleration is constrained, as was the normal acceleration of the point mass model. The lateral acceleration is given by:

$$a_y = \dot{v} + \omega_z u \qquad (22)$$

In the optimization routine, the lateral acceleration is calculated from the state and rate vectors in Eq. (21) at every time step up to the preview horizon. The optimization scheme constrains these predicted values between upper and lower bounds.

Therefore the VCM solves the following optimization problem at $t = k$:

$$\underset{\boldsymbol{u}}{\text{minimise}} \quad J = \sum_{k=t+1}^{t+N_{\text{vc}}} \|\boldsymbol{z}_k - \boldsymbol{z}_k^r\|_{\boldsymbol{Q}_{\text{vc}}}^2 + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}_{\text{vc}}}^2 + \|\Delta\boldsymbol{u}_{k-1}\|_{\boldsymbol{S}_{\text{vc}}}^2 \qquad (23\text{a})$$

subject to
$$\boldsymbol{z}_k = \boldsymbol{z}_{k-1} + \Delta t \cdot \boldsymbol{f}(\boldsymbol{z}_{k-1}, \boldsymbol{u}_{k-1}), \qquad (23\text{b})$$

$$\Delta\boldsymbol{u}_{k-1} = \boldsymbol{u}_{k-1} - \boldsymbol{u}_{k-2}, \qquad (23\text{c})$$

$$u^2 \Delta t^2 = (X_{\text{vc},k} - X_{\text{vc},k-1})^2 + (Y_{\text{vc},k} - Y_{\text{vc},k-1})^2, \qquad (23\text{d})$$

$$-\bar{\boldsymbol{u}} \leq \boldsymbol{u}_{k-1} \leq \bar{\boldsymbol{u}}, \qquad (23\text{e})$$

$$-\overline{\Delta\boldsymbol{u}} \leq \Delta\boldsymbol{u}_{k-1} \leq \overline{\Delta\boldsymbol{u}}, \qquad (23\text{f})$$

$$\hat{\boldsymbol{Y}}_{\text{lb}}(\boldsymbol{X}_{\text{vc}}) \preceq \boldsymbol{Y}_{\text{vc}} \preceq \hat{\boldsymbol{Y}}_{\text{ub}}(\boldsymbol{X}_{\text{vc}}), \qquad (23\text{g})$$

$$-\bar{a}_y \leq a_y \leq \bar{a}_y \qquad (23\text{h})$$

Where $\boldsymbol{u} = \left(\boldsymbol{u}_t, \ldots, \boldsymbol{u}_{t+N_{\text{vc}}-1}\right)$ is the sequence of optimal control vector inputs applied to the system, which for this application is the steer angle of the front tires. Like in Eq. (20c), interpolation is used in Eq.. (23g) to obtain the upper and lower bounds at each predicted $(X_{\text{vc},k}, Y_{\text{vc},k})$ point.

3.5 Summary of the control tasks

To conclude the control methodology section, a summary of the steps taken by the hierarchical controller is given below:

1. The geometric path generation module generates a set of N_{pg} $\boldsymbol{Y}_{\text{pg}}$ coordinates based on a uniformly spaced vector of $\boldsymbol{X}_{\text{pg}}$ coordinates.
2. The path is then parametrized based upon arc length.
3. The set of N_{pg} coordinate pairs is passed to the re-planning module. The controller then returns a reference trajectory of length N_{po} for the vehicle control module. The re-planning module is called ρ_1 times for every path planner call.
4. For each call to the vehicle control module, it selects the relevant trajectory of length N_{vc} from the re-planner trajectory. It tracks this reference trajectory by optimizing the steer angle of the nonlinear bicycle model front wheels. The path planning controller is called ρ_2 times for every trajectory re-planner call.
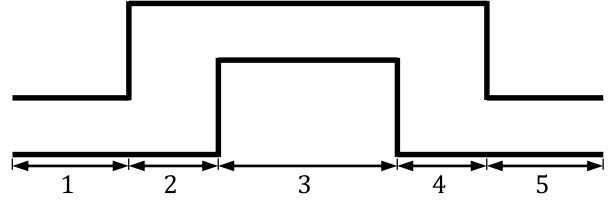

Fig. 4. The double-lane-change maneuver course. Not to scale

Table 2. Dimensions of the course

Section	1	2	3	4	5
Length [m]	15	40	25	25	55
Width [m]	3.5	6.5	3.5	6.5	3.5
Offset [m]	0	0	3	0	0

4 Results

The vehicle controller is evaluated in simulation, performing a double-lane-change maneuver. The course consists of three straight line sections, with the middle section offset by a distance of 3.5 m (see Figure 4). The longitudinal distance is 150 m, with the individual section length's shown in Table 2.

To demonstrate the performance of the proposed hierarchical controller, it is compared against two other controllers derived from it. For ease, the controllers shall be referred to as controller A, controller B and controller C, with specifications as follows:

- Controller A: The hierarchical controller; with PGM, POM and VCM,
- Controller B: A controller consisting of path generation and active front steering, i.e. no active POM. The POM for controller B simply returns parametrized values (by arc length) of the PGM reference path.
- Controller C: The PGM is replaced by a smooth reference trajectory generator, created for the specific double-lane-change maneuver.

The purpose of controller B is to demonstrate that the POM performs a valuable function and that splitting the control task into separate control layers is a worthwhile design goal, even when there is a path provided by. Conversely, controller C's purpose is to demonstrate that the performance of controller A is comparable to that of a specifically crafted, continuously smooth reference trajectory, while being completely flexible to the driving scenario.

Table 3 shows the preview horizon and time period for the three control modules. The values chosen for the module time periods results in the call ratios $\rho_1 = 2$ and $\rho_2 = 5$. The PGM preview horizon was chosen to comfortably cover the majority of the scenario in a single optimisation call. This is possible as it is a QP problem and so the execution time of the controller is not a limiting factor. The values for the POM and VCM were chosen to give stability to the controller
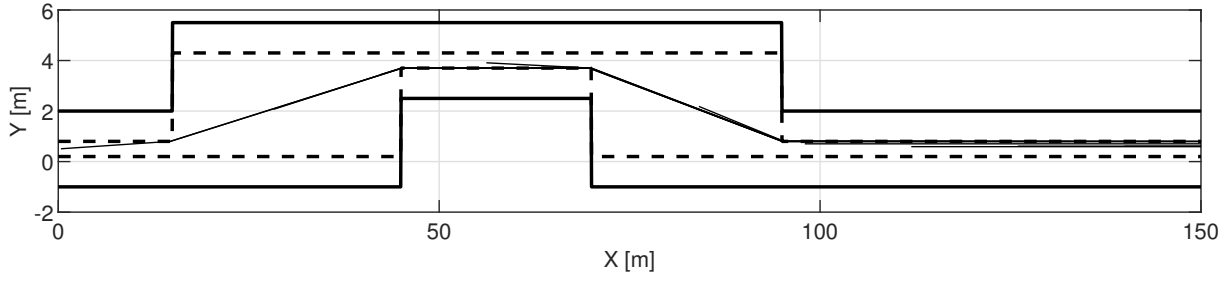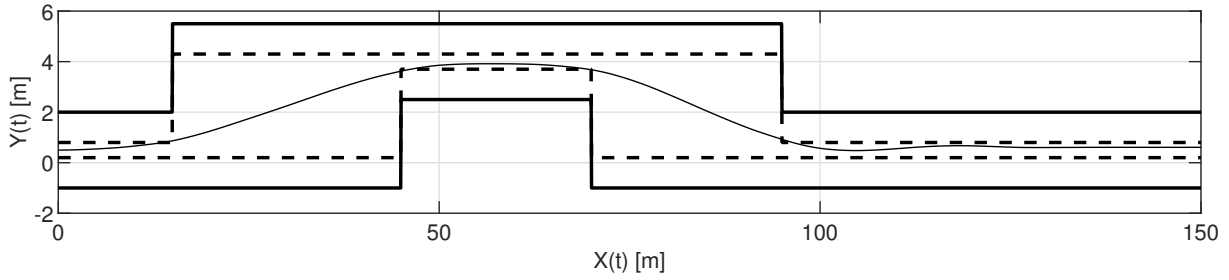
(a) Lateral reference coordinates of the path generation module
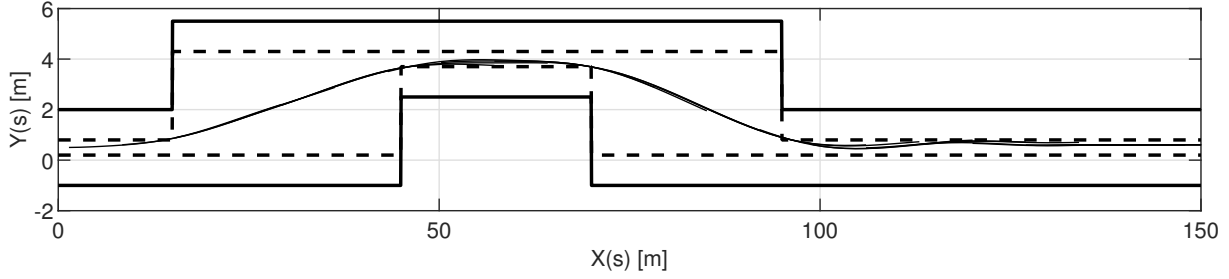
(b) Parameterized coordinates of the path optimization module

(c) The vehicle's displacement throughout the simulation

Fig. 5. Trajectory generation, optimization and tracking during the double-lane-change maneuverer at 14 m/s. The physical road limits (solid lines) and reduced bounds, including the lateral safety margin, (dashed lines) are also shown

Table 3. Controller prediction horizon values used for the simulation runs

Module	PGM	POM	VCM
Preview horizon, N	300	30	16
Time period, ΔT [s]	1	0.5	0.1

Table 4. Controller weighting values

$\boldsymbol{Q}_{PO}$	$\text{diag}(10, 10, 5)$
$\boldsymbol{Q}_{VC}$	$\text{diag}(10, 10, 250)$
$\boldsymbol{R}_{VC}$	1
$\boldsymbol{S}_{VC}$	50

at both 14 m/s (31 mph) and 20 m/s (45 mph). Table 4 shows the weighting values used by the three control variants, chosen to give stability and adequate performance based on controller, A determined during a trial-run of the lane change scenario. Tables 5 and 6 show the values used as constraints in the POM and VCM respectively.

Table 5. Constraint values used in the path optimisation module

Parameter	Value	Units
$\bar{a}_n$	0.3	g
$\overline{\Delta a}_n$	0.25	g/s

Table 6. Constraint values used in the vehicle control module

Parameter	Value	Units
$\bar{a}_y$	0.3	g
$\bar{\boldsymbol{u}}$	6	°
$\overline{\Delta \boldsymbol{u}}_n$	5	°/s

4.1 Controller A simulation results

Figure 5(a) shows all sets of the Y-coordinates generated over the course of the maneuver; representing stage 1 of the control scheme. As would be expected, the PGM generates straight line paths from the location of the vehicle when the PGM is called and between the apexes of the constraint
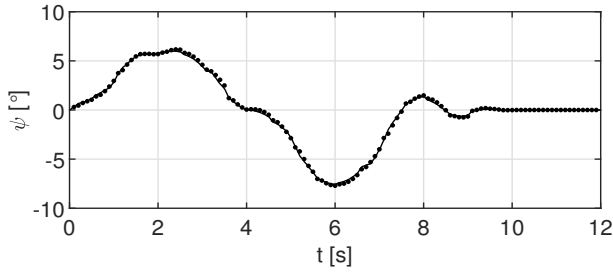
8

Fig. 6. The yaw trajectory of the simulation model at 14 m/s. The reference yaw trajectory, provided by controller A at penultimate calls ans used to calculate the RMS tracking error, are shown as points

(a) Lateral acceleration of simulation vehicle, control limits of ± 0.3g shown as dashed line

(b) AFS input applied to the vehicle by controller A

Fig. 7. The hierarchical MPC controller is capable of navigating the double-lane-change maneuverer at 14 m/s

bounds.

Figure 5(b) shows the effect the POM has on the PGMs initial path. There are twice as many reference trajectories in Figure 5(b) than Figure 5(a), matching the call ratio between the two modules. The POM smooths out the PGM path around the apexes, where the road bounds change.
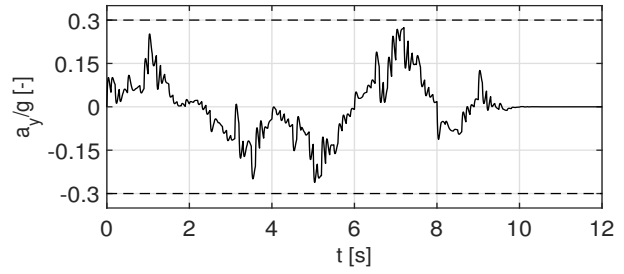
At 14 m/s (31 mph) the controller can closely track the references provided to it. The effect of solving the path generation problem from the vehicle's current location can be observed in Figure 5(a), showing the primary benefit of the hierarchical controller; the feasible reference path is always in the close neighborhood of the vehicle.

Figure 5(c) shows the displacement of the simulation vehicle and it can be observed that the simulation vehicle sucessfully navigates the course at 14 m/s. Here the effect of including bound constraints on the global lateral coordinates can be observed. In [12], the path optimization minimizes the vehicles lateral acceleration by reducing the maximum Y value, i.e. straightening the reference path. In this scenario, it is not possible to reduce the maximum lateral coordinate, so instead, the POM increases the maximum lateral coordinate value to ensure bound constraints, and acceleration constraints are satisfied.
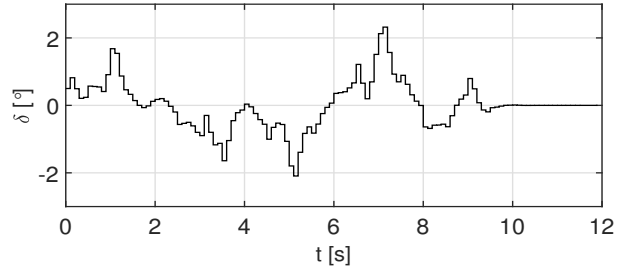
There are small constraint violations that occur on several of the course apexes. These occur due to the way the lateral coordinate constraints are enforced. In the interval between two discrete time-steps, there is no guarantee that the vehicle will not violate the road bounds. It is therefore necessary when choosing the lateral safety margin to factor in the constraint violation that can occur between VCM time-steps, along with the vehicles width.

Figure 6 shows the yaw profile of the simulation vehicle along with the reference profile made up of the most up-to-date value from the last POM call. The yaw profile is tracked closely by vehicle, and at the end of the run, the penalty on yaw deviation in the vehicle control module sucessfully damps the vehicle trajectory to a straight line at the end of the simulation run.

Figure 7 shows the lateral acceleration of the vehicle and the corresponding steering input used to track the coordinates. Figure 7(a), which shows the lateral acceleration, shows the inclusion of a penalty on lateral acceleration in the optimization cost function. The acceleration bounds im-

posed on the VCM are not saturated by the simulation vehicle. In this instance, this is caused by the differences in the VCM's prediction model and the simulation model, namely the tire transience and the numerical integration method. The VCM saturates the lateral acceleration limits, as it attempts to track the lateral displacement reference as closely as possible.

4.1.1 High-speed run

The simulation manoeuvre is repeated, at a higher longitudinal speed of 20 m/s, to show that the proposed controller maintains its performance at higher speeds.

Figure 8(a) shows that the lateral displacement of controller A at 20 m/s, with the road and controller bounds, along with the reference trajectory. The vehicles displacement looks similar, although there is an overshoot as the vehicle returns to the initial lane. This overshoot occurs to decrease the path curvature, reducing the lateral acceleration.

Table 7 shows that at higher speeds, there is a slight degradation in both maximum and RMS lateral tracking errors, reflecting the increase in difficulty of tracking at higher speeds.

Figure 8(b) shows that the hierarchical controller maintains good yaw tracking at the higher longitudinal speed. Although the RMS tracking error is almost double the value that it is at 14 m/s, the maximum tracking error is only increased by 7% and the absolute value of the RMS error is still more than adequate for automotive scenarios.

In Figure 8(c), it can be seen that the lateral acceleration of the simulation vehicle exceeds the limits imposed on the hierarchical controller, with a maximum acceleration of 0.37 g occurring during the simulation run. This excessive acceleration can be attributed to both the tire transience included

9

Table 7. Controller A tracking errors for the low and high-speed scenarios

u	Y_{max}	Y_{RMS}	ψ_{max}	ψ_{RMS}	$a_{y,RMS}/g$
[m/s]	[cm]	[cm]	[deg]	[deg]	[−]
14	3.98	1.30	0.82	0.17	0.09
20	6.34	1.94	0.88	0.31	0.15

Table 8. Tracking errors for controllers B and C at 20 m/s

Controller	Y_{max}	Y_{RMS}	ψ_{max}	ψ_{RMS}	$a_{y,RMS}/g$
	[cm]	[cm]	[deg]	[deg]	[−]
B	22.33	7.98	3.58	0.90	0.16
C	8.14	2.22	1.28	0.34	0.15

in the simulation model, and the more accurate numerical integration scheme used in the simulation compared to the MPC prediction model. This constraint violation is not predicted by the MPC controller and the simulation models response appears to suggest that the rapidly switched steering input has a significant effect on the common assumption that steady state tire force values are appropriate in control synthesis and evaluation of discrete-time steering controllers.

Table 7 shows the 67% increase in RMS lateral acceleration. Despite the fact that this is an average measure of lateral acceleration, including the final two seconds during which it is very low, this 67% increase is more than four times smaller than the increase in centripetal acceleration that might be naively expected due to the increase in longitudinal velocity, suggesting that the constraints effectively change the dynamic behavior of the vehicle.

4.2 Comparison of controller performance

To provide a reference to the performance of controller A, the simulation vehicle is guided through the double-lane-change maneuver is done again with two derivative controllers. Controller B does not perform the path optimization routine, and instead supplies parameterized coordinates from the initial PGM path for the VCM to track. The purpose of this controller is to demonstrate the performance benefit of a hierarchical architecture containing path optimization, both from a tracking point of view, along with respect to constraint satisfaction.

Controller C replaces the PGM of controller A with a smooth reference trajectory, based on the longitudinal displacement of the vehicle. The lateral displacement reference is given by

$$Y^r(X) = \frac{\Delta y_1}{2}\left(1 + \tanh z_1\right) - \frac{\Delta y_2}{2}\left(1 + \tanh z_2\right) \quad (24)$$

where

$$z_1(X) = \alpha\frac{X - X_{s1}}{\Delta X_1} - \frac{\alpha}{2}, \quad z_2(X) = \alpha\frac{X - X_{s2}}{\Delta X_2} - \frac{\alpha}{2} \quad (25)$$

and $\alpha = 1.4$, $\Delta X_1 = 20$, $\Delta X_2 = 20$, $\Delta Y_1 = 4$, $\Delta Y_2 = 4.25$, $X_{s1} = 24$ and $Y_{s2} = 71.25$. These values were chosen by visually fitting a curve to the feasible road space.

Figure 8(d), shows that the lateral displacement of the

vehicle when steered by controller B is similar to that of controller A (Figure 5(c)), although Table 8 details the degradation in tracking performance measured via both RMS and maximum deviation. There is a rather significant deviation in the maximum lateral tracking error, being three and a half times greater than controller A at 20 m/s. Part of the reason for this is that as the POM is inactive, the reference path to the VCM controller is only updated once every second compared to the two times a second that the occurs in controllers A and C due to the POM.

Figure 8(e) shows in greater clarity the effect of deactivating the POM; as the POM takes no action to minimize the second derivatives of their reference trajectory. This results in large steps in the yaw trajectory, which is not physically realizable, resulting in the comparatively poor yaw tracking performance. As with controller A, the lateral acceleration of the simulation vehicle exceeds the acceleration limit imposed by the controller, particularly when the vehicle begins to return to the initial lane.

Figure 8(g) shows the lateral tracking performance of controller C. As would be expected, the displacement of the simulation vehicle, when tracking a smooth reference, is smooth. Controller C produces a smooth yaw profile, with good tracking performance. Table 8 shows that the trajectory tracking performance is actually worse than controller A at 20 m/s (see Table 7). This is partly due to nature of the reference trajectory. For controller C, the reference trajectory is a function of longitudinal displacement only, and therefore any tracking errors accumulate over time. For controller A, this is not an issue as both the PGM and POM generate reference trajectories from the vehicles location. This means that the reference trajectory is generated so that it can be well tracked.

An argument is made by Wang and Boyd in [10], which is that with MPC optimization is used to find good control, rather than solving the optimization problem accurately. Therefore, if MATLAB's optimization solver 'fmincon' does not find an optimal control sequence to solve the MPC optimization problem to within the specified solution tolerances, the suboptimal solution is still applied to the system as otherwise. As can be observed by the results, this still results in adequate tracking performance, and the lateral acceleration constraints, along with the control input constraints are always observed.

5 Conclusions

A hierarchical model predictive controller for an autonomous double lane change maneuver has been presented.

10

(a) Lateral tracking   (b) Yaw tracking of controller A at 20 m/s   (c) Lateral acceleration

(d) Lateral tracking   (e) Yaw tracking of controller B   (f) Lateral acceleration

(g) Lateral tracking   (h) Yaw tracking of controller C   (i) Lateral acceleration
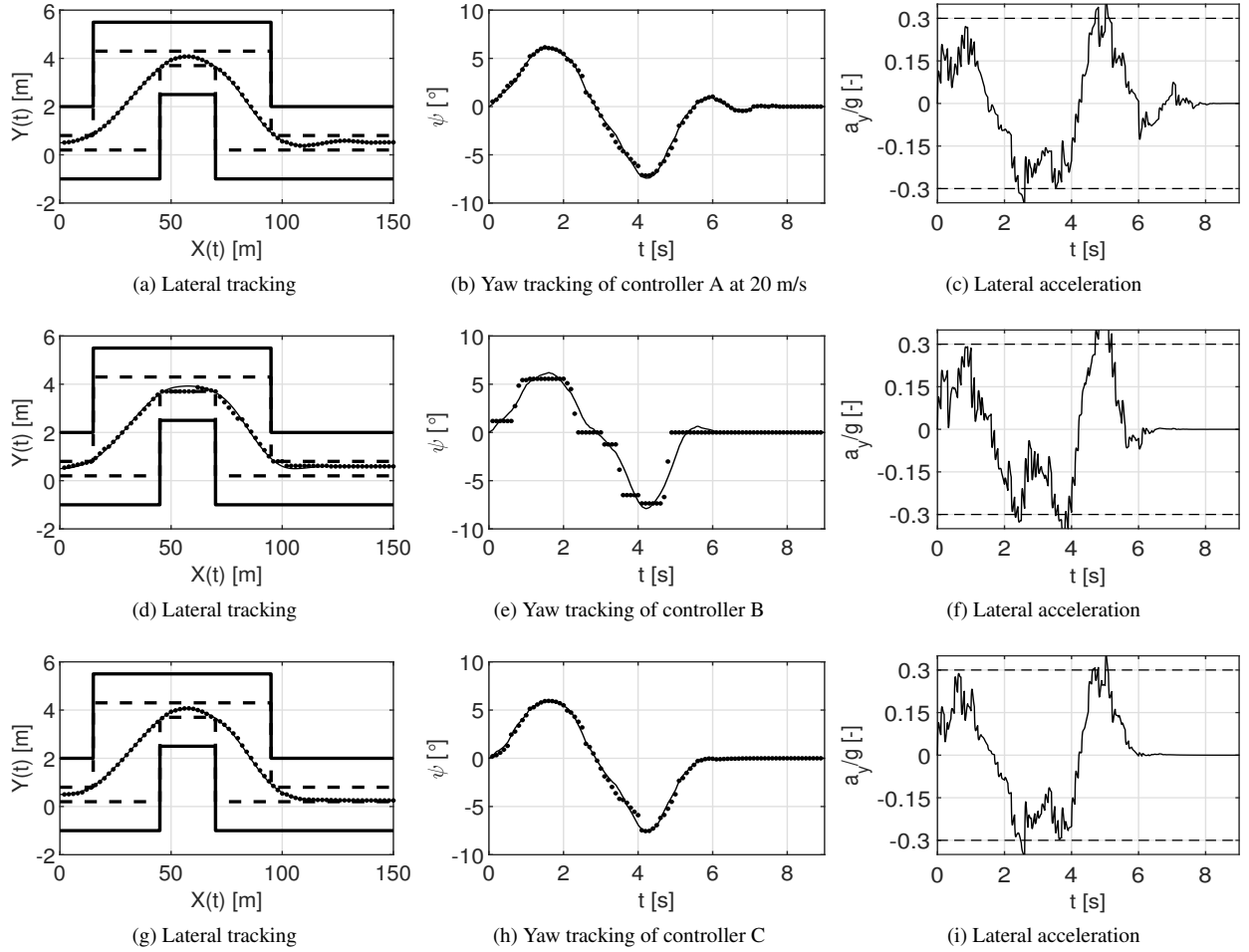
Fig. 8.

Using simple, fundamental information about the driving scenario, the controller generates optimal steering values that allowed the simulation vehicle to navigate the double-lane-change course. The controller achieves this by generating a reference trajectory in two stages: first generating a feasible geometric reference path between road bounds; this path is then passed to the path optimization module, which optimizes it to ensure that the path curvature does not exceed values that would cause high lateral acceleration values.

Once an optimal path has been calculated, it is passed to the final stage of the controller. In this, a nonlinear MPC scheme with a nonlinear model bicycle model as the system plant is used to calculate the optimal front steer-angle. The effectiveness of the controller was investigated via simulation during a double-lane-change maneuver, and the effectiveness of the POM was shown with controller A exhibiting a greater than four-times reduction in the root mean square lateral tracking error compared to controller B at 20 m/s. The performance was also shown to be comparable to a controller using a smooth initial reference trajectory, showing that the flexibility of the PGM does not compromise dynamic performance. The proposed controller is an example of the flexibility of this extended hierarchical control framework. The chosen implementations of the individual modules can be changed independently of one another, enabling greater flexibility in the range of applications for this framework.

Future work for this proposed framework includes terminal constraint sets in addition to adequate preview horizons to provide system stability. Investigating the capabilities of the controller during different driving scenarios should also be studied; particularly if the framework is extended to allow for a varying longitudinal velocity.

Nomenclature

a Acceleration, distance from front axle to center of mass
b Distance from rear axle to center of mass
m Vehicle mass
I Vehicle's yaw inertia
N Preview horizon in discrete steps
$\mathbf{p}$ Position point
s Displacement
T Prediction horizon in time domain
$\boldsymbol{u}$ Control vector
$\boldsymbol{\mathscr{u}}$ Control vector sequence
u Longitudinal velocity in body-fixed coordinate frame
v Lateral velocity in body-fixed coordinate frame

11

X	Longitudinal displacement in global coordinate frame
Y	Lateral displacement in global coordinate frame
z	State vector
α	Slip angle
δ	Steer angle
κ	Path curvature
σ	Tire relaxation length
ω_z	Yaw velocity
ψ	Yaw angle from global to local coordinate frame
ρ	Call ratio
ΔT	Call period: interval between controller calls
Δt	Discrete time-step

subscripts and superscripts

cf	Cornering force from Magic Tire Formula
lf	Longitudinal force from Magic Tire Formula
yf	Front Tire force acting in the lateral direction of the local coordinate frame
yr	Rear Tire force acting in the lateral direction of the local coordinate frame
lb	Lower bound
ub	Upper bound
f	Front Tire
r	Rear Tire, reference value for given variable
pg	Path generation
po	Path optimization
vc	Vehicle Control

References

[1] Gordon, T., and Lidberg, M., 2015. "Automated driving and autonomous functions on road vehicles". *Veh. Syst. Dyn.,* **53**(7), pp. 958–994.

[2] Huh, K., and Kim, J., 2001. "Active Steering Control Based on the Estimated Tire Forces". *J. Dyn. Syst. Meas. Control,* **123**(3), pp. 505–511.

[3] Mokhiamar, O., and Abe, M., 2005. "Simultaneous Optimal Distribution of Lateral and Longitudinal Tire Forces for the Model Following Control". *J. Dyn. Syst. Meas. Control,* **126**(4), mar, pp. 753–763.

[4] Cole, D. J., Pick, A. J., and Odhams, A. M. C., 2006. "Predictive and linear quadratic methods for potential application to modelling driver steering control". *Veh. Syst. Dyn.,* **44**(July 2015), pp. 259–284.

[5] Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D., 2007. "Predictive Active Steering Control for Autonomous Vehicle Systems". *566 IEEE Trans. Control Syst. Technol.,* **15**(3), pp. 566–580.

[6] Hrovat, D., Di Cairano, S., Tseng, H., and Kolmanovsky, I., 2012. "The development of Model Predictive Control in automotive industry: A survey". In 2012 IEEE Int. Conf. Control Appl., pp. 295–302.

[7] Li, H., Shi, Y., Yan, W., and Liu, F., 2018. "Receding horizon consensus of general linear multi-agent systems with input constraints: An inverse optimality approach". *Automatica,* **91**, pp. 10–16.

[8] Lars, G., and Panne, 2017. *Nonlinear model predictive control: theory and algorithms*, 2nd ed. Communications and control engineering. Cham.

[9] Allgöwer, F., Findeisen, R., and Nagy, Z. K., 2004. "Nonlinear model predictive control: From theory to application". *J. Chinese Inst. Chem. Eng.,* **35**(3), pp. 299–315.

[10] Wang, Y., and Boyd, S., 2010. "Fast Model Predictive Control Using Online Optimization". *IEEE Trans. Control Syst. Technol.,* **18**(2), pp. 267–278.

[11] Scattolini, R., 2009. "Architectures for distributed and hierarchical Model Predictive Control - A review". *J. Process Control,* **19**(5), pp. 723–731.

[12] Falcone, P., Borrelli, F., Tseng, H. E., Asgari, J., and Hrovat, D., 2008. "A hierarchical Model Predictive Control framework for autonomous ground vehicles". In 2008 Am. Control Conf., pp. 3719–3724.

[13] Chang, S., and Gordon, T. J., 2008. "A flexible hierarchical model-based control methodology for vehicle active safety systems". *Veh. Syst. Dyn.,* **46**(SUPPL.1), pp. 63–75.

[14] Ranney, T. A., 1994. "Models of Driving Behavior: A Review of Their Evolution". *Accid. Anal. Prev.,* **26**(6), pp. 733–750.

[15] MacAdam, C. C., 2003. "Understanding and Modeling the Human Driver". *Veh. Syst. Dyn.,* **40:1-3**(January 2014), pp. 101–134.

[16] Michon, J. A., 1976. "The Mutual Impacts of Transportation and Human Behaviour". In *Transp. Plan. a Better Environ.*, P. Stringer and H. Wenzel, eds. Springer US, Boston, MA, pp. 221–235.

[17] Michon, J. A., 1985. "A critical view of driver behavior models: what do we know, what should we do?". *Hum. Behav. traffic Saf.*, pp. 485–520.

[18] Joshi, A., and James, M. R., 2015. "Generation of accurate lane-level maps from coarse prior maps and lidar". *IEEE Intell. Transp. Syst. Mag.,* **7**(1), pp. 19–29.

[19] Toledo-Moreo R., R., Bétaille, D., and Peyret, F., 2010. "Lane level integrity provision for navigation and map-matching with GNSS, dead-reckoning and enhanced maps". *IEEE Trans. ITS,* **11**(1), pp. 100–112.

[20] MATLAB, 2019. *R2019a*. The MathWorks Inc.

[21] Pacejka, H., 2012. *Tire and Vehicle Dynamics*, 3rd ed. Butterworth-Heinemann, Oxford.

[22] Chen, W. H., Ballance, D. J., and O'Reilly, J., 2000. "Model predictive control of nonlinear systems : Computational burden and stability". *IEE Proc. - Control Theory Appl.,* **147**(4), pp. 387–394.

[23] Maciejowski, J. M., 2002. *Predictive Control: with Constraints*. Prentice Hall, London.

[24] Boyd, S., and Vandenberghe, L., 2009. *Convex Optimization*. Cambridge University Press, Cambridge.

[25] Lee, A. Y., 2002. "Coordinated Control of Steering and Anti-Roll Bars to Alter Vehicle Rollover Tendencies". *J. Dyn. Syst. Meas. Control,* **124**(1), pp. 127–132.