

Optimization-Based Path-Planning for Connected and non-Connected Automated Vehicles

Panagiotis Typaldos^{a,*}, Markos Papageorgiou^a and Ioannis Papamichail^a

^a*Dynamic Systems and Simulation Laboratory, School of Production Engineering & Management, Technical University of Crete, 73100, Chania, Greece*

ARTICLE INFO

Keywords:

Path Planning
Model Predictive Control
Automated Vehicles
Vehicle Trajectory Optimization
Connected Vehicles

ABSTRACT

A path-planning algorithm for connected and non-connected automated road vehicles on multilane motorways is derived from the opportune formulation of an optimal control problem. In this framework, the objective function to be minimized contains appropriate respective terms to reflect: the goals of vehicle advancement; passenger comfort; and avoidance of collisions with other vehicles and of road departures. Connectivity implies, within the present work, that connected vehicles can exchange with each other (V2V) real-time information about their last generated short-term path. For the numerical solution of the optimal control problem, an efficient feasible direction algorithm (FDA) is used. To ensure high-quality local minima, a simplified Dynamic Programming (DP) algorithm is also conceived to deliver the initial guess trajectory for the start of the FDA iterations. Thanks to very low computation times, the approach is readily executable within a model predictive control (MPC) framework. The proposed MPC-based approach is embedded within the Aimsun microsimulation platform, which enables the evaluation of a plethora of realistic vehicle driving and advancement scenarios under different vehicles mixes. Results obtained on a multilane motorway stretch indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to ordinary manually driven vehicles. Increased penetration rates of automated vehicles are found to increase the efficiency of the overall traffic flow, benefiting manual vehicles as well. Moreover, connected controlled vehicles appear to be more efficient in achieving their desired speed, compared also to the corresponding non-connected controlled vehicles, due to the improved real-time information and short-term prediction achieved via V2V communication.

1. Introduction

In the past decade, automated driving has attracted strong interest in industry and scientific community. This is fostered by strong technological advancements, compared to which human driving capabilities appear limited in terms of perception of the driving environment, reaction time, and real-time decision efficiency. In addition, variations in driving behavior from person to person or short inattention at high speeds may result in accidents. In fact, the vast majority of road accidents are attributed to human error on the account of, e.g., insufficient sensory information, lack of attention, shortcomings in driving skill or reckless driving. Each year, road accidents result in approximately 1.35 million fatalities and leave some 50 million of injured or disabled worldwide. Road congestion is another major issue, causing excessive delays, fuel consumption and emissions around the globe (Goniewicz et al., 2016; Montanaro et al., 2019). On the other hand, vehicle automation is a challenging area due to the variety and complexity of real-world environments, including avoidance of static and moving obstacles, compliance with traffic rules and consideration of human driving behavior aspects (Gu and Dolan, 2014).

With the recent advances in vehicle communications, either vehicle-to-vehicle (V2V) or infrastructure-to-vehicle (I2V), new communication channels, carrying a potentially wide range of information becomes available in real time. Connected automated vehicles (CAVs) may receive or exchange relevant information, including vehicle state, the current traffic conditions, the next switching time of a traffic signals etc. This extended information entails better knowledge of the driving conditions for CAVs and may be beneficial in various driving situations concerning road safety, flow efficiency and environmental sustainability (Sjoberg et al., 2017; Tian et al., 2018). Specifically, automated driving may improve various driving aspects, such as lane-changing, obstacle avoidance, forming of platoons with short inter-vehicle distances and the application of smoother acceleration or deceleration.

*Corresponding author
ORCID(s):

The development of fully automated driving algorithms is inherently related to planning and updating a vehicle path, which should be efficient, collision-free and user-acceptable. Planning such a path can be viewed as a trajectory generation problem, i.e., creation in real time of a quasi-continuous sequence of states that must be tracked by the vehicle via appropriate steering, throttle and braking actions. Inspired by earlier studies on motion planning of robot vehicles in other contexts (e.g. (González et al., 2015)) and driven by imminent implementations and optimistic forecasts of CAV technologies (e.g. (Nagy and Kelly, 2001; Gu and Hu, 2002)), studies on CAV trajectory optimization in the road traffic context have boomed in the past decade. However, path-planning for road vehicles is a difficult task, since speeds are very high, and safety of the passengers must be guaranteed. Thus, automation on roads calls for sophisticated approaches, such as optimal control methods, advanced feedback control or reinforcement learning (Clausmann et al., 2019; Haydari and Yilmaz, 2020).

In the related literature, many works exist for trajectory generation of longitudinal motion for CAVs, considering cooperative adaptive cruise control (CACC) and platooning (Dey et al., 2015; Wang et al., 2018b); cooperative merging at highway on-ramps (Rios-Torres and Malikopoulos, 2016; Ntousakis et al., 2016); speed harmonization on highways (Ghiasi et al., 2017; Malikopoulos et al., 2018) etc. On the other hand, there are relatively few studies on CAV trajectory optimization considering lane-changing (Wang et al., 2018c), which indicates that optimization of CAV trajectories in both longitudinal and lateral directions is worth additional investigations. Some of these studies approach the problem with optimal control and model predictive control (MPC), while avoidance of collision with other vehicles is handled through potential-field like penalty functions (Dixit et al., 2019; Rasekhipour et al., 2016; Wang et al., 2018a; Jalalmaab et al., 2015; Makantasis and Papageorgiou, 2018). In comparison with other aforementioned approaches, e.g. reinforcement learning, optimal control techniques require substantially less data as input. Moreover, depending on the numerical solution algorithms, the execution times of the problem should be sufficiently small to enable their implementation within a real-time MPC framework, i.e. real-time feasibility.

As far as the connectivity capabilities of the CAVs is concerned, most of the related mentioned works consider limited knowledge of their surroundings, specifically knowledge of the current position and speed of surrounding vehicles, e.g. thought their own sensors. Thus, the short-term future path of the obstacles is considered as a projection of their initial states, e.g. assuming zero acceleration and no lane change. For example, in (Sadat et al., 2020; Shao et al., 2021), the prediction of the surrounding environment is handled through marginal distributions of semantic occupancy over time; or considered as static region, respectively. In the same context, (Saxena et al., 2020) studies the problem of successfully executing a safe and comfortable merge into dense traffic, while the surrounding obstacle vehicles are handled either as static or as constant-speed obstacles. Note that a variety of path planning approaches that emerged in decade-long research in the Robotics domain are valuable for road vehicle path planning as well, but there is a crucial difference in the respective requirements due to automated highway vehicles (and dynamic obstacles) driving much faster than typically considered autonomous robots. This important difference calls for enhanced prediction of the obstacles behaviour, which is indeed achieved via vehicle connectivity, as considered in this work.

Compared to approaches presented in the past by some of the authors (Makantasis and Papageorgiou, 2018), the present work marks significant differences and advances. In that early-stage work, a basic optimal control problem (OCP) formulation was developed, providing a solid basis for subsequent significant advances, both in the conceptual design and in its evaluation. Specifically, significant differences in the present work include: the newly designed collision avoidance term, the number of involved state variables, the prediction of the surrounding vehicles' paths and the comprehensive and realistic evaluation at traffic level, to mention just the most important ones. In more detail, in the previous work, the collision avoidance term was formulated as a non-smooth function, which may entail difficulties for gradient-based numerical solution algorithms, such as the employed feasible direction algorithm (FDA). Therefore, in this work, the corresponding term has been modified, and a novel smooth function is appropriately designed. In addition, the longitudinal control variable in this work is the jerk, instead of the acceleration, which ensures smoother trajectories and improves the passengers' comfort. Moreover, the specific previous work did not consider connectivity for the vehicles, which implies that the prediction of the surrounding vehicle paths has limited accuracy, and the impact of vehicle connectivity cannot be assessed. Finally, compared to the previous work, which only reports on limited open-loop evaluation instances involving just a few vehicles, without testing its applicability within a closed-loop MPC framework, this work develops the proposed approach fully, along with the appropriate MPC framework; and implements it within a realistic microsimulation platform (Aimsun Next, 2019) in order to comprehensively evaluate the traffic-level effects of the controlled AVs at various demand levels and for different penetration rates of vehicle automation and connectivity.

Compared to other works in the literature, the proposed approach delivers a very efficient algorithm, which is

readily applicable in real time due to very low computational times (fractions of a second). Moreover, the introduction of the enhanced information of the CAVs enables more accurate prediction of their surrounding and is demonstrated to be crucial, particularly in scenarios of high demand levels. Finally, based on the authors' knowledge, there are no comparable works considering such a comprehensive OCP and MPC-based approach within a microscopic simulator for various demand levels and, especially, for various penetration rates; and reporting on the effects of the AVs on their own efficiency and on the traffic environment, as the penetration rate rises.

In summary, the main contributions of this work are the following:

- A highly efficient (fractions of a second) numerical approach for the solution of the OCP for path planning of AVs.
- The full exploitation of vehicle connectivity via asynchronous path decision exchange that leads to more accurate prediction of the surrounding CAV behaviour.
- Comprehensive traffic-level evaluation and analysis, including the impact of lane changes, for different penetration rates and demand levels.

In more detail, this paper explores the impact of vehicle connectivity on the efficiency of advancement of automated vehicles (AVs) and on the efficiency of the emerging traffic flow. The investigations address multi-lane motorways with mixed traffic, comprising both automated and manually driven vehicles at different penetration rates. An optimal control problem is employed for the path-planning of AVs, comprising three main elements:

- A simple kinematic model describing the vehicle movement process.
- An objective function to be minimized, which contains respective terms to reflect efficient vehicle advancement; passenger comfort and fuel consumption; and avoidance of collisions with other vehicles and of road departures.
- Short-term prediction of the trajectories of other neighboring vehicles (obstacles), which is crucial for pro-active collision avoidance.

For the numerical solution of the optimal control problem (OCP), a very efficient iterative feasible direction algorithm (FDA) is used. To ensure high-quality local minima, a simplified Dynamic Programming (DP) algorithm is also employed to deliver the initial guess trajectory for the FDA. Thanks to low computation times, the overall approach is readily executable within a MPC framework, applying updated initial state and obstacle path prediction at each repetition. MPC has a long history of applications in control, automation, and chemical industries (Mayne and Michalska, 1988; Qin and Badgwell, 2003; Mayne, 2014). Due to the strongly dynamic environment of the path-planning problem for AVs, the accurate prediction of the surrounding vehicles' paths is crucial, particularly at high highway speeds. However, the uncertainties related to the changing environment and to the actual vehicle advancement inevitably increase over time. Therefore, the MPC approach is often used for online path planning of AVs, as it is very efficient in handling the arising uncertainties due to its re-planning ability (Claussmann et al., 2019). One possibility is to solve the OCP at each time step, with a corresponding shift of the planning horizon (Nilsson et al., 2015; Murillo et al., 2018; Howard et al., 2010). An alternative possibility is to update path decisions in event-based mode (Earl and D'Andrea, 2007; Khazaeni and Cassandras, 2016), which is pursued in the current study.

Vehicle connectivity refers here to the capability of AVs to exchange with each other, in an asynchronous mode, real-time information about their current state (position and speed) and their latest generated path, where the latter strongly enhances the prediction accuracy for obstacle movement in the short-term future. The MPC-based approach is embedded within the Aimsun micro-simulation platform (Aimsun Next, 2019), which enables driving evaluation in countless appearing driving scenarios. Specifically, we investigate two cases, each of them at different penetrations of AVs:

- No connectivity: Each AV is aware of the current position and speed of obstacles (via its own sensors). Short-term prediction of obstacle movement is based on extrapolation, assuming zero acceleration and same-lane movement.
- Connected automated vehicles: Each AV is aware of the current position and speed of obstacles; in addition, it receives the path-planning decisions of other automated vehicles, which facilitates the short-term prediction of their movement.

In both cases, the short-term movement prediction for manually driven vehicles is based on zero-acceleration and same-lane extrapolation.

Demonstration results are reported for a motorway pipeline section. The results indicate higher efficiency of the optimally controlled vehicles in driving closer to their desired speed, compared to non-automated vehicles. In addition, increasing penetration rates of controlled vehicles are found to lead to more efficient traffic flow. Specifically, as the penetration rate of controlled vehicles rises, there is a significant increase of the average speed and, consequently, a decrease on the average delay time and travel time for all vehicles. These conclusions apply mainly for demand levels where the controlled vehicles have space to manoeuvre. Specifically, the proposed approach is efficient due to the fact that the controlled vehicles are able to apply smarter manoeuvres. Also, connected controlled vehicles are more efficient than non-connected controlled vehicles, due to the improved real-time information that enables more pertinent obstacle movement prediction. Finally, the emerging traffic flow and even the manually driven vehicles are found to benefit from improved operation of the AVs.

It should be noted that urban road environments are out of the scope of this work, which focuses on vehicles moving on a highway. We believe that application of a similar approach for urban networks is possible with appropriate modifications or extensions, but this is left to future work.

The rest of the paper is organized as follows: Section 2 presents the dynamics of each AV and the components of the objective function that lead to the OCP definition. Section 3, describes the numerical solution algorithms and explains the procedure used for MPC. Section 4 presents the simulation results, while Section 5 concludes this work.

2. Optimal Control Problem Formulation

This section describes the proposed path-planning strategy for automated road vehicles on motorways. At first, the simple vehicle kinematic motion dynamics and considered hard constraints are defined. Then, an objective function is designed, which includes appropriate terms regarding the efficient vehicle advancement, the obstacle and off-road avoidance and the passenger convenience. Finally, the proposed path-planning algorithm is generated based on a combination of FDA and Dynamic Programming techniques.

2.1. Problem variables, state-equations and constraints

We consider a straight road on a two-dimensional plane, and the vehicle's position on this plane is expressed in global Euclidean coordinates. Each controlled vehicle is described by five state equations, corresponding to the equations of motion, which are expressed in discrete time, assuming time-steps of length T , as follows:

$$x(k+1) = x(k) + v_x(k) \cdot T + \frac{1}{2}a_x(k) \cdot T^2 + \frac{1}{6}j_x(k) \cdot T^3 \quad (1)$$

$$y(k+1) = y(k) + v_y(k) \cdot T + \frac{1}{2}a_y(k) \cdot T^2 \quad (2)$$

$$v_x(k+1) = v_x(k) + a_x(k) \cdot T + \frac{1}{2}j_x(k) \cdot T^2 \quad (3)$$

$$v_y(k+1) = v_y(k) + a_y(k) \cdot T \quad (4)$$

$$a_x(k+1) = a_x(k) + j_x(k) \cdot T \quad (5)$$

where $x(k)$, $y(k)$, $v_x(k)$, $v_y(k)$, $a_x(k)$ correspond to the longitudinal and lateral position, the longitudinal and lateral speed and the longitudinal acceleration at time-step k , respectively; while the control variables $j_x(k)$, $a_y(k)$ refer to the longitudinal jerk and the lateral acceleration, respectively, which are kept constant for the duration T of each time-step k ; hence the above state equations are derived from the exact time-integration of the corresponding continuous-time differential equations of motion. Note that the consideration of jerk (rather than the acceleration) as a control variable for the longitudinal direction leads to smoother vehicle trajectories, which consequently improve the convenience of the vehicle passengers. On the other hand, for the lateral movement, such detail is not necessary, as the lateral speed and movement is only needed for lane changing, and a lane change maneuver is substantially less frequent compared to the continuous longitudinal motion. It should be noted that, in contrast to other works, the proposed approach produces continuous, rather than discrete, lateral movement, thus avoiding the usage of integer variables that increase the computational effort for the numerical OCP solution.

In this work, the model for expressing the vehicle dynamics is a double (lateral) and triple (longitudinal) integration model, which is quite common for the problem at hand, involving high-speed longitudinal movement with occasional lane changes. This model considers the lateral and longitudinal movement to be decoupled. An alternative would be the use of a so-called bicycle model (Rajamani, 2011). However, in a lane-based highway environment, a vehicle mostly follows a leader or drives on its lane with its desired speed, as the lane-change manoeuvres are not very frequent. In these conditions, explicit consideration of steering is not deemed important.

The control variables $j_x(k)$, $a_y(k)$ are bounded according to vehicle specifications, whereby the longitudinal upper bound and the lateral bounds are constant, while the longitudinal lower bound is state-dependent, as follows:

$$j_{x,\min}(v_x(k), a_x(k)) \leq j_x(k) \leq j_{x,\max} \quad (6)$$

$$a_{y,\min} \leq a_y(k) \leq a_{y,\max} \quad (7)$$

The constant upper bound $j_{x,\max}$ and the lateral bounds $a_{y,\min}$, $a_{y,\max}$ may be set with consideration of the vehicle capabilities and the passengers' convenience. Furthermore, a vehicle, having a currently non-negative longitudinal speed $v_x(k)$, should not have negative longitudinal speed at the next time step, i.e.,

$$v_x(k+1) \geq 0 \quad (8)$$

should hold. To ensure this, while avoiding state constraints that may complicate the numerical solution of the OCP, the state equation (3) is replaced in (8) and, after rearrangement, this yields the state-dependent control constraint

$$j_x(k) \geq -\frac{2}{T^2}v_x(k) - \frac{2}{T}a_x(k) \quad (9)$$

This bound may be unrealistically low (negative) at higher speeds, due to the magnitude of the coefficients, which can be a cause of discomfort for the passengers. Note that the equation-version of (9) may be interpreted as a dead-beat controller that drives the speed v_x to zero in exactly two time steps. However, the magnitude of the resulting jerk bounds can be mitigated by choosing more moderate "controller" coefficients $0 < K_1 \leq 2/T^2$ and $0 < K_2 \leq 2/T$, which would drive v_x to zero asymptotically. This way, accordingly moderate lower jerk bounds are obtained, while guaranteeing that the constraint (8) is always satisfied. In conclusion, the considered lower bound on longitudinal jerk is

$$j_{x,\min}(v_x(k), a_x(k)) = -K_1 \cdot v_x(k) - K_2 \cdot a_x(k) \quad (10)$$

and (6) is replaced by the following state-dependent constraint to be considered in OCP

$$h = [j_x(k) - j_{x,\max}][j_x(k) - j_{x,\min}] \leq 0 \quad (11)$$

2.2. Optimization Objectives

2.2.1. Objective Function

The objective function to be minimized, in the frame of the OCP formulation, includes a number of terms, which consider efficient, convenient and safe driving, each with a weighting parameter to reflect the corresponding priorities. Among these terms, several safety constraints are considered as soft constraints, so as to drastically speed up the numerical solution. Although alternative approaches may indeed lead to respect of the safety constraints in a more direct way, our indirect approach (using soft constraints) also satisfies these constraints and fulfils this task to the full, while featuring very low CPU times, which make it readily applicable in real time.

The optimisation criterion, which extends over a time horizon of K steps in the future, reads

$$J = \sum_{k=0}^{K-1} \left[w_1 j_x^2(k) + w_2 a_x^2(k) + w_3 a_y^2(k) + w_4 [v_x(k) - v_{d,x}]^2 \right. \\ \left. + w_5 [v_y(k) - v_{d,y}]^2 + w_6 f_r(y(k)) + w_7 \sum_{i=1}^n [c_i(x(k), y(k))] \right] \quad (12)$$

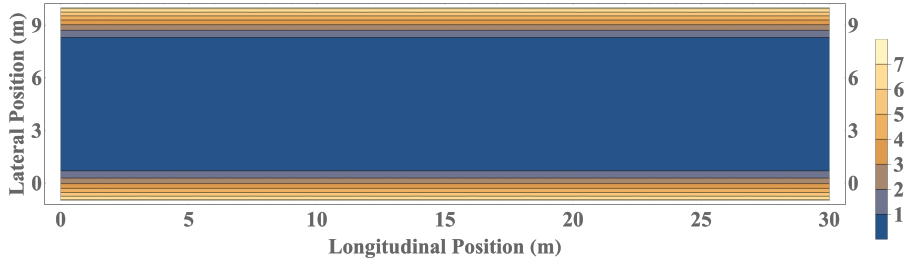


Figure 1: The penalty function of road boundaries.

where w_1, \dots, w_7 are non-negative penalty parameters.

The first three quadratic penalty terms concern the comfort of the passengers, which is related to the magnitude of lateral and longitudinal acceleration, as well as of the longitudinal jerk. Note that the quadratic penalty term of longitudinal acceleration acts also as a good proxy for deriving fuel-minimizing vehicle trajectories (Typaldos et al., 2020). More specifically, it has been demonstrated that the simple square-of-acceleration term delivers excellent approximation of fuel-optimal vehicle trajectories, when compared with the use of complex and realistic fuel-consumption models in the objective function.

The fourth and fifth penalty terms reflect the vehicle advancing goals. These terms account for pre-specified desired longitudinal and lateral speeds by penalizing speed deviations from those values. In the current work, $v_{d,x}$ has a positive value, corresponding to the vehicle type or driver choice of desired longitudinal speed; while $v_{d,y}$ is set to zero to suppress unnecessary lateral movements. Note that, $v_{d,y}$ can be set to non-zero values in case, for example, a vehicle is bound to exit the motorway at a downstream off-ramp or in emergency cases, where the controlled vehicles should create appropriate space, e.g., for an ambulance or a fire truck to pass.

The two last penalty terms, which are related to the lateral road boundaries and the obstacle avoidance, respectively, are described in detail in the following.

2.2.2. Road Boundaries Term

The penalty function for the road boundaries must be designed so as to disallow the automated vehicle (AV) to depart from the road. To this end, the following smooth quadratic function, dependent on the width w of the road, is adopted, which is equal to zero when the vehicle moves within the road boundaries; while its value increases the more the vehicle would depart from the boundaries

$$f_r = \begin{cases} (y - d)^2 & , y < d \\ 0 & , d \leq y \leq w - d \\ (y + d - w)^2 & , y > w - d \end{cases} \quad (13)$$

where the small positive parameter d is used to prevent the ego vehicle from moving too close to the road boundaries; and w is the road width. Figure 1 displays the penalty function generated for the road boundaries for a three-lane motorway with $w = 9$ m and $d = 1.5$ m.

2.2.3. Collision Avoidance Term

Crash avoidance is an absolute necessity in vehicle path planning, that may be achieved in different ways. Note that crash avoidance is closely related to the highest deceleration available, not just the way (hard or soft) of considering the related constraints. Thus, it is crucial that deceleration vis-à-vis an obstacle is applied timely, so as to avoid the need for harsh or infeasible deceleration eventually; and this is indeed a feature enabled via OCP with future time horizon (that helps anticipate imminent crash risks) and soft constraints (that exert timely smooth repulsion while the controlled vehicle is approaching obstacles).

The penalty function (similar to potential field functions) for obstacle avoidance should feature high values at the gross obstacle space, so that the ego vehicle is repulsed, and potentially unsafe trajectories are suppressed; and low (or virtually vanishing) values outside of that space. To this end, we adopt, for each obstacle, an ellipsoid penalty function,

which is best understood in its one-dimensional form $f(x) = (1 + (x/a)^{p_1})^{-1}$. This smooth function has a maximum of 1 for $x = 0$; for $|x| < a$, the function retains values close to 1 (the maximum), while for $|x| > a$, it reduces to very small values or virtually zero. The even integer parameter p_1 influences the sharpness of the smooth transition of function values from 1 (for $|x| < a$) and 0 (for $|x| > a$).

For the present needs, the function is generalised to two dimensions, and its two respective arguments δ_x and δ_y reflect the longitudinal and lateral distances between the ego vehicle and each obstacle i , while the respective counterparts of parameter a should be selected based on the vehicle dimensions, taking into account also the vehicle and obstacle speeds for safe car-following distances. Thus, in two dimensions, the following ellipsoid penalty function for each obstacle i is used

$$c_i(x, y) = \frac{1}{\left(\frac{\delta_x + s}{0.5 \cdot r_x}\right)^{p_1} + \left(\frac{\delta_y}{0.5 \cdot r_y}\right)^{p_2} + 1} \quad (14)$$

where p_1 and p_2 are positive even integers, $\delta_x = x - x_i$ and $\delta_y = y - y_i$ are the longitudinal and lateral distances from obstacle i , r_x and r_y determine the dimensions of the ellipse, and the term s introduces a shift of the ellipse's longitudinal centre position, as will be explained in the following.

For safety, the width of the ellipse, r_y , should be equal to the width of a motorway lane, W , to avoid lateral intrusion of the ego vehicle into a lane occupied by an obstacle. On the other hand, the length of the ellipse, r_x , should consider, beyond the physical vehicle length, a safe space gap in front and behind of each obstacle i . In case the ego vehicle is behind the obstacle, a safe space-gap, equal to $\omega \cdot v_x$, should be maintained between the two vehicles, where ω is the time-gap parameter (Rajamani, 2011). On the other hand, when the ego vehicle is in front of the obstacle, a safe space-gap, equal to $\omega \cdot v_i$, should be considered, with v_i being the longitudinal speed of the obstacle. Moreover, the physical dimensions of both ego and obstacle vehicles, $L_i = (\mu_e + \mu_{o,i})/2$, with $\mu_e, \mu_{o,i}$ being the ego and obstacle's i lengths, respectively, should be considered as a minimum safe space-gap in case of zero speeds. Thus, the longitudinal ellipsoid dimension is defined as

$$r_x = \omega \cdot v_x + \omega \cdot v_i + L_i \quad (15)$$

Due to the difference in the longitudinal speeds of the ego vehicle and the obstacle, the above space gaps in front of and behind the obstacle are accordingly different, hence the ellipse is longitudinally asymmetric with respect to the physical centre of the obstacle. Therefore, the ellipse centre must be appropriately shifted, depending on ego vehicle and obstacle speeds difference. Specifically, the position of the ellipse centre is expressed as the summation of δ_x and the shift term s given by

$$s = \frac{\omega \cdot (v_x - v_i)}{2} \quad (16)$$

For safe car-following, the longitudinal size of the ellipse depends on the controlled vehicle's and obstacle's speeds, as well as the time-gap value employed. Thus, the size is not fixed, but changes over time according to the mentioned variables. For example, when the vehicles are stopped (zero speed), the ellipse's longitudinal dimension equals the average length of the controlled vehicle and the leading vehicle, in both sides of the ellipse; while, as the speed of both vehicles increases, the dimension of the ellipse increases accordingly.

In Figures 2a-2g, different cases for the proposed ellipsoid are displayed for illustration. We assume an obstacle with its centre point at 50 m. The ellipsoids in each case are displayed in orange color and are contrasted to the corresponding space gaps resulting from the constant time-gap policy (red rectangle) (Rajamani, 2011). In Figure 2a, both the ego vehicle and the obstacle have zero speed, leading to a symmetric ellipse shape around the obstacle centre, as both the front and the back part around the vehicle centre (dashed black line) are equal to L_i . Note that the shape of the ellipse is close to a rectangle due to the use of high values for both exponents p_1 and p_2 in (8) (e.g. $p_1 = p_2 = 18$). In Figures 2b-2g, using a time-gap value $\omega = 1.2$ s and a longitudinal obstacle speed $v_i = 20$ m/s, it is observed how the ellipse shape changes, and the ellipse centre shifts in dependence of the ego vehicle speed. Specifically, when the ego vehicle has zero speed (2b), the centre of the ellipse is shifted to upstream to accommodate a back space-gap equal to the average length of both vehicles; while the front part of the ellipse equals the space-gap, which depends on the

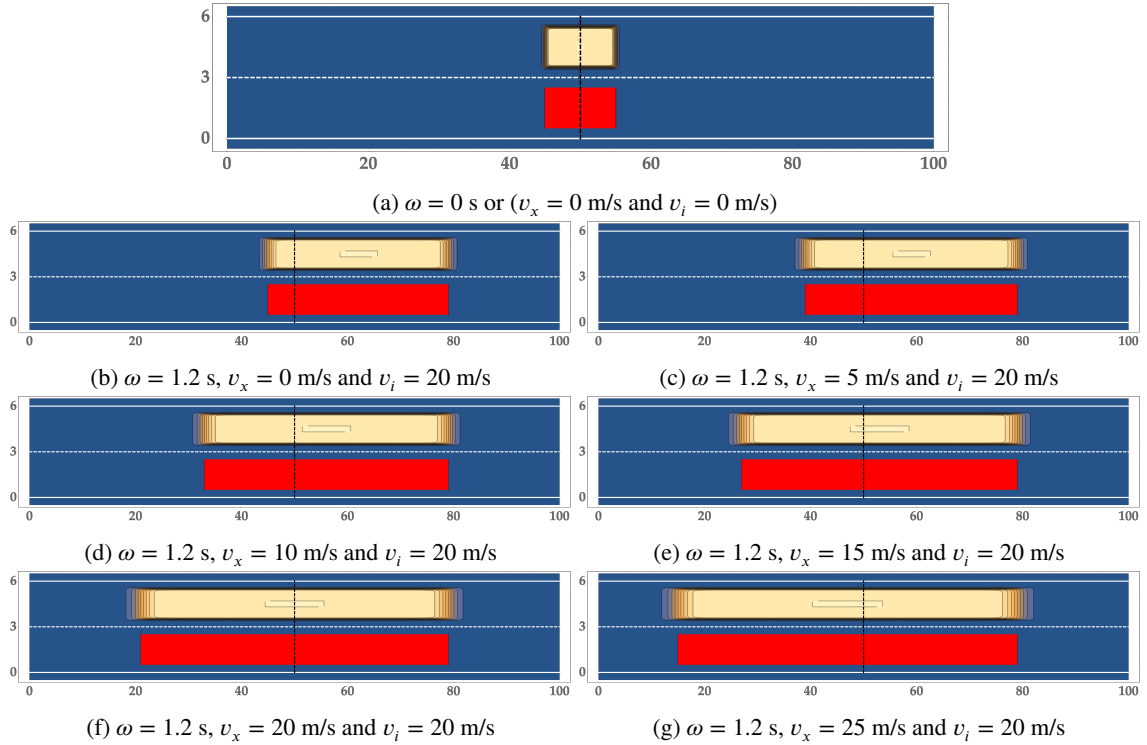


Figure 2: Illustration of the collision avoidance term (orange ellipsoid), $c_i(x, y)$, for different ego vehicle speeds with time gap $\omega = 1.2$ s and obstacle speed $v_i = 20$ m/s, compared to the corresponding space gap of the constant time gap policy (red rectangle). The lateral dashed line indicates the physical vehicle centre location.

speed of the obstacle. For higher ego vehicle speeds, it can be observed that the length of the ellipse increases, and, at the same time, its centre shifts downstream. In the specific representation, the length of the front part of the ellipse does not change, as the obstacle speed remains constant.

Potential obstacles, to be considered in the sum of the collision avoidance term in (12), are all vehicles around the ego vehicle that might appear on its way, causing potentially a crash risk, during the considered planning horizon. Thus, all vehicles within a longitudinal zone in front and behind the ego vehicle should be considered as potential obstacles, and the length of this zone is taken proportional to the ego vehicle's desired longitudinal speed times the time horizon.

2.3. Problem Formulation

In conclusion, the path-planning problem may be formulated as OCP. The difference equations (1)-(5) may be organised in the following vector form

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k], \quad k = 0, \dots, K-1 \quad (17)$$

where \mathbf{x} and \mathbf{u} are the system states and control variable vectors, respectively. With known initial state $\mathbf{x}(0) = \mathbf{x}_0$ and obstacle trajectories, the optimal control problem consists in minimizing the objective function (12) subject to (17) and control bounds

$$\mathbf{h} = [\mathbf{u}(k) - \mathbf{u}_{\max}][\mathbf{u}(k) - \mathbf{u}_{\min}] \leq \mathbf{0}, \quad k = 0, \dots, K-1 \quad (18)$$

which reflect constraints (7) and (11) in a unified way.

The general form of the objective function (12) is given by

$$J = \sum_{k=0}^{K-1} \phi[\mathbf{x}(k), \mathbf{u}(k)] \quad (19)$$

and then, the Hamiltonian function is given by (see (Papageorgiou et al., 2016, 2015))

$$H[\mathbf{x}(k), \mathbf{u}(k), \lambda(k+1), k] = \phi[\mathbf{x}(k), \mathbf{u}(k)] + \lambda(k+1)^T f[\mathbf{x}(k), \mathbf{u}(k), k] + \boldsymbol{\mu}(k)^T \mathbf{h}[\mathbf{x}(k), \mathbf{u}(k)] \quad (20)$$

where $\lambda(k+1)$ stands for the co-state vector (equivalent to Lagrange multipliers) for the corresponding state equations.

The following conditions of optimality for the discrete-time OCP must be satisfied for $k = 0, \dots, K-1$ (notation: $z_w = \partial z / \partial w$)

$$\mathbf{x}(k+1) = H_{\lambda(k+1)} = f[\mathbf{x}(k), \mathbf{u}(k), k] \quad (21a)$$

$$\lambda(k) = H_{\mathbf{x}(k)} = \phi_{\mathbf{x}(k)} + f_{\mathbf{x}(k)}^T \lambda(k+1) + h_{\mathbf{x}(k)}^T \boldsymbol{\mu}(k) \quad (21b)$$

$$H_{\mathbf{u}(k)} = \phi_{\mathbf{u}(k)} + f_{\mathbf{u}(k)}^T \lambda(k+1) + h_{\mathbf{u}(k)}^T \boldsymbol{\mu}(k) = 0 \quad (21c)$$

Equations (21a) and (21b) are the state and co-state difference equations, respectively, while (21c) specifies the optimal control variables. Furthermore, the following boundary conditions must be satisfied

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (22a)$$

$$\lambda(K) = \mathbf{0}. \quad (22b)$$

Although expressing a dynamical process, the above minimization problem is, from a mathematical point of view, a Nonlinear Programming Problem (NLP) due to the discrete-time nature of the involved process model, see ((Papageorgiou et al., 2016), (Papageorgiou et al., 2015)). However, thanks to the structure of the state equation (17), which allows for the state variables to be efficiently eliminated as functions of the control variables, the optimisation problem may be solved, by use of reduced gradients, in the reduced space of the control variables much more efficiently, compared to a general NLP problem with full dimension.

3. Numerical Solution and Model Predictive Control

3.1. Feasible Direction Algorithm (FDA)

The solution of the formulated OCP is computed by use of the very efficient feasible direction algorithm (FDA) (Papageorgiou et al., 2016; Typaldos et al., 2020), which exploits the structure of the state equations to map the OCP into an NLP problem in the reduced space of control variables. Thus, the algorithm attempts the calculation of a control trajectory $\mathbf{u}(k), k = 0, \dots, K-1$, which corresponds to a local minimum of the cost function, in the mK -dimensional space, where m is the number of control variables. This marks a substantial reduction of the problem dimension, as the state variables are eliminated. More specifically, FDA exploits the fact that $g(k) = [\frac{\partial f}{\partial \mathbf{u}(k)}]^T \lambda(k+1) + \frac{\partial \phi}{\partial \mathbf{u}(k)}$ equals the reduced gradient in the mK -dimensional reduced space of the control, if the states and co-states involved in the partial derivatives satisfy the state and co-state equations. The details about this fact may be read in optimal control textbooks (e.g. (Papageorgiou et al., 2015)).

The algorithm is iterative, starting with an initial-guess feasible control trajectory; feasible meaning that it satisfies all state equations and control inequality constraints. The multipliers of the inequality constraints that define bounds are calculated, using (21c), as $\boldsymbol{\mu}(k) = -\mathbf{g}(k)/[\partial \mathbf{h}/\partial \mathbf{u}(k)]$ for active constraints and $\boldsymbol{\mu}(k) = 0$ for inactive constraints. At each iteration, using reduced gradient information, a descent search direction in the (reduced) mK -dimensional control space is calculated based on conjugate gradients (or quasi-Newton). Subsequently, a line-search procedure delivers the optimal step along the search direction, and this optimal step leads to an enhanced feasible control trajectory,

with improved objective function value. This enhanced trajectory is fed to the next iteration; and so forth, until a sufficiently low reduced-gradient norm is obtained, which marks convergence to a virtually optimal control trajectory. The algorithm guarantees improved objective values at each iteration and features global convergence, from any starting control trajectory, to a local minimum.

The algorithmic steps of FDA are presented below (superscripts (i) indicate the iteration index);

- 1: Receive initial state values.
- 2: Guess an initial control trajectory $\mathbf{u}^{(0)}(k), k = 0, \dots, K - 1$
- 3: Calculate states $\mathbf{x}^{(0)}(k)$
- 4: In a unique loop: calculate $\mathbf{g}^{(0)}(k), \boldsymbol{\mu}^{(0)}(k)$ and co-states $\boldsymbol{\lambda}^{(0)}(k)$ for $k = K - 1, \dots, 0$, starting with $\boldsymbol{\lambda}^{(0)}(K)$.
- 5: Set iteration index $i = 0$.
- 6: **while** $i < \text{max_iterations}$ **do**
- 7: Specify a descent direction $\mathbf{s}^{(i)}(k), k = 0, \dots, K - 1$.
- 8: Specify an optimal scalar step $\xi^{(i)} > 0$ through line optimization.
- 9: Calculate $\mathbf{u}^{(i+1)}(k) = \mathbf{u}^{(i)}(k) + \xi^{(i)} \mathbf{s}^{(i)}(k), k = 0, \dots, K - 1$ and $\mathbf{x}^{(i+1)}(k)$ for $k = 0, \dots, K - 1$ (apply bounds on control)
- 10: In a unique loop: calculate $\mathbf{g}^{(i+1)}(k), \boldsymbol{\mu}^{(i+1)}(k)$ and co-states $\boldsymbol{\lambda}^{(i+1)}(k)$ for $k = K - 1, \dots, 0$, starting with $\boldsymbol{\lambda}^{(i+1)}(K)$.
- 11: Calculate projected gradient when the control bounds are applied.
- 12: **if** not converged **then**
- 13: index increment, $i := i + 1$
- 14: **continue**
- 15: **else**
- 16: **break**
- 17: **end if**
- 18: **end while**
- 19: Generate optimal control input $\mathbf{u}(k), k = 0, \dots, K - 1$

3.2. Dynamic Programming (DP)

As the OCP at hand is non-convex, FDA may converge to a local minimum. Note that, each single-lane vehicle path can give raise to a local minimum, even though lane changing could possibly improve the cost function. More precisely, gradient-based algorithms search, in each iteration, for descent directions for the control variables. However, in the specific problem, due to the penalisation of lateral acceleration and the rectangular shape of the ellipse in the collision avoidance term (14), a controlled path on a lane may feature positive gradients w.r.t. the lateral control, i.e. any lateral deviation from the current path would locally increase the objective value. Thus, if there is no lateral movement in the initial control trajectory, the controlled vehicle will remain in the same lane for the whole time horizon (local minimum), e.g. trapped behind an obstacle on the same lane.

In order to overcome this difficulty and to enhance the quality of the optimal paths delivered by the gradient-based FDA, Dynamic Programming (DP) is first used appropriately to deliver an initial guess trajectory. Despite the involvement of DP in an initial problem solution phase, the overall required computation time for the optimal path generation remains extremely low and certainly real-time feasible. The DP methodology is known to deliver a globally optimal solution trajectory for optimal control problems. However, DP is characterized by computational cost, which increases exponentially with the problem dimensions; and this cost is indeed too high in the present application for efficient real-time path-planning. However, DP is not considered here stand-alone, as it is not real-time feasible for the original path-planning problem. Instead, a simplified problem is solved through DP in very short time, in order to be used as a good initial guess for FDA. Using this combination of methods, the weaknesses of both, i.e. FDA leading to bad local minima and DP being infeasible in real time due to the curse of dimensionality, are strongly mitigated, resulting in good vehicle paths within very short computation times.

In particular, such rough, but globally optimal solutions of the DP simplified problem were found to include lane changes in driving situations where a lane change is indeed beneficial. Note again that the DP algorithm is only used to deliver the initial trajectory for the FDA. Therefore, sacrificing some accuracy due to the simplifications is not important, as the FDA has the final responsibility to deliver an optimal path that satisfies all vehicle tasks and constraints.

The simplifications introduced in the DP problem are:

- A larger step size T (1 s) is used, versus the four times smaller time step (0.25 s) used in FDA and deemed appropriate for path-planning of AVs. As a consequence, the number K of time steps within each planning horizon decreases accordingly.
- The state equation (5) is dropped, and the longitudinal acceleration a_x is used as a control variable instead of the jerk.
- The lateral vehicle motion is also simplified by dropping state equation (4) and modifying equation (2) assuming that the vehicle's lateral position is discrete and lane-based. Specifically, the lateral control is limited to three distinct values at each time step, namely $\{-1, 0, 1\}$, meaning that the vehicle can only apply a lane change towards the adjacent left or right lanes or stay on the same lane.
- Only one lane change is allowed at each planning horizon, in order to reduce the amount of options to be explored by DP.
- The longitudinal acceleration is also roughly discretised and may obtain one out of three values, namely $\{-3, 0, 3\}$ m/s².

Given the above modifications and the fact that state constraints may be directly handled by the DP algorithm, the cost function used with DP is simpler than the corresponding cost function of the OCP and is described as follows

$$J_{DP} = \sum_{k=0}^{K-1} [w_1 a_x^2(k) + w_2 u_y^2(k) + w_3 (v_x(k) - v_d)^2] \quad (23)$$

where a_x and u_y are the control variables, corresponding to the longitudinal acceleration and the discrete lateral movement, which reflects the lane changes. Collision avoidance, road departure avoidance and suppression of negative speeds are taken care by the DP algorithm directly, as will be discussed later.

In more detail, the vehicle kinematics (with bigger time step T) in the simplified DP problem are considered as follows

$$x(k+1) = x(k) + v_x(k)T + \frac{1}{2}a_x(k)T^2 \quad (24)$$

$$y(k+1) = y(k) + u_y(k) \quad (25)$$

$$v_x(k+1) = v_x(k) + a_x(k)T \quad (26)$$

where $x(k)$, $y(k)$, $v_x(k)$ correspond to the state variables of the simplified problem and reflect the vehicle's longitudinal position, lane and speed at discrete times $k = 0, \dots, K-1$, respectively. The control variables a_x, u_y reflect the longitudinal acceleration and lateral change of lane, as mentioned. The state and control variables are bounded within the following feasible regions

$$\mathbf{x}(k) \in \mathbf{X} \quad (27)$$

$$\mathbf{u}(k) \in \mathbf{U} \quad (28)$$

where the admissible state region \mathbf{X} does not allow for road departures, vehicle interference or negative longitudinal speeds; while the admissible control region \mathbf{U} contains the mentioned discrete values of the control variables in the simplified problem, i.e. $a_x \in \{-3, 0, 3\}$ m/s² and $u_y \in \{-1, 0, 1\}$.

After discretisation of the longitudinal position, speed and acceleration with consistent respective increments, the standard discrete DP algorithm is designed as follows. The algorithm starts with time step $K-1$ and advances backward, step-by-step. At every step k , all discrete states are branched into all possible transitions (reflecting all combinations of the discrete a_x and u_y values), whereby infeasible transitions (road departure, obstacle collision, negative longitudinal speed) are ignored. For each feasible discrete state, the corresponding optimal controls, along with the corresponding optimal cost-to-go value, are stored. The algorithm ends, when the initial state at $k = 0$ has

been evaluated. Eventually, starting from the given initial state and progressing forward step-by-step by following the respective optimal controls at each encountered discrete state, the optimal trajectories are obtained.

In view of the positive semi-definite nature of the objective function (23), alternative forward-branching procedures can be employed for the solution of the simplified DP problem, such as branch-and-bound or DFS (Depth-First-Search)-based approaches (Cormen et al., 2001). In the current work, in addition to the aforementioned standard DP, a forward-branching procedure, which employs DFS-based branching (Bar-Yehuda et al., 1989; Cho and Shaw, 1997), was also tested. In this approach, states $\mathbf{x}(k)$ are considered as nodes that can be branched, by application of discrete control combinations, to corresponding subsequent nodes $\mathbf{x}(k+1)$, whereby transitions that lead to infeasible states (nodes) are ignored. The initial state (root) $\mathbf{x}(0)$ is first branched towards one single (feasible) transition (i.e. using a specific control combination) all the way, until a state (node) $\mathbf{x}(K)$ of the final time K is reached. The cost of this path (trajectory) is provisionally stored as the current optimal cost. Then, the algorithm backtracks continuously, according to the DFS procedure, and creates new paths to the final time K if the arrival cost of all intermediate nodes is lower than the provisional optimal cost. In case a new complete path from $k = 0$ to K is found, which has lower cost, the previous optimal cost is replaced. The algorithm ends when all open (not branched) nodes have a higher arrival cost than the current optimal cost. Thus, the algorithm may leave many feasible nodes unvisited, which leads to corresponding reduction of the computational effort, without affecting the optimality of the solution. Remarkably, this DFS-based procedure, which leads to identical solutions as the standard DP algorithm, was found to take a computation time ten times lower, on average, compared to the standard DP algorithm for the problem at hand. Specific run times for all employed algorithms are reported in the next section.

More formally, the algorithmic steps of the forward-branching procedure are as follows:

- 1: Initialize: root vertex p (initial state), $J_{Best} = \infty, k = 0$
- 2: Call DFS algorithm for traversing, recursively:
- 3: **procedure** DFS(p)
- 4: label p as visited
- 5: **if** $k > K$ or $p == \text{infeasible}$ **then**
- 6: **return**
- 7: **end if**
- 8: **if** $k == K$ and $J_{DP,p} \leq J_{Best}$ **then**
- 9: $J_{Best} = J_{DP,p}$
- 10: **else if** $J_{DP,p} \geq J_{Best}$ **then**
- 11: **return**
- 12: **end if**
- 13: **for** each child q (combination of $\mathbf{u} \in U$) of p **do**
- 14: **if** q is undiscovered **then**
- 15: calculate $\mathbf{x}(k+1), \mathbf{y}(k+1), \mathbf{v}(k+1)$ and $J_{DP,q}$
- 16: **if** $\mathbf{x} \notin X$ **then**
- 17: flag q as infeasible
- 18: **end if**
- 19: $k = k + 1$
- 20: call DFS(q)
- 21: **end if**
- 22: **end for**
- 23: **end procedure**

where p is the parent nodes, q are the children nodes of a parent p , $J_{DP,q}$ is the cost obtained at a node q and J_{Best} is the optimal cost.

The complexity of both DP algorithms can be considered to be known beforehand and constant, due to the fact that the dimensions of the problem are fixed. Specifically, the problem is solved for fixed time-horizon, which means that the dimensions of the control and state variables are not changing. Of course, the actual execution complexity is reduced due to a number of factors, such as: surrounding vehicle density (affecting the amount of feasible states), current lane (one or both lateral movements possible), the size of a time step, the time-horizon length.

Due to the introduced simplifications, the resulting DP optimal control trajectories have to be processed appropriately, before being used as an initial control trajectory for FDA, so as to be consistent with FDA control inputs. More

specifically, the longitudinal acceleration resulting from the DP procedure is transformed to the corresponding jerk via simple Euler differentiation; while any lane changing delivered by the DP procedure is transformed into a lateral acceleration trajectory that leads to the intended lane change also in the FDA settings.

3.3. Safety Override

The proposed optimal control approach, with the use of a penalty function (14) for collision avoidance does not absolutely guarantee crash-free path generation. Although in the vast majority of cases, the solution of the optimization problem produces trajectories that do not include crashes, thanks to the selected high value of the corresponding weighting parameter, there are very rare cases where, due to the conflicting goals of the collision avoidance term versus the desired speed term, a trajectory including a crash may result. These two terms may be conflicting, as the first term may be striving to decrease the ego vehicle speed, in presence of a slower leading obstacle, while the second term is striving to increase the speed towards the desired speed. A balance between these two terms is typically reached (through the corresponding penalty weights in (12)), which guarantees efficient vehicle advancement while suppressing collisions. However, under extreme conditions, e.g. in high density scenarios, the possibility of a collision cannot be utterly excluded.

In order to avoid such decisions and ensure safety, an emergency rule is activated if the solution procedure produces an unsafe path. In this case, the just generated path is dropped, and a new one is generated, with reduced desired speed and planning horizon. Specifically, the desired speed is set equal to a percentage of the leading obstacle's speed (e.g. 95% of obstacle's current speed) and the planning horizon is reduced by half. Both these measures reduce the size of the desired-speed term and enable crash-free vehicle advancement. In particular, the reduction of the planning horizon is helpful because the objective function (12) is additive over the time steps; thus, in some rare driving scenarios, it may appear less costly to crash with the leading vehicle for the first few time steps and then achieve all goals for the rest of the planning horizon; instead of avoiding the collision and bearing a high cost each time step due to the desired-speed deviation. Factually, these actions definitely suppress collisions, as the desired speed term is not competing with the collision avoidance term anymore.

3.4. Model Predictive Control

Summarising, the presented numerical solution approach requires, as input data, the current (initial) ego vehicle (EV) state, as well as the current and future positions and speeds of obstacle vehicles (OVs); to produce optimal EV controls and states over a future time horizon KT . This is an open-loop solution, and, given the dynamic environment (moving OVs), the time horizon should be long enough to anticipate and prepare for future situations and avoid myopic control actions. On the other hand, as time advances, the uncertainties related to the changing environment (actual versus predicted OV movement) and to the actual vehicle advancement (versus the open-loop solution) increase, as increasing deviations from the assumed predictions are inevitable. To address these uncertainties, the open-loop solution procedure is cast in a model predictive control (MPC) frame, whereby the solution is re-computed online, using the same horizon KT (rolling or receding horizon), whenever substantial changes regarding the initial predictions are detected at any time before the end of the time horizon. The new computation uses updated initial states and updated predictions about the movement of OVs. This calls for computation times smaller than the path update period, something that is indeed satisfied by the presented efficient solution procedure.

Considering traffic flow with many vehicles, as considered in the subsequent simulation investigations, three types of vehicles are distinguished:

1. Manually driven vehicles, which, in the simulation investigations, are navigated by the employed microscopic simulator (Aimsun).
2. Automated vehicles without V2V communication capabilities, which are navigated according to the presented procedure. Such vehicles rely only on their own sensors to sense the current position and speed of other surrounding vehicles of any type, whose paths are predicted simply by assuming that they will keep their current lane and speed fixed over the EV planning horizon KT .
3. Connected automated vehicles (with V2V communication capabilities), which are also navigated according to the presented procedure. However, such vehicles broadcast their latest path decision to other surrounding connected vehicles; and can receive the latest path decisions of surrounding connected vehicles. Note that this information is broadcasted asynchronously, i.e. a path-planning decision by a vehicle is broadcasted as soon as

it is produced. Thus, connected AVs rely also on their own sensors to sense the current position and speed of other surrounding vehicles; but, in addition, they receive the latest path planning decision by other surrounding vehicles of the same type.

In the scenarios tested in this work, two cases of AV connectivity are considered:

- Non-connected automated vehicles
- Connected automated vehicles.

Both cases are tested for different penetration rates, along with manually driven vehicles, which are guided by the microsimulation platform.

As evidenced by the above statements, for manually driven vehicles and non-connected AVs, no enhanced information is available, i.e. the non-connected AVs have information about the surrounding vehicles only through their sensors. Consequently, they only know the initial position and speed of the obstacles. So, in order to predict their surrounding vehicles path, the AVs assume that all the obstacles keep their speed and lane fixed over the planning horizon.

On the other hand, connected AVs, having some obstacles of the same type, enjoy enhanced information exchange with such obstacles. Specifically, connected obstacle vehicles (which are also controlled by the proposed approach) send their last generated path to other connected AVs around them. Thus, whenever a connected AV needs to generate a path, it requests the last generated paths of its surrounding connected obstacles, which contains not only the initial position and speed, but also the acceleration and the lane information over the planning horizon. Note that the communication between the connected vehicles is asynchronous, which means that each controlled vehicle generates its path independently, in terms of time, of the other vehicles. Thus, the time a surrounding connected OV last generated its path may deviate from the time that the EV is calculating its own path, e.g., the EV is calculating at time k , while the OV had calculated its path at time $k - n$. For this reason, the available decided path for each connected OV extends up to time $k + K - n$, where K is the time horizon, and must be extended for the $n - 1$ missing time steps. This is done by considering that the OV speed and lane, for these remaining time-steps, remain constant and equal to the last communicated values (of time $k - n + K$).

As mentioned earlier, the EV path is re-generated in real time to address evolving deviations from the last predicted driving conditions. More specifically, a new updated path is generated in the following cases:

- The EV has driven for the duration of half planning horizon ($KT/2$) according to the last generated trajectory. The plan is then updated, even if no deviations are observed, because application of the second half of the last path may lead to myopic actions.
- One or more surrounding vehicles deviate substantially from their predicted paths, e.g. an OV changed lane or changed its speed significantly, compared to its predicted movement.
- A new vehicle enters into the planning zone around the EV, corresponding to a new OV that was not accounted for in the last EV planning.
- The controlled vehicle cannot track the produced path (e.g. when certain safety-related lane-changing restrictions are violated in the microscopic simulation environment).

It may be useful to highlight that the pursued OCP approach aims at minimising an objective function, which reflects the various vehicle tasks, the successful fulfillment of which may entail complex vehicle manoeuvres. In other words, the goal of the vehicle movement strategy is not to track a pre-specified vehicle path, but to create an opportune path; therefore, classic stability is not a major issue here, the main interest focussing on the quality (in terms of the degree of task fulfillment) of the created path.

4. Simulation Testing and Results

4.1. Simulation Environment

In order to evaluate the proposed path-planning approach, including its MPC-based application, in realistic environments, the procedure was implemented in the Aimsun's (Aimsun Next, 2019) micro-simulation platform with the

use of both provided API and SDK tools to integrate the developed path-planning procedures in the traffic context. This implementation enables the investigation of how vehicles, guided by our path-planning approach, interact with each other and with manually driven vehicles, that emulate human driving, in countless driving situations occurring for a variety of traffic conditions. Due to the discrete nature of the micro-simulator with respect to lateral vehicle movement, the produced path of each AV must be modified appropriately to enable its application within the simulator. Specifically, Aimsun does not allow for continuous lateral vehicle positioning or movement, other than discrete lane assignment and instantaneous (vertical) lane changing. Therefore, the continuous lateral AV movements, produced by the path-planning approach, must be translated appropriately in terms of Aimsun lane positioning. For example, in a three-lane motorway section with each lane being 3 m wide, Aimsun assumes that the right-most lane is lane 1, the middle lane is lane 2 and the left-most lane is lane 3. Thus, for Aimsun usage, the lateral EV position at any time must belong to one of these lanes. On the other hand, in the proposed path-planning approach, lateral vehicle position is continuous (as in real conditions), hence, the AV assignment to a discrete lane in the simulator is effectuated according to the AV lateral position: if the AV lateral position is within a range $[0, 3]$ m, the AV is assigned to the right-most lane; if it is within range $[3, 6]$ m, it is assigned to the middle lane; and for range $[6, 9]$ m, the AV is assigned to the left-most lane. The same applies also in case the AV is executing a lane change, leading to a “vertical” lane change, as required by Aimsun, according to Figure 3. This issue is not part of the proposed vehicle movement strategy, which, in a real application, would produce a continuous path. It is only for the needs of the commercial micro-simulation platform (Aimsun) that this discretisation is introduced, so the procedure is deemed to be of minor importance for the designed movement strategy, but is necessary to enable usage of Aimsun for the comprehensive traffic-level evaluation.

In fact, the OCP decides whether a controlled vehicle will apply a lane change or not, which depends on the time the optimal control decided this lane change to happen. As already mentioned in the previous section, that the OCP is solved for a fixed time horizon, K , however only half of this horizon is applied (at most). Thus, if a lane change was decided during the first $K/2$ period, then the vehicle actually applies the lane change manoeuvre; otherwise, if the lane change was decided to happen during the second half of the horizon, the vehicle just stays at its current lane, with its lateral position being the middle of this lane. Note that, in a lane-based highway environment, a vehicle mostly follows a leader or drives on its lane with its desired speed, as the lane-change manoeuvres are not frequent. So, in these cases, explicit consideration of wheel steering is not important. On the other hand, when a lane-change manoeuvre is decided, the vehicle must safely drive from the centre of a lane to the centre of an adjacent lane. This lane change manoeuvre, in real application, can be considered as a tracking problem, with pre-fixed lane-change path, for the corresponding vehicle. For example, the following procedure can be followed:

- The algorithm decides about when and where to effectuate a lane-change within a time horizon.
- Whenever a lane change is decided, a pre-fixed lane-change trajectory can be applied, e.g. simply using a sigmoid function from the centre of the vehicle's lane to the centre of the target lane (Claussmann et al., 2015).

For the simulation investigations, two cases were considered, each of them at different penetrations of AVs:

- No connectivity: Each AV is aware only of the current position and speed of obstacles (via its own sensors).
- Connected automated vehicles: Each AV is aware of the current position and speed of obstacles; in addition, it receives the path-planning decisions of other AVs, which facilitates more accurate short-term prediction of their movement.

In both cases, manually driven vehicles are moved according to Aimsun's Gipps (lane-changing) (Gipps, 1986) and IDM (car-following) (Treiber and Kesting, 2013) models.

All investigations use as a testbed a homogeneous motorway section of 3 km in length, with three lanes (each lane being 3 m wide). Two different levels of inflow, 3.000 veh/h and 5.000 veh/h, into this section are simulated, and vehicle trajectories and traffic conditions are monitored over a simulation horizon of 60 min. Entering vehicles are randomly assigned their characteristics (type of vehicle, dimensions, desired speed, initial lane and time gap). In particular, vehicle type is selected randomly, according to the examined penetration rate of AVs. All vehicles are "passenger cars" with dimensions selected randomly, from a default range, by the Aimsun simulator. The desired speed of each vehicle (of both types) is selected randomly, with uniform distribution, from a range $[80, 120]$ km/h. The constant time gap, ω , for both automated and manually driven vehicles, is also selected randomly, with uniform distribution, from a range $[0.8, 1.8]$ s (Spiliopoulou et al., 2018).

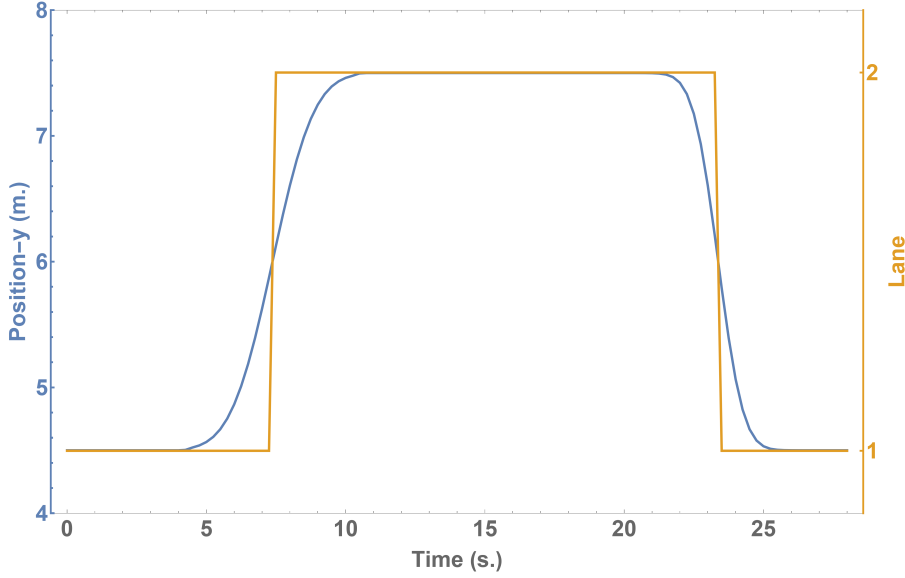


Figure 3: AV lateral position trajectory (blue line) and its mapping into Aimsun lanes (orange lines).

For AVs, a planning horizon of 8 s is used, with a path-planning step of $T = 0.25$ s. Thus, each plan consists of controls $j_x(k)$, $a_y(k)$ for $K = 32$ time steps. However, as mentioned earlier, half planning horizon is applied at most, which corresponds to 4 s, before a re-plan. The control bounds have been set to $j_x \in [-4.0, 4.0]$ and $a_y \in [-1.5, 1.5]$ for longitudinal and lateral controls, respectively, but it should be noted that high (absolute) control values are virtually never reached. In (13) $d = 1.4$; and in (14) $p_1 = p_2 = 18$, which lead to rectangular-like ellipses.

In an initial offline trial-and-error procedure, where many different driving situations were tested, the penalty weights in (12) were set to $[w_1, w_2, w_3, w_4, w_5, w_6, w_7] = [1.5, 1.0, 1.5, 0.05, 1.0, 15.0, 15.0]$. The choice of the penalty weights does not depend on the vehicle class or the infrastructure. Problems with multiple sub-objectives, reflected in corresponding weighted penalty terms, have a long history in optimisation. The appropriate specification of weights, so as to reflect the desired relative importance of the sub-objectives, is not a trivial task and is usually addressed via trial-and-error, i.e. by running experiments and evaluating their outcome. This typical trial-and-error procedure was pursued also in this work. Starting from an initial estimate about the appropriate weight values, the following steps were executed:

- (a) Run a simulation involving many different driving circumstances for the AVs.
- (b) Evaluate the AV behaviour during the simulation (e.g. crashes, road departures, travel time etc.).
- (c) If the vehicle behaviour is not satisfactory with respect to some sub-objectives, modify the weights accordingly and go to a).
- (d) Else appropriate weights have been found.

4.2. Results

Figure 4 displays the trajectories of several representative AVs, extracted from a simulation scenario with low inflow (3,000 veh/h) and 50% penetration of connected AVs. The trajectories refer to the longitudinal and lateral acceleration and speed; the longitudinal jerk; and the lateral continuous position, produced by FDA, along with the mapping based on lanes, required by the Aimsun microsimulation environment. In Figure 4a, it is observed that the AV cannot reach its desired speed (orange lines). This happens because there is no space to overtake and, consequently, it simply follows the leader by adapting its longitudinal speed in a car-following mode. In Figure 4b, the AV starts with a longitudinal speed equal to its desired speed. In this case, it can be seen that the vehicle needed to slow down

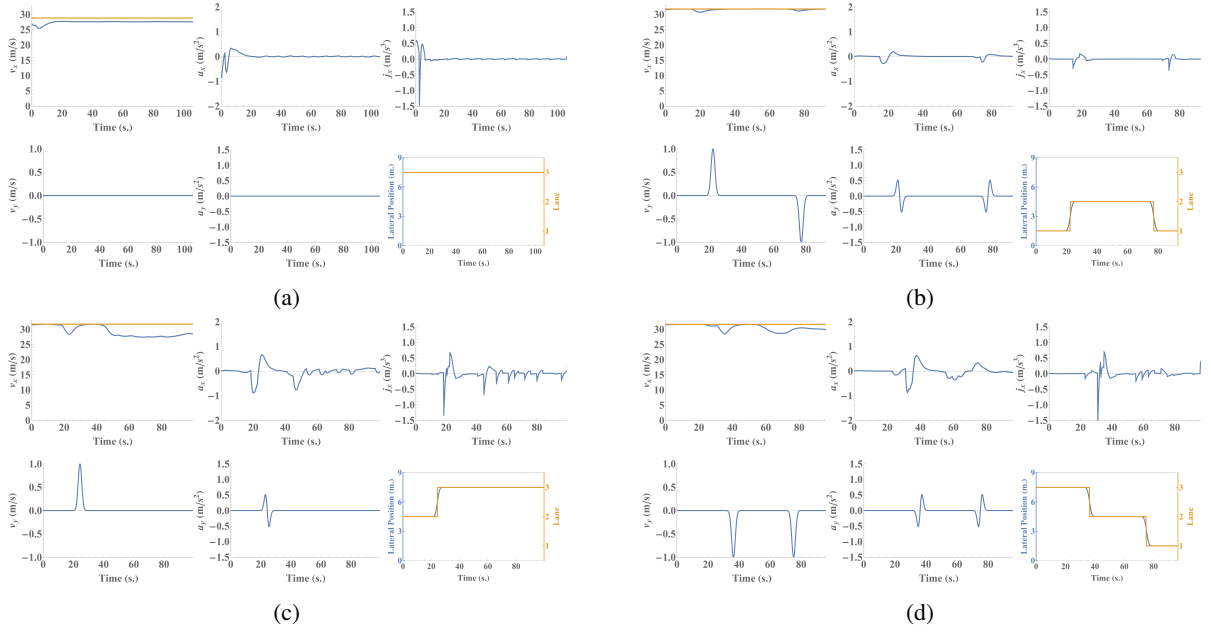


Figure 4: Trajectories of several representative AVs during the simulation, representing longitudinal and lateral speeds (orange line is the longitudinal desired speed) and accelerations, longitudinal jerk, and lateral continuous position produced by FDA (blue lines), along with the corresponding mapping on lanes (orange lines).

and apply lane changes twice, in order to maintain its desired speed. Finally, Figures 4c, 4d, combine both first two simpler cases, as the corresponding AVs are manoeuvring through the surrounding slower traffic, managing to achieve their desired speeds for small parts of their trip, while for the rest part they simply follow their leaders, waiting to find appropriate space to overtake, as human drivers would also do. As far as the longitudinal jerk and the longitudinal and lateral accelerations are concerned, the magnitudes are limited and quite smooth, which is good for passenger comfort and fuel consumption.

Figures 5 and 6 present results obtained for the two different demand levels, namely for 3.000 veh/h and 5.000 veh/h, respectively. Each figure contains results of different penetration rates of AVs. In addition, each figure displays and contrasts results corresponding to the two evaluated cases of AV connectivity: connected AVs (green lines) and non-connected AVs (blue lines). For each case, the solid lines reflect on the average results of the whole vehicle population, including both automated and manually driven vehicles, while the dense-dashed and sparse-dashed lines reflect on the average results of automated and manually driven vehicles, respectively. These summarized results concern the average delay time; the average speed; the average number of lane changes; and the average deviation from the desired speed.

Under both considered demand levels, the online path-planning approach appears to be more efficient in navigating AV speeds closer to the respective desired speeds, compared to manually driven vehicles. Apparently, the suggested approach is more successful at exploiting gaps through traffic and applies "smarter" maneuvers; which leads not only to better performance of each AV, but also to increased overall traffic performance. Specifically, in both Figures 5 and 6, as the penetration rate of AVs rises, an increase of the average speed, and consequently a significant decrease of the delay, for the whole traffic, is observed. Note that, this improvement also affects the manually driven vehicles, which appear to also benefit from the AVs presence and decisions.

In terms of lane changing, the number of lane changes of all vehicles, in the lower demand level (Figure 5c), is higher compared to the higher demand level (Figure 6c). This is due to the density prevailing in each case, where, for the lower demand levels there is more space for the vehicles to apply a lane change in order to overtake slower traffic. Moreover, in Figure 5c, it can be observed that the number of lane changes of AVs is decreasing as their penetration rate rises. This happens as, for low penetration rates, the AVs need to overtake slower traffic, including other slower AVs or manually driven vehicles, which do not have the same capability to achieve their target speed; while for higher penetration rates, the need for overtaking is reduced, as the increased number of AVs ensures speeds closer to the target

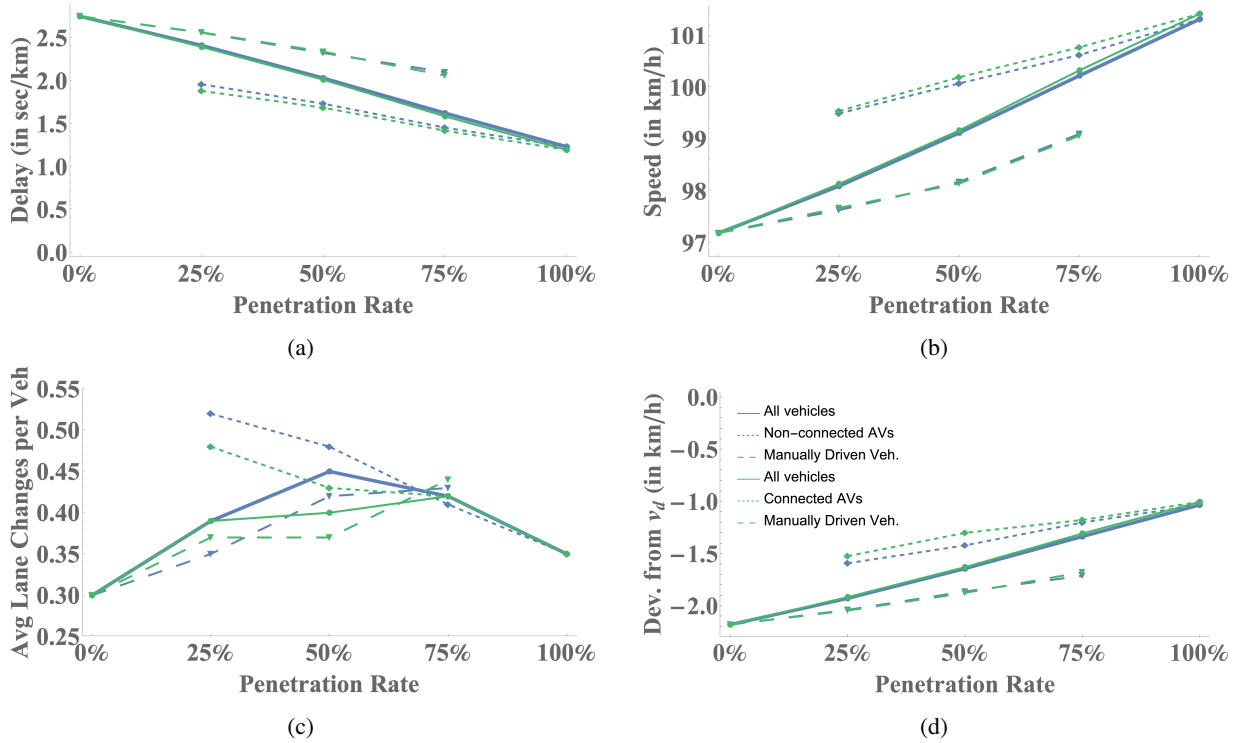


Figure 5: Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 3.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.

speed of each vehicle; thus, it is less probable for a vehicle entering the section to face a slower one. On the other hand, in higher demand levels (Figure 6), the number of lane changes of AVs is increasing with their increasing penetration. In this case, where traffic is denser, it is harder, for both the manually driven vehicles and the AVs to overtake. Thus, as the penetration rate of AVs rises, there are more overtakes from AVs, which exploit the available space better, helping to maintain increased speed and consequently create spaces for the following traffic. In both demand levels, the AVs efficient maneuvering behavior allows for the manually driven vehicles to increase their lane changes as well.

Contrasting the two types of AVs, the results in Figure 5 indicate only small differences, with connected vehicles being able to achieve slightly better performance in higher penetration rates. This similar outcome is due to the fact that, in lower demand levels, driving space is ample for an AV to maneuver efficiently, which is also evident from Figure 5c, where the difference in the average number of lane changes is moderate. For the same reason, all vehicles (manually driven and automated) do not need to change their speed frequently or strongly, hence the added value of receiving improved information (last path decision) from the surrounding AVs, through the connectivity with other vehicles, is not significant. On the other hand, in Figure 6, where the demand level is higher, vehicle connectivity is seen to have a high impact on the vehicles performance. In this scenario, it is noticed that, although both types of vehicles manage to achieve improved performance, connected AVs outperform the non-connected ones, as the penetration rate increases, with most noticeable differences in the area of 75%-100%. This is explained as, in denser conditions, the enhanced information that the AVs have about the surrounding traffic enable them to achieve better reactions in need of a lane change. That means, that the connected AVs are able to apply few lane changes due to better timing, compared to the non-connected ones, which also leads to keeping their speed closer to the desired speed.

Finally, Figure 7 reports on the average number of plans (and re-plans) of the two types of AVs, in dependence of the penetration rates and for both demand levels. In Figure 7a, it is noticed that both types of AVs have approximately the same average number of (re-)plans. In this low demand case, all AVs are able to navigate close to their desired speeds, with no significant changes to their predicted paths. The slightly reduced values for the connected AVs are due

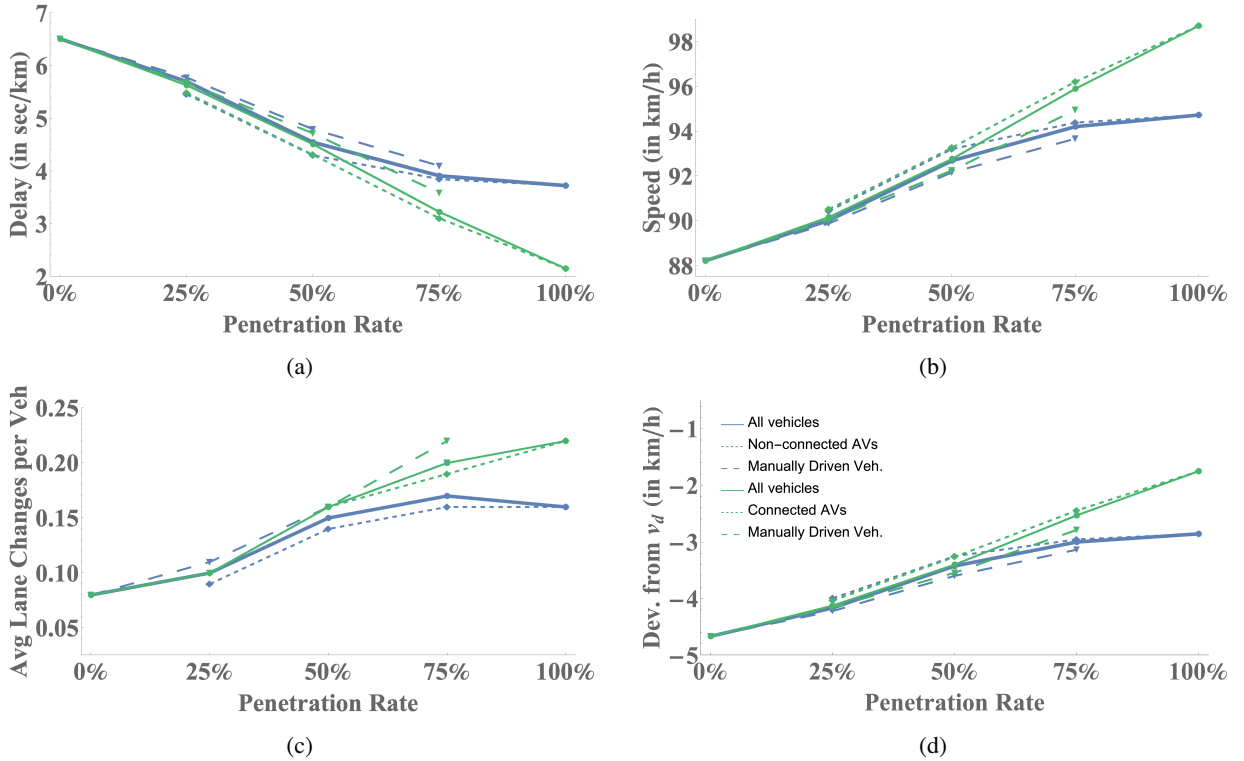


Figure 6: Average delay time, speed, deviation from target speed and number of lane changes for: i) connected (green lines); and ii) non-connected (blue lines) AVs; for a demand of 5.000 veh/h. Solid lines represent the average of the whole section, while dense-dashed and sparse-dashed lines reflect on average results for automated and manually driven vehicles, respectively.

to their enhanced information, specifically the knowledge of the lane changes that other neighboring AVs have planned to apply. On the other hand, in Figure 7b, where the higher demand level case is presented, it is observed that, as the penetration rate increases, there is an increase of plan numbers for both types of AVs. As also mentioned above for the lane changing behavior, this is because in this case, for low penetration rates, the surroundings of each AV do not change much, hence there are few deviations from the predicted paths for the obstacles. However, as the penetration rate rises, the AVs are more capable in maneuvering through traffic, which increases the need for re-plans. In these conditions, the number of lane changes increases, vehicles may accelerate or decelerate after an overtake or may reach a vehicle downstream, which was not included in their initial prediction. This increase of the average number of plans is different for the two types of AVs, with connected AVs demanding less re-plans compared to the non-connected ones, as their enhanced knowledge allows them to have more accurate view of the surrounding traffic, including the intended lane changes and the tendency of other AVs to accelerate or decelerate.

Regarding computation times, the average CPU-time per planning of an AV path during the simulation is 0.1 s for the DP, 0.01 s for forward-branching DP and 0.01 s for FDA, with the corresponding maximum values during a whole simulation being 0.2 s for DP, 0.06 s for forward-branching DP and 0.2 s for FDA, which indicates that the proposed approach is clearly real-time feasible. In addition, in Table 1, the average execution times of the three algorithms, for different planning horizons are reported, which verify their polynomial increase against the time-horizon length; and the real-time applicability of the proposed approach, even for increased time horizons.

Demonstration of the vehicle movements in Aimsun micro-simulation platform is available as videos at <https://bit.ly/3m2nPa2>. Specifically, there are three videos, showing the connected AVs' behavior in both demand levels, i.e. 3.000 veh/h (video-1) and 5.000 veh/h (video-2 and video-3), both for 100% penetration rate. It is evident from the videos that the above OCP formulation is conceived for American freeway traffic rules, whereby vehicles may use any lane at any speed and may overtake on the left or right. Adoption of European driving rules, where overtaking is only from left, may be easily accommodated according to (Makantasis and Papageorgiou, 2018).

Table 1

Average execution times of the three proposed algorithms, FDA, DP and Forward DP, for different planning horizons.

| Time Horizon (KT) | Average CPU Time | | |
|-----------------------|------------------|------|------------|
| | FDA | DP | Forward DP |
| 6 seconds | 0.002 | 0.04 | 0.001 |
| 8 seconds | 0.010 | 0.09 | 0.010 |
| 10 seconds | 0.020 | 0.12 | 0.015 |
| 12 seconds | 0.035 | 0.17 | 0.018 |

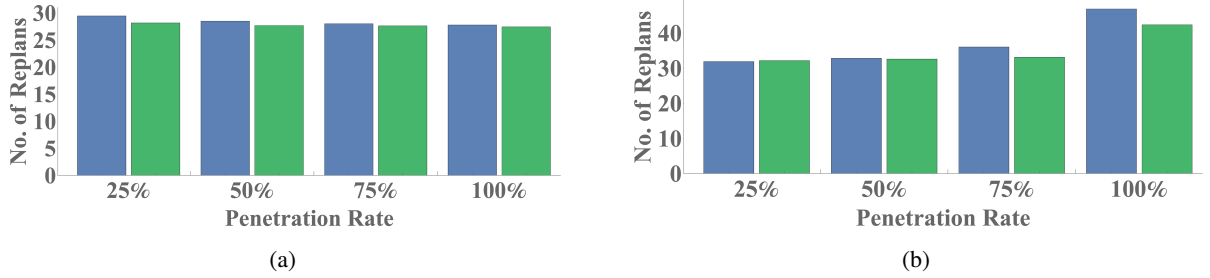


Figure 7: Average number of (re-)plans for different penetration rates of the two evaluated cases: non-connected AVs (blue); and connected AVs (green) and for different demand levels: a) 3.000 and b) 5.000 veh/h

5. Conclusions

Automated vehicle path-planning has been expressed as an optimal control problem. A combination of DP and NLP techniques allows obtaining good local minima for this non-convex optimization problem efficiently. This efficiency facilitates an online MPC-based (re-)planning approach, by observing or receiving information (according to the connectivity features) and predicting the trajectory of the surrounding vehicles and adapting the EV path accordingly. The proposed MPC-based approach is embedded within the Aimsun micro-simulation platform, enabling us to thoroughly examine the behavior of the approach in presence of vehicles emulating human driving behavior and in a plethora of realistic driving instances. For the simulation investigations, two cases were considered, each of them at different penetration rates of AVs: i) non-connected vehicles, where each AV is aware only of the current states of obstacles, which are either manually driven or automated; and ii) connected AVs, where each AV is aware of the current states of obstacles, but, in addition, it receives (asynchronously) the path-planning decisions of other AVs.

Demonstration results are reported for a homogeneous motorway stretch and different lane-capacity utilizations. Based on the results, it is concluded that the introduction of AVs, guided by the suggested approach, benefits the overall traffic performance. Specifically, in all scenarios, as the penetration of AVs rises, an increase of the average speed and consequently a decrease of the average delay of both automated and manually driven vehicles are observed. AVs appear to be more effective to navigate closer to their desired speed compared to the manually driven vehicles in all tested scenarios. Specifically, in the case of lower demand level, both non-connected and connected AVs manage to improve their average delay time by 44%, with the speed being increased by 4% compared to the manually driven vehicles. On the other hand, for the higher demand level scenario, the improvement of the average delay time was 38% and 69% for the non-connected and the connected AVs respectively; while the average speed was increased by 6% and 10%, respectively. The improved performance of controlled vehicles stems from their ability to apply smarter manoeuvres, while in very high density or congested cases, the performance of both controlled and non-controlled vehicles is expected to be similar, due to very limited or no space to exploit. Of course, further investigation, in such cases, is required, which is part of future work. As far as the AVs are concerned, in lower demand levels both connected and non-connected AVs perform similarly, due to sufficient space for maneuvering. The superiority of connectivity becomes evident in higher demand levels, as the enhanced information about the surrounding traffic is crucial, and connected AVs appear more efficient at increased penetration rates, due to the improved real-time information that enables more pertinent obstacle movement prediction.

Current and future work is focused on:

- Investigation of scenarios, which feature very high demand levels and congested areas.
- Consideration of platooning and multiple vehicle coordination and study of the synergistic effects and impact on the overall traffic flow.
- Application of a similar approach for urban road networks.
- Introducing on-ramps, off-ramps and emergency cases, e.g. controlled vehicles need to create appropriate space for an ambulance.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 Research and Innovation programme / ERC Grant Agreement n. [833915], project TrafficFluid.

References

- Aimsun Next, ., 2019. Transport simulation system (tss). <https://www.aimsun.com/aimsun-next>.
- Bar-Yehuda, R., Feldman, J.A., Pinter, R.Y., Wimer, S., 1989. Depth-first-search and dynamic programming algorithms for efficient cmos cell generation. *IEEE transactions on computer-aided design of integrated circuits and systems* 8, 737–743.
- Cho, G., Shaw, D.X., 1997. A depth-first dynamic programming algorithm for the tree knapsack problem. *INFORMS Journal on Computing* 9, 431–438.
- Claussmann, L., Carvalho, A., Schildbach, G., 2015. A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams, in: 2015 European Control Conference (ECC), IEEE. pp. 2976–2982.
- Claussmann, L., Revilloud, M., Gruyer, D., Glaser, S., 2019. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* 21, 1826–1848.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., et al., 2001. Introduction to algorithms, chapter 11.
- Dey, K.C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C., Soundararaj, V., 2015. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc). *IEEE Transactions on Intelligent Transportation Systems* 17, 491–509.
- Dixit, S., Montanaro, U., Dianati, M., Oxtoby, D., Mizutani, T., Mouzakitis, A., Fallah, S., 2019. Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc. *IEEE Transactions on Intelligent Transportation Systems* 21, 2310–2323.
- Earl, M.G., D'Andrea, R., 2007. A decomposition approach to multi-vehicle cooperative control. *Robotics and Autonomous Systems* 55, 276–291.
- Ghiasi, A., Ma, J., Zhou, F., Li, X., 2017. Speed harmonization algorithm using connected autonomous vehicles, in: 96th Annual Meeting of the Transportation Research Board.
- Gipps, P.G., 1986. A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological* 20, 403–414.
- Goniewicz, K., Goniewicz, M., Pawłowski, W., Fiedor, P., 2016. Road accident rates: strategies and programmes for improving road traffic safety. *European journal of trauma and emergency surgery* 42, 433–438.
- González, D., Pérez, J., Milanés, V., Nashashibi, F., 2015. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 17, 1135–1145.
- Gu, D., Hu, H., 2002. Neural predictive control for a car-like mobile robot. *Robotics and Autonomous Systems* 39, 73–86.
- Gu, T., Dolan, J.M., 2014. Toward human-like motion planning in urban environments, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings, IEEE. pp. 350–355.
- Haydari, A., Yilmaz, Y., 2020. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 1–22doi:10.1109/TITS.2020.3008612.
- Howard, T.M., Green, C.J., Kelly, A., 2010. Receding horizon model-predictive control for mobile robot navigation of intricate paths, in: *Field and Service Robotics*, Springer. pp. 69–78.
- Jalalmaab, M., Fidan, B., Jeon, S., Falcone, P., 2015. Model predictive path planning with time-varying safety constraints for highway autonomous driving, in: 2015 International Conference on Advanced Robotics (ICAR), IEEE. pp. 213–217.
- Khazaeni, Y., Cassandras, C.G., 2016. Event-driven cooperative receding horizon control for multi-agent systems in uncertain environments. *IEEE Transactions on Control of Network Systems* 5, 409–422.
- Makantasis, K., Papageorgiou, M., 2018. Motorway path planning for automated road vehicles based on optimal control methods. *Transportation Research Record* 2672, 112–123.
- Malikopoulos, A.A., Hong, S., Park, B.B., Lee, J., Ryu, S., 2018. Optimal control for speed harmonization of automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* 20, 2405–2417.
- Mayne, D.Q., 2014. Model predictive control: Recent developments and future promise. *Automatica* 50, 2967–2986.
- Mayne, D.Q., Michalska, H., 1988. Receding horizon control of nonlinear systems, in: *Proceedings of the 27th IEEE Conference on Decision and Control*, IEEE. pp. 464–465.
- Montanaro, U., Dixit, S., Fallah, S., Dianati, M., Stevens, A., Oxtoby, D., Mouzakitis, A., 2019. Towards connected autonomous driving: review of use-cases. *Vehicle system dynamics* 57, 779–814.

- Murillo, M., Sánchez, G., Genzelis, L., Giovanini, L., 2018. A real-time path-planning algorithm based on receding horizon techniques. *Journal of Intelligent & Robotic Systems* 91, 445–457.
- Nagy, B., Kelly, A., 2001. Trajectory generation for car-like robots using cubic curvature polynomials. *Field and Service Robots* 11.
- Nilsson, J., Falcone, P., Ali, M., Sjöberg, J., 2015. Receding horizon maneuver generation for automated highway driving. *Control Engineering Practice* 41, 124–133.
- Ntousakis, I.A., Nikolos, I.K., Papageorgiou, M., 2016. Optimal vehicle trajectory planning in the context of cooperative merging on highways. *Transportation research part C: emerging technologies* 71, 464–488.
- Papageorgiou, M., Leibold, M., Buss, M., 2015. *Optimierung*. volume 4. Springer.
- Papageorgiou, M., Marinaki, M., Typaldos, P., Makantasis, K., 2016. A feasible direction algorithm for the numerical solution of optimal control problems—extended version. Chania, Greece: Technical University of Crete, Dynamics Sysyems and Simulations Laboratory , 2016–26.
- Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control engineering practice* 11, 733–764.
- Rajamani, R., 2011. *Vehicle dynamics and control*. Springer Science & Business Media.
- Rasekhipour, Y., Khajepour, A., Chen, S.K., Litkouhi, B., 2016. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems* 18, 1255–1267.
- Rios-Torres, J., Malikopoulos, A.A., 2016. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems* 18, 1066–1077.
- Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R., 2020. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations, in: *European Conference on Computer Vision*, Springer. pp. 414–430.
- Saxena, D.M., Bae, S., Nakhaci, A., Fujimura, K., Likhachev, M., 2020. Driving in dense traffic with model-free reinforcement learning, in: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. pp. 5385–5392.
- Shao, Y.S., Chen, C., Kousik, S., Vasudevan, R., 2021. Reachability-based trajectory safeguard (rts): A safe and fast reinforcement learning safety layer for continuous control. *IEEE Robotics and Automation Letters* 6, 3663–3670.
- Sjoberg, K., Andres, P., Buburuzan, T., Brakemeier, A., 2017. Cooperative intelligent transport systems in europe: Current deployment status and outlook. *IEEE Vehicular Technology Magazine* 12, 89–97.
- Spiliopoulou, A., Manolis, D., Vadorou, F., Papageorgiou, M., 2018. Adaptive cruise control operation for improved motorway traffic flow. *Transportation research record* 2672, 24–35.
- Tian, D., Wu, G., Boriboonsomsin, K., Barth, M.J., 2018. Performance measurement evaluation framework and co-benefit/tradeoff analysis for connected and automated vehicles (cav) applications: A survey. *IEEE Intelligent Transportation Systems Magazine* 10, 110–122.
- Treiber, M., Kesting, A., 2013. *Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg .
- Typaldos, P., Papamichail, I., Papageorgiou, M., 2020. Minimization of fuel consumption for vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems* 21, 1716–1727.
- Wang, H., Huang, Y., Khajepour, A., Liu, T., Qin, Y., Zhang, Y., 2018a. Local path planning for autonomous vehicles: Crash mitigation, in: *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE. pp. 1602–1606.
- Wang, Y., Li, X., Yao, H., 2018b. Review of trajectory optimisation for connected automated vehicles. *IET Intelligent Transport Systems* 13, 580–586.
- Wang, Z., Wu, G., Barth, M.J., 2018c. A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications, in: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE. pp. 2884–2891.