Sim-to-real transfer and reality gap modeling in model predictive control for autonomous driving

(article starts on next page)

Sim-to-real transfer and reality gap modeling in model predictive control for autonomous driving

Iván García Daza[1] · Rubén Izquierdo[1] · Luis Miguel Martínez[1] · Ola Benderius[2] · David Fernández Llorca[1,3]

Abstract

The main challenge for the adoption of autonomous driving is to ensure an adequate level of safety. Considering the almost infinite variability of possible scenarios that autonomous vehicles would have to face, the use of autonomous driving simulators is becoming of utmost importance. Simulation suites allow the used of automated validation techniques in a wide variety of scenarios, and enable the development of closed-loop validation methods, such as machine learning and reinforcement learning approaches. However, simulation tools suffer from a standing flaw in that there is a noticeable gap between the simulation conditions and real-world scenarios. Although the use of simulators powers most of the research around autonomous driving, and is generally used within all domains it is divided into, there is an inherent source of error given the stochastic nature of activities performed in real world, which are unreplicable in computer environments. This paper proposes a new approach to assess the real-to-sim gap for path tracking systems. The aim is to narrow down the sources of error between simulation results and real-world conditions, and to evaluate the performance of the simulation suite in the design process by employing the information extracted from gap analysis, which adds a new dimension of development against other approaches for autonomous driving. A real-time model predictive controller (MPC) based on adaptive potential fields was developed and validated using the CARLA simulator. Both the path planning and vehicle control systems where tested in real traffic conditions. The error between the simulator and the real data acquisition was evaluated using the Pearson correlation coefficient (PCC) and the max normalized cross-correlation (MNCC). The controller was further evaluated on a process of sim-to-real transfer, and was finally tested both in simulation and real traffic conditions. A comparison was performed against an optimal-control ILQR-based model predictive controller was carried out to further showcase the validity of this approach.

1 Introduction

In recent years, autonomous driving has attracted the interest of many stakeholders, such as car manufacturers, tech companies and policy makers, as it can significantly improve road safety and energy efficiency in transportation. Autonomous vehicles face multiple technical challenges that can be broadly categorized into five main areas [1]: (1) route generation, (2) perception, (3) decision making,

(4) local path planning and (5) control. Route generation is responsible of finding a global route between origin and destination, identifying the waypoints needed to travel between these two objectives from a pre-known digital map. Perception deals with accurate localization and dynamic scene understanding, detecting all agents in a scene and predicting their motion. The decision making process deals with high-level planification of driving manouvers, such as performing an overtake or giving way. Local path planning receives information from all other previous subsystems to decide a short-term route according to some optimization criteria. Finally, the controller performs the actions needed so that the car accurately follows the trajectory, be it via steering (lateral control) or throttling and braking (longitudinal control). These concepts are further explained and detailed in the works of Paden [2] and Milanes [3].

Rubén Izquierdo, Luis Miguel Martínez, Ola Benderius and David Fernández Llorca contributed equally to this work.

✉ Iván García Daza
 ivan.garciad@uah.es

Extended author information available on the last page of the article.

One of the most successful techniques in path planning and control systems is based on the *model predictive control* (MPC) framework, which evaluates both optimal vehicle trajectory and control actions over a finite time-horizon. However, effective MPC systems are based on constrained optimization processes, which involve high computational costs, rendering them mostly unable to perform as real-time systems. Moreover, real-world tuning of parameters is often unfeasible, due to the hight number of different values that dictate the behaviour of the system.

This makes the MPC design process benefit greatly from the use of simulation environments. Autonomous driving simulators are playing an increasingly important role in the development and testing of systems in autonomous vehicles, since they allow massive automated validation and testing in any type of scenario, as well as parameter refinement without physical or real-time constraints. Nevertheless, this powerful methodology is only applicable when the discrepancy between simulated and real environments is reasonably low and is thoroughly narrowed. This discrepancy is usually referred to as *sim-to-real gap* (simulation-to-reality) or *reality gap*.

One approach to ensure the *reality gap* is as small as possible resides in the use of realistic driving simulators, suites that show a high defree of similarity in terms of the physical beahvior of the vehicle under different weather conditions, road rugosity and friction, road profile, and several other factors, which are a fundamental part in the development and testing of innovative algorithms of various kinds applied to autonomous driving. Tackling a development process by using full-environment simulation as a tool that shortens the development cycle simplifies validation and increases efficiency, given that the *reality gap* is within critical limits.

1.1 Motivation and problem statement

Figure 1a shows the life cycle through which the development and research of a new system or function. The life cycle is exemplified by the analysis, design, development and testing/validation of a path planning and control system for an autonomous vehicle. A clear benefit of the acknowledgment of the *reality gap* and its mitigation arises in the capability of testing a given system within its simulation environment of development, since the sources of error between the simulation and is projected behaviour in the real world are well known and its values are narrowed down into a known interval with a specific trust value.

Modern and domain-specific simulators cover various aspects of the development process, such as scenario generation, data gathering, realistic physics laws, natural traffic flow, and photo-realistic computer graphics. Both automotive and *Big tech* companies invested in self-driving vehicles

and have widely adapted simulation environments to test and validate their systems. However, this is only feasible if it can be justified that the implemented system behaves similarly in both environments, demonstrating a sufficiently small *reality gap*, so that the simulated validation can be used to qualify the system as viable. Therefore, quantifying the simulation and real-world transferability is an essential step for developing autonomous driving algorithms and systems in simulation.
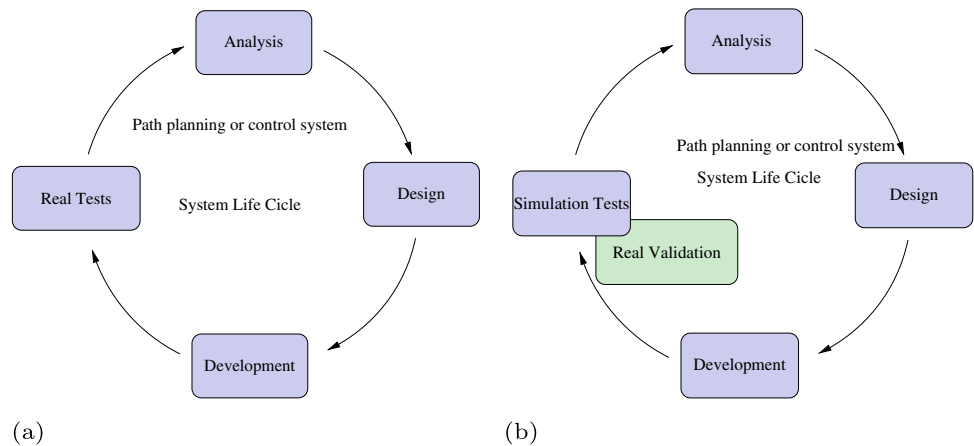
Waymo has accumulated the most virtual mileage of all companies investing in self-driving technology, with a total of eleven years and fifteen million simulated miles in February 2021 [4]. Their simulator, Carcraft, transforms real-world scenarios into virtual formats and runs 25,000 virtual cars simultaneously. The massive data flow from this process assists engineers in locating bugs and adjusting models efficiently. Apart from Carcraft, Apollo (Baidu) [5], XVIZ (Uber), AVS (GM Cruise), VIRTTEX (Ford), Nvidia Driveworks are some of the premium driving simulators used by the self-driving vehicle industry. However, this simulation suites are often proprietary, and require a high fee to be deployed in institutions other than the ones they were developed. Academia relies on a number of open-source or otherwise readily available alternatives, such as CARLA [6], Microsoft AirSim [7], VisSim, CarSim [8] and Gazebo [9].

However, as far as the authors are aware, there has been little to no work performed in quantifying correlation between simulation results and different real world scenarios. There have been a number of recent studies which review simulators and their performance when benchmarked with open real-world datasets revelant to the development of autonomous vehicles, such as [10], but to the best of our knowledge the *reality gap* remains unexplored as a means to further empower the design process inside simulation suites.

The aim of this work is to model *sim-to-real* transferability in a manner that allows the development of a systematic and efficient process for designing, testing, verifying and validating a steering controller for an autonomous vehicle. What this framework hopes to accomplish is to eliminate the need of real-world prototypes up until the very last step of design, which is projected to be the evaluation of comfort and perceived safety by the occupants of the vehicle. Our solution was tested on a MPC with artificial potential fields, which is the controller employed in both the simulation environment and the prototype to test our approach.

The *reality gap* for autonomous driving can be analyzed from different perspectives depending on the specific task. In our case, we showcase a statistical approach to quantifying the *reality gap* in the context of path planning. The analysis of acquired data both in the DRIVERTIVE platform [11] and in the simulation environment allows for

Fig. 1 Life cycle of the development of a system. a Development, testing and validation in real environments; b development and testing in simulated environments and validation in real environments

(a)   (b)

both parameter optimization and to spot poorly handled driving scenarios that need to be specifically recognised and included within the design aspect of the controller. This leads to a more efficient, elegant and effective solution for path planning and vehicle control, the aspects of autonomous driving that this controller is in charge of handling.

The authors note that, while this work has been performed in a specicic simulation suite -namely, CARLA-, the framework presented relies only on data acquisition and statistical analysis, which makes it transferable to any an all other simulators relevant to the autonomous driving research scene. The methodology described in this paper is exemplified with the use of a given MPC controller and an already established platform, but the statistical framework holds true for any given implementation of a simulated solution or a prototyped one.

1.2 Paper structure

The contents of this paper are structured as follows: Section 1 conveys an introduction in which the scope of the problem is introduce. Section 2 includes a discussion on the state of the art concerning *reality gap* and its assessment, along with a discussion on predictive control and its application in autonomous driving and a concise and clear statement on what the main contribution of this work aspires to be. Section 3 is composed of several subsections in which the entirety of the framework of our investigation is based upon, namely the autonomous driving platform, the model of motion and its application in the MPC controller, the design characteristics of the adaptive potential field subsystem, and a definition of the statistical analysis that will be carried out on the acquired data streams. Section 4 presents the results of our experiments both in terms of controller performance and perceived safety, as well as a comparison against an existing MPC controller. Finally, Section 5

includes the conclusions that can be derived from this work, as well as the future lines of work that arise from it.

2 State of the art

2.1 Reality gap in autonomous driving control systems

The analysis of the *reality gap* in control systems has been carried out in depth in the field of robotics. Wenshuai Zhao et al. present in [12] a survey on sim-to-real transfer techniques applied in robotics by developing control systems in simulated robotic systems based on reinforcement learning. They also discuss the need to evaluate the *reality gap* between simulated and real behaviour. Salvato et al. [13] conducted a recent survey about closing the *reality gap* and reduce the *sim-to-real* transferability when developing a robot controller based on reinforcement learning. The *reality gap* was identified and the implications of its closure and development states were discussed. However, as identified by Valassakis et al. [14] when analyzing how to reach a zero *sim-to-real* transfer in complex dynamics systems, the correlation between simulation and real-world results is rarely quantified, mainly due to the difficulty and risks inherent to testing in real environments.

Regarding autonomous vehicles, the authors have not found a solid line of research on how to evaluate and quantify the *reality gap* derived from using simulation as a tool for designing real-world control systems (e.g., trajectory tracking). Some preliminary work attempted to address *sim-to-real* transferability in highly simplified simulation environments with a loose resemblance to real driving scenarios, as in the case of miniature cars [15]. The same type of simplified environments has recently been proposed to quantify the *reality gap* in autonomous driving (i.e., localization) in terms of accuracy, behaviour and failures [16].

However, the conclusions that can be drawn from these simple environments are not directly applicable to real autonomous driving environments.

2.2 Predictive control for autonomous driving

Recently, several attempts have been carried out with the goal of employing MPC in the realm of autonomous driving, typically in path planning, trajectory tracking, obstacle avoidance, or overtaking manoeuvres. In [17], a high-speed obstacle avoidance system was developed based on MPC, manoeuvring in the shortest time achievable and under dynamic safety limits. The obstacle avoidance would be executed when the vehicle does not have enough time to stop. This work considered a four-wheel dynamic model involving lateral tire forces and slip angle, where lateral tire forces are defined as non-linear. The high number of constraints defined to fulfil the optimization process renders excessive computation time and a not properly convergent process. Thus, p-norm aggregation techniques were applied to improve both the computational cost and the optimization result. In [18] a linear-time-varying MPC (LTV-MPC) was integrated as a slip angle controller, to stabilize a vehicle during a sudden lane change or or when entering a curve with excessive speed. The MPC controller was designed using a three degree of freedom non-linear vehicle model to fulfil the dynamic constraints such as slip angle or lateral tire forces. However, even though both examples provide interesting and academically meaningful discussion, the results and conclusions are entirely based on simulation environments.

In two examples of obstacle avoidance based on artificial potential fields [19, 20], a two stage system involving trajectory planning and trajectory tracking was used. Trajectory planning was implemented with artificial potential fields considering surrounding vehicles, road boundaries, and lane centers. In the first controller [19], several possible safe trajectories are generated. One of them is selected based on the state of the potential field of the surrounding environment, solving an optimization problem that generates a sub-optimal solution to the problem. An MPC controller then follows the sub-optimal selected trajectory in the later stage. In the second controller [20], the optimization problem is designed to move over the minimum potential filed values, so the path-planning and control tasks are performed in the same stage. Both works showcase splendid results in simulation settings, but do not discuss an implementation in a real prototype to assess its performance in a real world environment. A hybrid planning approach based on MPC and parametric curves for overtaking manoeuvers has also been presented in [21]. In this work, the passenger comfort associated with the smoothness of Bézier curves was

combined with the reliable capabilities of MPC to react to unexpected conditions, such as lane obstacles, overtaking maneuvers and lane changes.

A hybrid trajectory planning approach for roundabout merging scenarios has been demonstrated in [22]. Roundabout merging is solved with a nominal trajectory generated through Bézier curves combined with an MPC to ensure safe future states. The computation time is reduced without sacrificing the performance of both techniques while providing safe execution of the manoeuvre. The approach has been tested in a real-world setting. However, no *reality gap* indicator is evaluated for improving development scalability.

2.3 Contributions

The main contribution of this work is to provide a solution to the trajectory tracking problem in real world environments using a simulation based approach. Our hope is to circumvent some of the risks and difficulties involved in carrying out the desing with a real prototype, such as road hazards, glitches and high costs.

To achieve this goal, we showcase a MPC steering controller based in adaptive potential fields that has been designed to be a reliable system both via including physical constraints in the optimization stage that deal with saturation in the input signal by enforcing a steering and steering speed limit, and by optimizing the code to run as fast as possible, so that the system can be guaranteed to operate within a specified time interval. We have developed a simulation environment that replicates a real trajectory as closely as we could manage, in hopes of reducing the variability of behaviours between the simulator and our real prototype. By statistically analising data acquired in both settings, we are able to determine if the behaviour of the simulation closely ressembles its counterpart, which enables a full software approach on the desing of the controller, thus supressing the need of testing in the real platform until the later stages of deployment. Our testing setting of choice includes well-differentiated scenarios and road structures, such as straights, curves and roundabouts, with a variety of curvature radii, which ensures the adaptability of the controller and makes for a complete suite of situations in which to test our approach.

A lesser contribution of our approach resides in the spatial distribution of the prediction horizon in the MPC controller. A difference in performance was noticed when the spatial horizon of the controller was distributed other than linearly, which will be assessed in terms of trajectory tracking.

To the authors' knowledge, this is the first study dealing with *sim-to-real* transfer and *reality gap* quantification of

autonomous vehicle control task that includes results with a real automated vehicle in real traffic conditions, including different types of intersections, curves and roundabouts.

3 Framework

The paper aims to evaluate the *reality gap* of a steering control system of an autonomous vehicle due to *sim-to-real* transfer done in design and implementation, with a methodology that compares the controller's performance in a real environment and its virtual equivalent. The driving simulator used in this work does not include a Citröen C4 vehicle in its collection; so a vehicle with similar characteristics (Toyota Prius) was used, parameterizing its localization device with a precision of 2 cm, such as the real car counterpart. The real and simulated car geometry was based on the center of gravity and is further described in Section 3.2. Standard road features were used in both real and simulated scenarios. The vector $[x_w, y_w, v, \psi]^T$ defines the state vector of the system, where x_w is the Easting, y_w is the Northing, v longitudinal velocity of the vehicle and ψ heading of the vehicle. The state vector was sampled at $100ms$ in the real vehicle, and the simulator was configured to provide the same rate.

Please note that the work does not aim to reduce the *reality gap* understudy, improving the *sim-to-real* transfer with modifications to the simulator environment or changing the MCP movement model. Instead, the work formally evaluates the behaviour differences a posteriori to quantify whether the proposed simulator and design are, in general, valid for developing real-world trajectory tracking systems for autonomous vehicles. For example, the procedure may be helpful if the *reality gap* assessment is below a threshold set by experts. Thus, in the proposed scenario, an MPC steering controller bases on artificial-potential fields, described in Sections 3.3 and 3.4, was implemented with an interactive process under simulation and then compared the behaviour in the two environments, simulation and real. The comparison is made in terms of lateral error and controls action variation, as is described in Section 4.

3.1 Autonomous driving platform

The autonomous driving prototype is a commercial Citröen C4 with no low-level access to any of the actuators. A complete rundown of the automation systems and their characteristics can be found in [11].

The vehicle is equipped with a MPU 9255 Inertial Mass Unit (IMU) and a BMP180 altimeter, rendering a 10 degrees of freedom sensing system. In our control process, only the accelerometer and gyroscope of the IMU are evaluated. Car odometry is obtained through the Control Area Network (CAN) bus, and is read by the computer that implements the control algorithm as a state input. Position feedback is provided by a kinematic differential GPS, combined with the odometry of the vehicle by an Extended Kalman Filter (EKF). A DC motor with an EPOS driver is in charge of actuating over the steering wheel and positions it at the angle determined by our control system.

System state, i.e. vehicle variables, are used as input of the control architecture. While the hardware installed in the car permits acquiring plenty of information about its full state, we have limited the control inputs to those than can be faithfully recreated within the simulator, as to enable comparisons between it and the real world.

The simulation suite that powers this work is based on CARLA (CAR Learning to Act). Our workflow within the simulator is based on control development. A real world route is chosen as a path to benchmark our controller. which is then captured with software tools and follows an ingest process in order to adapt real world data from Geographical Information Systems (GIS) sourced from OpenStreetMap to a set of waypoints that can be recreated within the simulator. This process, paired with a faithful model of the environment of the path allows us to construct maps that accurately represent the conditions in which the car will be found when real world tests are performed.

This software structure allows for monitoring both the state of the car and its behavior in simulation in real time from within the vehicle itself. This feature has been specifically implemented to provide a real-time comparison between the simluation and its real counterpart, and allows for quick assessment of strange dynamics and sources of error.

All tests have been performed on an Intel i7-7700K platform, equipped with 32 GB of RAM and a SATA SSD, and under the Ubuntu operating system. This hardware configuration has remained constant through both simulation and real tests, and has been proven to comply with the real-time requirements of the control system.

3.2 Motion model

The behaviour of the MPC heavily relies on the intrinsic model used. In this work, the well-known kinematic bicycle model is used, where the position, heading, and velocity variables represent the system state vector $[x_w(t), y_w(t), \psi(t), V(t)]$ and the front tire angle variable is the control action $\delta(t)$. The model assumes that the vehicle is symmetric and has no motion on the z-axis.

The model is based on estimating the direction of the velocity vector, $\beta(t)$, at the simplified vehicle centre of gravity concerning the vehicle's longitudinal axis. This angle is a function of the vehicle length $l_f + l_r$ and the tire wheel angle $\delta(t)$. Vehicle yaw $\phi(t)$ is evaluated

with the projection on the longitudinal axis of the velocity vector $V \cos(\beta(t))$ and the tangent of the tire wheel angle $\tan(\delta(t))$. The heading $\psi(t)$ is evaluated with the accumulated ϕ previously evaluated, as expressed in (1a).e. Finally, the decomposition of the velocity vector $V\,dt$ concerning the world reference frame allows the evaluation of $[x_w(t), y_w(t)]$ (car position in the world reference), as shown in (1a).(c-d).

$$\beta(t) = \tan^{-1}\left(\frac{l_f\ \tan(\delta(t))}{l_f + l_r}\right) \tag{1a}$$

$$\phi(t) = \frac{v\,dt\ \cos(\beta(t))}{l_f + l_r}\ \tan(\delta(t)) \tag{1b}$$

$$x_w(t) = v\,dt\ \cos(\psi(t) + \beta(t)) + x_w(t-1) \tag{1c}$$

$$y_w(t) = v\,dt\ \sin(\psi(t) + \beta(t)) + y_w(t-1) \tag{1d}$$

$$\psi(t) = \phi(t) + \psi(t-1) \tag{1e}$$

where β is the velocity vector direction relative to the longitudinal axis, ϕ represents the vehicle's yaw angle along its longitudinal axis, ψ is the heading angle, x_w and y_w are the X_w and Y_w components in the global reference system and δ is the tire angle.

Figure 2 illustrates the model behaviour, which is executed recursively to determine the predicted state of the vehicle over the number of tentatives defined by the MPC controller.

Table 1 defines the parameters used on the model for simulation and real conditions. Simulated parameter values are obtained from CARLA simulator databases, and real parameters are measured in the Citroën-C4 real car.

3.3 Artificial potential field design

Potential fields are widely used to develop path planning and trajectory tracking systems. The algorithms that incorporate

Table 1 Model parameters on simulated and real ego-vehicle

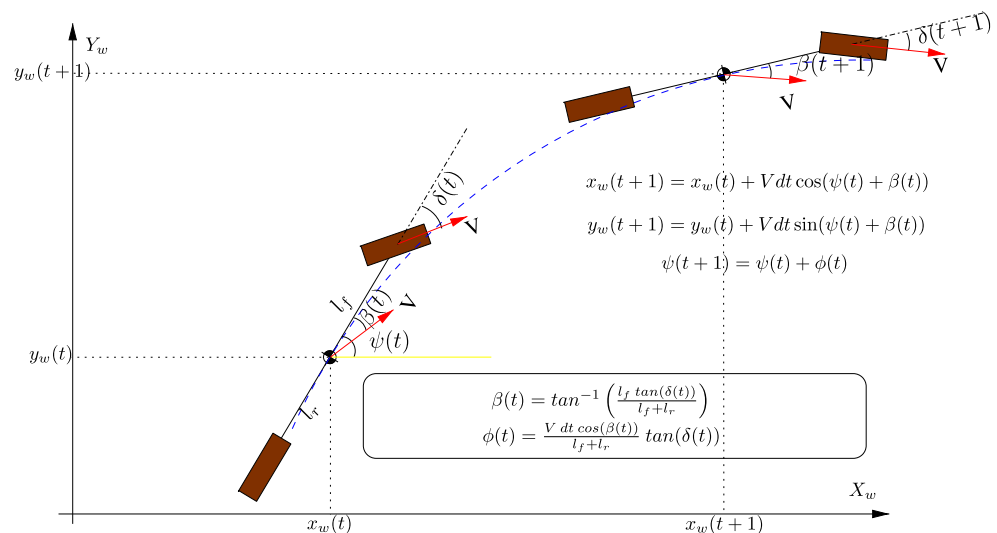| Parameter	Symbol	Simulated	Real
Rear Wheel Base	$l_r\ [m]$	2.82	2.68
Forward Wheel Base	$l_f\ [m]$	1.41	1

this technique base their operation on moving through the minimum of the artificial potential field to achieve optimal planning results. This work integrates potential fields in the cost function of the MPC controller in such a way that the control actions of the prediction horizon move the vehicle through the minimum value of the field. This case is framed within the trajectory tracking paradigm, defining a potential field of Gaussian symmetry. Its minimum value coincides with the road map reference, corresponding with the center of the lane, preventing the vehicle from moving laterally out of the lane. Equation (2) represents the potential field used, where the transversal standard deviation σ_y is a linear function of vehicle speed, as seen in (3). This design makes the field narrow with increasing vehicle speed to improve the convergence of the optimization algorithm, reducing the control actions' increments.

$$Pf_{lane} = -a_{lane}\ e^{-\left(\frac{(y_{tentative} - y_{ref})^2}{2\sigma_y(v)^2}\right)} \tag{2}$$

$$\sigma_y(v)[m] = \begin{cases} 4 \text{ if } v < 30\ ms^{-1} \\ 5.8 - 0.06\,v \\ 1 \text{ if } v > 80\ ms^{-1} \end{cases} \tag{3}$$

The position and orientation of each Gaussian curve are defined on the road map, a set of points equidistant at 10 cm representing the circulation path. Through testing, it was found that a value that yielded good results was $a_{lane} = 2$. The increment $(y_{tentative} - y_{ref})$ is oriented with the heading of the reference. Thus, the proposed

Fig. 2 Motion model included in the MPC controller

solution corresponds to a Gaussian weighting process of the lateral error. The image Fig. 3 represents the potential field assigned to the testing roadmap at a speed of 30km h^{-1}.

3.4 MPC formulation

Predictive control systems base their operation on an optimization process with constraints that use a given model to predict how the system will behave over a discrete control horizon, and deriving optimal control actions from that prediction. When designing MPC controllers that need to generate a control action in less than a given time quantity, as is the case in control systems applied in real systems, it must be ensured that the optimization process converges in less time than specified. For this reason, the number of iterations and the convergence criterion (*ftol*) in the optimization process are the main parameters that fix the computation time of the controller and must be evaluated. If the number of iterations is reduced or the convergence criterion is lax, the computation time is reduced, but the control actions will not be optimal, increasing output error. Therefore, to find a compromise between the number of iterations and the convergence criterion to implement a robust controller in real-world conditions evaluating the *reality gap* is essential. Through testing, it has been validated that the MPC controller proposed in this works fulfills an established requirement of generating a control action every 100ms. This time constraint has been chosen through experience in design and by assessing the needs of a steering controller and the capabilites of the hardware employed.

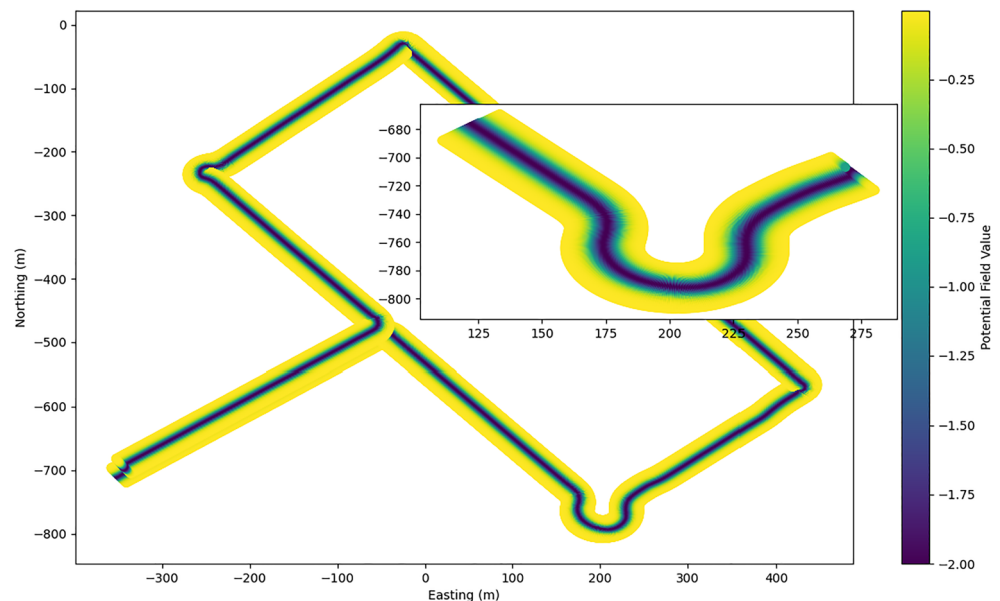Considering the factors mentioned above, the controller is designed with four tentatives (predictions) to cover a prediction horizon of 25m, which is valid for any velocity. The tentatives distribution is non-uniform to improve the controller behaviour. Starting at zero from the vehicle position, the first is at 25/8m, the second at 25/4m, the third at 25/2m and the fourth at 25m, as justified in Section 3.4.1.

If focusing on solving a non-convex optimization problem solutions associated with local minima are generated, differing from an optimal or global solution. The gradient descent algorithm, with momentum descent, which evaluates the rate of change of the variables to be optimized, or random stochastic gradient descent, mitigating the local minima problem. However, the optimization techniques based on Newton's method also offer variants to avoid the problem, although they do not guarantee it either.

The optimization stage of a MPC controller usually involves non-convex oprimization, which leads to search spaces that contain local minima and therefore can render solutions that do not reach the global optimum. Algorithms such as gradient descent (and its variants), which evaluates the rate of change to the variables under optimization, or those of the family of Newton's method, cannot deal with the complicated search space of a non-linear constrained cost function, even though they can be enhanced with techniques like the inclusion of stochastic terms or momentum concepts to improve its exploration capabilities.

Therefore, for cases where a minimization problem with constraints is involved, the techniques described above are not suitable, and therefore linear programming (LP) or non-linear programming (NLP) methods need to be applied. The paper [23] describes a computationally very efficient learning/optimisation algorithm, which is of interest for this work because of its capacity to increase the number of tentatives

Fig. 3 Potential field of real and simulated scenario. Roundabout detailed

for the optimisation process of the MPC controller, ensuring its real time characteristics. The optimisation process is usually based on backpropagation techniques to obtain the optimal fit of any neural system weights. However, there are other alternatives to backpropagation, such as a dendritic event-based processing, which efficiently communicates the error between iterations without sharing weights in the optimisation process. In practice, MPC controllers are solved with NLP algorithms with constraints due to the nonlinear nature of the objective function and constraints. Nowadays, there are a wide variety of optimizers such as sequential linear programming (SLP), sequential quadratic programming (SQP), sequential least squares programming (SLSQP) or generalized reduced gradient (CRG), among others. This family of algorithm relies on a Taylor approximation of the objective function near an initial point, from which the search space is explored. The optimizer used for the MPC controller in this work is SLSQP, due to its reduced computation time compared to other alternatives. The cost function, which includes the value of the potential field, the heading error, and the tire angle increments, is defined as

$$
\begin{aligned}
J = & w_{pf} \sum_{i=1}^{N} Pf_{lane}(i) + \\
& w_{\psi} \sum_{i=1}^{N} (\psi_{car}(i) - \psi_{ref}(i))^2 + \\
& w_{\delta} \sum_{i=1}^{N-1} (\delta(i) - \delta(i-1))^2
\end{aligned}
\tag{4}
$$

where N is 4 and the weights are $1\ m^{-1}$, $0.03\ rad^{-1}$ and $0.2\ rad/s^{-1}$ corresponding with the weights $[w_{pf}, w_{\psi}$ and $w_{\delta}]$, respectively. The constraints considered for the optimization process are classified into two types, tire angle limits and yaw angle limits, and are defined as

$$
\begin{aligned}
& \underset{\delta}{\text{minimize}} \quad J(\delta) \\
& \text{subject to} -60° \le \delta(i) \le 60°, \quad \forall i \in [1, 4] \\
& \qquad\qquad -5° \le \phi(\delta, i) \le 5°, \quad \forall i \in [1, 4]
\end{aligned}
\tag{5}
$$

The control horizon in the MPC implementation is 1. Thus, when the ego-vehicle is in time t, the control action applied in $t + 1$ will be the first tentative assessment by the controller. The control horizon is represented in Fig. 4 in red colour.

3.4.1 Tentative distance distribution analysis

In practice and simulation, it is observed how the controller yields more or less lateral error depending on the approach used in the spatial distribution of tentatives due to the discretisation that the prediction horizon undergoes. The distribution chosen will guarantee the requirements of

driving on any road type with lateral errors less than 30 cm, considering a good value for driving on any road type.

Figure 4 depicts the tested distributions. The uniform distribution is characterized by having equidistant tentatives at 25/4m. The non-uniform distribution considers the first sample at a distance of 25/8m, the second at 25/4m, the third at 25/2m, and the fourth at 25m from the ego vehicle. The uniform distribution shows greater lateral error when the curvature radius of the circulated path decreases, a situation given in curves or roundabouts. The non-uniform distribution reduces errors when driving around curves and roundabouts. The first distance interval is represented in green, and the time increment is fixed for both cases to 0.02 s.

Due to security concerns, the performance of the controller must be assessed firstly on a simulation environment, where a sudden unstability generated by the spatial distribution of the tentatives of the MPC controller might not pose any risk. Once validated in the simulator, any distribution can be recreated with the prototype platform, and therefore undergo a process in which *sim-to-real* transfer viability and *reality gap* can be evaluated.

3.5 Reality gap analysis

Variables related to the controller's performance and comfort are chosen as signals to evaluate the *reality gap* between simulation and real scenarios, since this is the behaviour pattern any simulation must aim to replicate with regards to its real counterpart. The similarity between two continuous signals will be the indicator used to evaluate the *reality gap*. The literature shows a high number of approaches based on correlation measurement to assess the similarity between two continuous signals. Some methods are the Pearson correlation coefficient (PCC), time-lagged cross-correlation (TLCC), windowed TLCC, dynamic time warping, instantaneous phase synchrony, or max normalized cross-correlation (MNCC). This work uses the PCC and the MNCC for *reality gap* assessment.

The PCC indicator represents the covariance of two continuous signals over time and denotes the linear relationship between 0 (not correlated, high *reality gap*) to 1 (perfectly correlated, low *reality gap*). Two essential things were considered in this indicator: (1) outliers can skew the results of the correlation estimation, and (2) the signals are homoscedastic such that the variance is homogeneous across the data range. The PCC is defined as

$$
\rho(X, Y) = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y},
\tag{6}
$$

where X and Y are two continuous temporal signals, μ_x and σ_x are the mean and standard deviation of signal X, and μ_y and σ_y are the mean and standard deviation of signal Y.
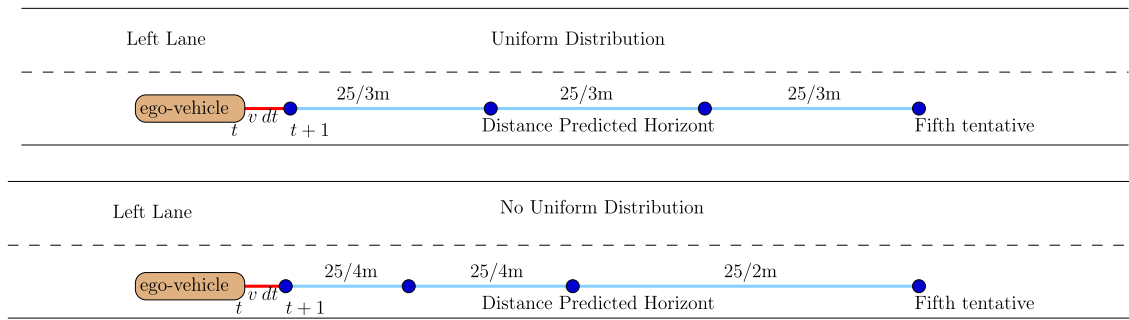
Fig. 4 Spatial tentative distributions

The MNCC measures how similar two continuous signals are. The result ranges between 0 (no similarity) and 1 (perfect similarity). The MNCC is defined as

$$\beta(X, Y) = \frac{X * Y}{max((X * X), (Y * Y))}, \tag{7}$$

where $(X * X)$ and $(Y * Y)$ are the autocorrelation of the X and Y signals, $(X * Y)$ is the cross correlation.

Section 4.1 presents the MPC performance in both settings and compare them with previous work [24]. Finally, Section 4.2 analyses comfort in both cases.

4 Results

This section presents the results and comments from the test of the MPC controller and the *reality gap* assessment, both in real world and simulation environments. Figure 5 illustrates the real-world and simulated circuit. The test environment includes a wide variety of driving elements, such as six straight sections, four curves with a radius of curvature of fewer than ten meter, and three roundabouts. The driving elements are labeled from 01 to 06, and Table 2 links the *Id* with a distance range for each *Id*. In total, the length of the circuit is approximately 3 km. The maximum and minimum traffic speeds are 40km h^{-1} and 20km h^{-1}. The scenario is sufficiently representative with a variability of structural traffic elements, which adds robustness to the obtained results. Further, the proposed controller is compared with an existing MPC controller, developed for path tracking, based on ILQR as found in previous work [24]. Since the ILQR based system does not fulfill any safety criteria, the comparison was carried out only in the simulation suite.

4.1 Reality gap assessment in MPC performances

In order to evaluate the *reality gap*, representative variables need to be selected to compare the performance of both the simulated and real world systems. Thus, from an

experimental point of view it was decided to select the lateral error and the control action, namely the tire angle. Figures 6 and 7 depict the evolution of the lateral error and tire angle for four possible scenarios, two in simulation and two in real-world conditions, according to the two setups described previously regarding spatial tentative distribution. The blue and green signals represent the system's behaviour when the spatial distribution of tentatives are uniform. In contrast, the orange and red signals represent the system's behaviour when the spatial distribution of tentatives is non-uniform. In both cases, there is a notable similarity in the shape of the signals. A structural road image has been overlapped to depict the driven scenario.

Table 3 presents the root mean square lateral errors for segments where there is a curve or roundabout in the circuit; distance intervals are detailed on Table 2. The maximum value of mean square lateral error is 0.42m, evaluated on curve 04 (110° with 5m curvature radius) when the setup is uniform in real-world conditions. In the same curve, the lateral error is reduced to 0.14m when a non-uniform distribution in real-world conditions is evaluated. Therefore, a non-uniform distribution reduces the lateral error up to 3 times. The behaviour in simulation is similar; the maximum lateral error is 0.32m and 0.09m on curve 04 when the setup is uniform and non-uniform tentatives distribution, respectively, reaching an improvement of 3.5 times.

With regards to *reality gap*, uniform tests show more similaritiy than nob-uniform ones, for both indicators. Thus the *sim-to-real* transfer is better when a non-uniform distribution is used. More than 91% and 71% of similarity is measured with PCC and MNCC on uniform tests, respectively. The mean percentage of similarity decreases with non-uniform tentatives in both indicators to 70% and 65%. So, both indicators depict a similarity decrement when a non-uniform tentative distribution setup is compared against uniform distributions. When dynamics are more demanding, the bycicle kinematic model and some CARLA features yield a poor representation of reality. Applying new *sim-to-real* transfer methods will be a solution to reduce the *reality gap* in this setup.

Fig. 5 Real and simulated path

Table 4 presents the root mean square tire angle for segments where there is a curve or roundabout in the circuit; distance intervals are detailed on Table 2. The maximum

Table 2 Driving elements and associated distance ranges

Id	Distance [m]
01	240 - 320
03	960 - 1060
04	1680 - 1760
05	1960 - 2080
06	2300 - 2400

tire angle value is 3:6°, evaluated on curve 04 (110° with 5m 655 curvature radius) when the setup is uniform in real-world conditions. In the same curve, the maximum tire angle value is reduced to 3:5° when a non-uniform distribution in real-world conditions is evaluated, with a difference of 0:1°. While the lateral error is reduced three times, the maximum tire angle is very similar in both cases, which prompts the conclusion that spatial distribution has little effect in the effort the controller needs to enact on the actuators. The analogy with a PID controller is an increment the derivative constant, reaching a faster PID behaviour. However, the maximum tire angle difference on simulation test increases to 0:4° on curve 04, where maximum tire angles

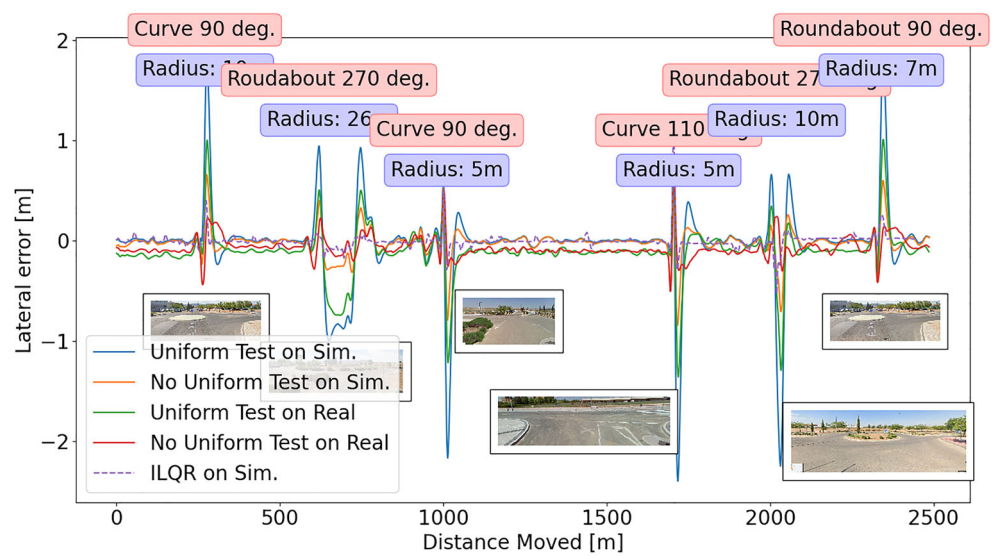**Fig. 6** Lateral error from real-world and simulated tests

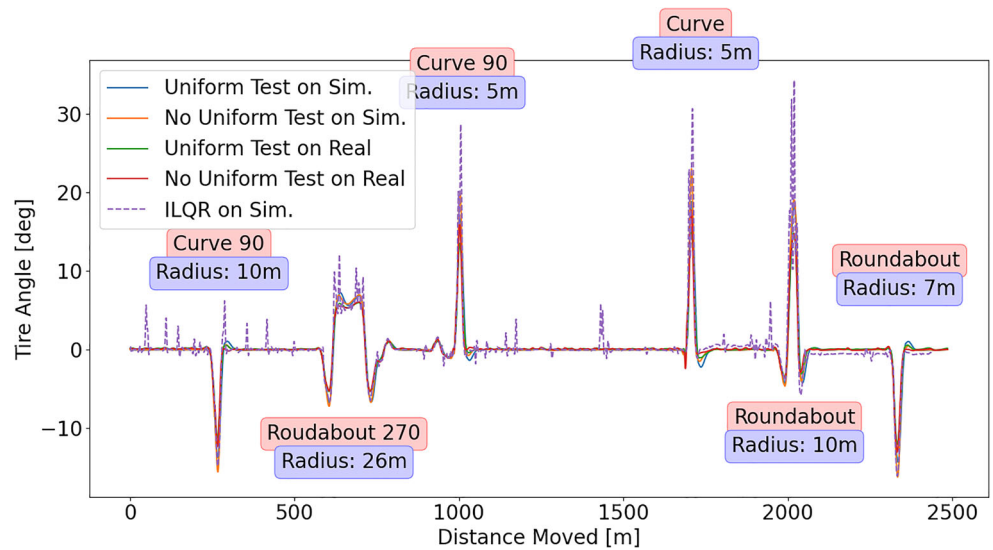**Fig. 7** Tire angle from real-world and simulated test

Table 3 Lateral error and *reality gap* associated

Id	Lateral Error				Reality Gap			
	Real [m]		Sim. [m]		PCC		MNCC	
	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.
01	0.35	0.06	0.19	0.02	0.96	0.5	0.71	0.57
02	−0.17	−0.06	−0.17	−0.06	0.98	0.7	0.95	0.7
03	−0.31	−0.07	−0.32	−0.05	0.91	0.5	0.76	0.59
04	−0.42	−0.14	−0.32	0.09	0.93	0.81	0.76	0.89
05	−0.24	−0.07	−0.21	−0.07	0.98	0.74	0.71	0.79
06	0.31	0.07	0.21	0.02	0.99	0.52	0.76	0.45

Table 4 Tire angle and *reality gap* associated

Id	Tire angle				Reality Gap			
	Real [°]		Sim. [°]		PCC		MNCC	
	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.
01	−3.28	−3.18	−3.13	−3.13	0.99	0.99	0.87	0.86
02	1.21	1.15	0.72	0.66	1.0	1.0	0.93	0.92
03	2.63	2.63	3.58	3.17	0.99	0.99	0.84	0.86
04	3.6	3.51	5.0	6.4	0.99	0.99	0.86	0.86
05	2.34	2.33	2.25	2.35	0.99	0.99	0.88	0.87
06	−2.86	−2.85	−3.11	−2.76	0.99	0.99	0.89	0.88

are 5:9° and 6:3° for uniform and non-uniform distribution, respectively.

As in the previous case, the *reality gap* is studied with PCC and MNCC analysis. The uniform and non-uniform comparatives depict 99% with PCC, regardless of tentative configuration. However, signal correlation analysis shows a mean of 85% for both tentative formats. So, vehicle control actions on real and simulation tests are equal, as evidenced by both indicators. These good *reality gap* results (99% and 85%) prove that the steering control system tested in simulation and real world environments is the same.

To further validate the behaviour of our MPC controller, a comparison against the MPC steering controller explained on [24] is performed. This controller arises as an extension of the well-known linear quadratic regulator (LQR), which can solve both non-linear and constrained problems by linearising and incorporating restrictions in the performance index. The algorithm has a mean computation time of 2 s, not enabling the real-time feature. On the other side, Fig. 7 depicts a non-smooth tire angle dynamic, which yields an uncomfortable and unsafe experience.

4.2 Reality gap assessment in comfort indicators

Although indicators related to lateral error and tire angle were analyzed, these do not evaluate whether the driving actions are comfortable for vehicle occupants. Driving comfort analysis is carried out based on previous work [25], which describes a *modeling occupant's preference metric* for setting a maximum and minimum value of acceleration and jerk along the transversal axis of the vehicle that ensures a comfortable driving experience. Figure 8 depicts the signal jerk assessment on real and simulation tests. The maximum values of acceleration and jerk across the vehicle are -2m s^{-2} and 0.75m s^{-3}, that are within a comfortable driving margin of $\pm$ 4ms $^{-2}$ and 0.9m s^{-3} as described in [25]. These variables depend on linear speed and curves/roundabout topology. A reference velocity of 20kmh^{-1} was used throughout the circuit, except in curve 04 (110° with 5m curvature radius) where it is reduced to 5km h^{-1} The lateral acceleration is not shown due to the smooth results recorded, which reveal no additional information.

Fig. 8 Lateral jerk behaviour. a Blue line: Uniform Test on Simulation. b Orange line: No Uniform Test on Simulation. c Green line: Uniform Test on Real. d Red line: No Uniform Test on Real. e Dash line: ILQR on Simulation. f Yellow line: Comfortable Jerk limits
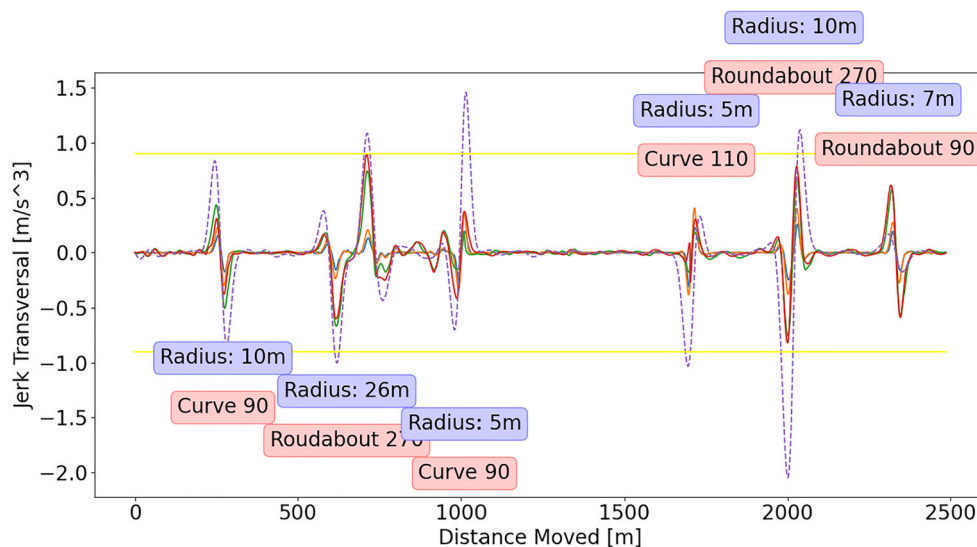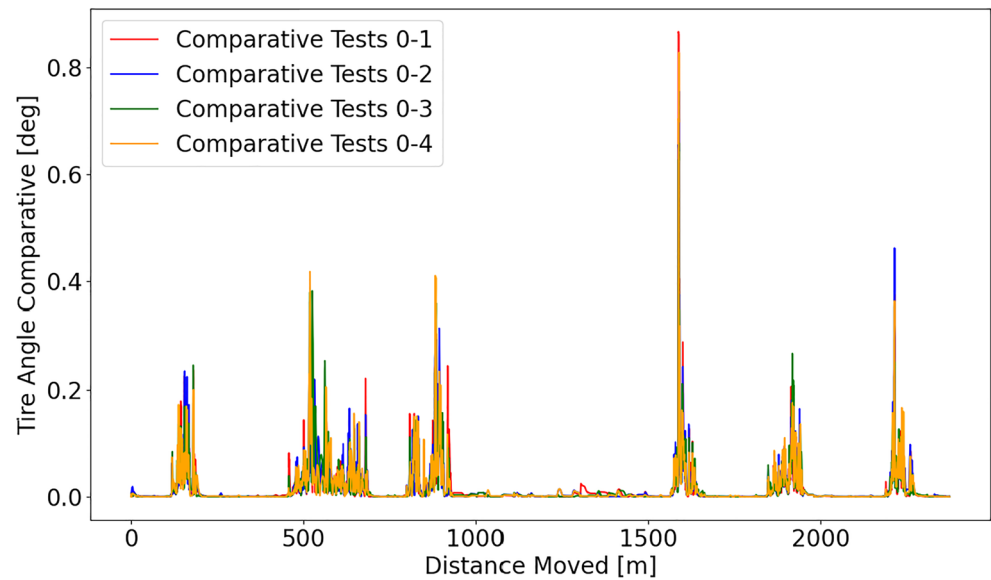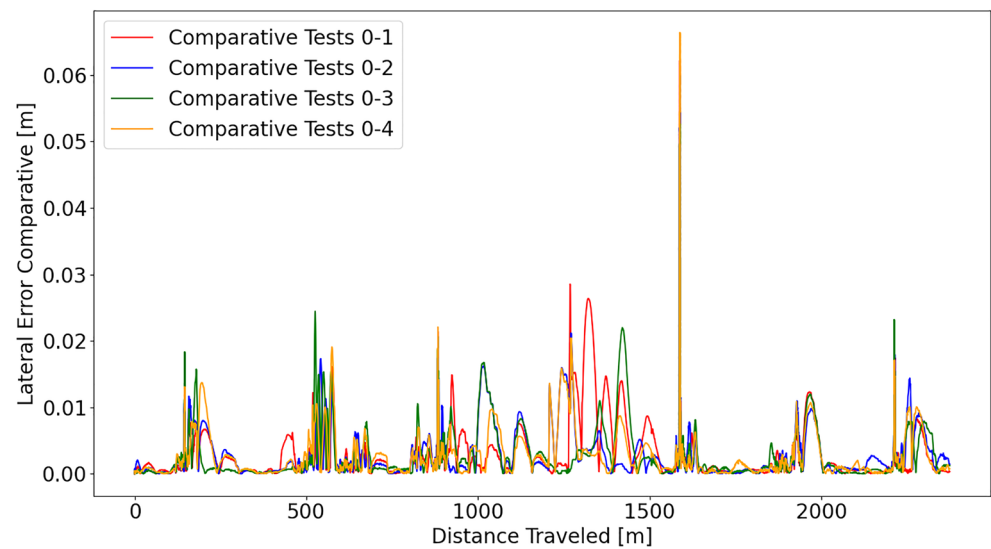
Table 5 Acceleration and jerk across vehicle *reality gap*

Id	PCC Acc. Y		PCC Jerk		MNCC Acc. Y		MNCC Jerk	
	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.	Uni.	NoU.
01	0.93	0.84	0.97	0.96	0.94	0.98	0.61	0.88
02	0.95	0.94	0.86	0.85	0.58	0.55	0.29	0.4
03	0.85	0.98	0.79	0.92	0.63	0.98	0.74	0.89
04	0.97	0.62	0.99	0.69	0.81	0.58	0.93	0.55
05	0.98	0.98	0.96	0.97	0.92	0.86	0.55	0.7
06	0.99	0.98	0.98	0.96	0.89	0.86	0.5	0.68
μ	0.94	0.89	0.93	0.89	0.8	0.8	0.6	0.68

Fig. 9 Tire angle comparative; five laps run

Fig. 10 Lateral error comparative; five laps run

As in the previous case, the *reality gap* is studied with PCC and MNCC analysis, separating uniform and non-uniform tests. Table 5 shows similarity percentage values for each segment on the test; the last row depicts the mean column value for every test. The Pearson indicator shows a mean value higher than 89% in all tests. On the other hand, MNCC analysis shows 80% of similitude on lateral acceleration for uniform and non-uniform configuration. Furthermore, 60% of similitude on jerk across vehicle with a uniform distribution is reached, except for in roundabout 02 where the similitude value is 29%, with a value of 0.6m s^{-3}. Finally, 68% of similitude on jerk across vehicle with a non-uniform tentatives distribution is reached, but with a similar exception as in the previous case with worse results from roundabout 02 where similitude value is 40% due to a signal amplitude difference. This results show that the behaviour of both the real and virtual MPC controllers differs noticeably in regards to comfort and safety, which is further reinforced by the results shown in Table 3.

The controller derived in [24] does not fulfill transversal jerk comfort constraints with maximum values of $2:5\text{ms}^{-3}$ as shown in Fig. 8.

Finally, a robustness and repeatability analysis of the MPC controller is depicted in the Figs. 9 and 10, showing the difference in the behaviour of the tire angle and the lateral error when the test circuit is run five times in a simulation environment. The comparisons are made concerning the first test carried out. The results achieved are an average over the distance travelled with a value of 0.2^o to the tire angle and $1cm$ to the lateral error saved by the controller. We believe that the results demonstrate the repeatability of the controller's behaviour and its robustness.

5 Conclusions and future work

This paper adresses a novel measurement technique for *sim-to-real* transferability in the realm of autonomous driving. A statistic analysis of acquired data in both real-world and simulation environments has been showcased in order to improve the validation stage of the design process of a steering controller, which in our case has been exemplified with the use of an MPC system based on adaptive potential fields. The proposed framework presents a procedure in which both data streams are studied to provide a degree of correlation between the behaviour of the system in a simulated environment and its real counterpart, so that the designer can have an accurate expectation of how the real system will perform after having been designed in a full-software workflow.

The statistical concepts used to evaluate the *reality gap* are the Pearson correlation coefficient (PCC) and max normalized cross-correlation (MNCC). These metrics

are applied over ego-vehicle behaviour (steering, lateral error) and comfort signals (lateral acceleration, jerk). The behaviour of the controller is evaluated on lateral error and tire angle, and comfort is evaluated using lateral acceleration and jerk. Our studies rendered a PCC of 99% and a MNCC greater than 85% between both tire angle signals, and a PCC of 85% and MNCC of 93% when analysing lateral jerk. The similarity results evidence that simulation approaches, such as a complete design procedure within a simulation suite such as CARLA are beneficial to designing trajectory tracking systems.

Another contribution of this work is the study of control tentative distribution over the prediction horizon of the proposed MPC controller. Our tests show that, while both configurations render a capable and safe controller, non-uniform distributions reduce lateral error up to 32%, achieving a maximum lateral error of 60 cm in the most challenging point of the test track. However, correlation results on both PCC and MNCC are suboptimal, which prompts a future line of work that derives a specific set of tests and metrics to assess a method of correction within the simulation environment to improve the design process when applying a non-uniform spatial distribution.

Comfort analysis is crucial in a simulation design process when a path tracking system is designed. Hence, the paper identifies limits on lateral acceleration and jerk values to assess the performance of the proposed controller in what relates to percieved comfort of the occupants of the vehicle. As demonstrated by the extensive analysis of results present in previous sections, our controller performs adequately and stays below the range that has been identified as comfortable via real-world testing. Besides, the correlation results pertaining both signals involved further reinforce the benefits of a full-software desing. On this note, a comparison between the proposed MPC controller and a ILQR based one has been carried out. The results of this comparison seem to indicate that our implementation has considerable benefits over an optimal control derived one, especially regarding passenger comfort and safety.

A clear benefit of taking into account the *reality gap* into the desing process of the MPC controller arises from the possibility of extending its capabilities beyond trajectory tracking with the inclusion of other autonomous driving subsystems. One of our future lines of work will reside in the design of a path planning system that responds to traffic issues such as obstacle avoidance and overtaking maneuvers. Other future improvements to the system include the implementation of a more complex movement model, which would further reduce *sim-to-real* transferability, and an enhancement of the software implementation, which would lower computational demand and therefore allow to extend the number of tentatives the MPC controller can consider or the number and nature of

the constraints included in its optimization stage, which would render a safer, more comfortable experience. In this regard, the ILQR implementation described previously, which we compared our proposed controller against, includes a novel implementation using auto-gradients and both backward and forward differentiability which makes for computation periods in the microseconds range, and therefore allows to substantially increase the number of tentatives considered. A similar implementation of the optimization procedure on our MPC controller has been identified as a future line of work.

If a sufficiently powerful *reality gap* analysis is performed, one of its main consequences resides in the design of neural steering controllers in a full-software setup, based on the beahviour of an MPC controller that has been sufficiently adapted to replicate the performance of a real implementation. With this controller, a wide array of driving situations can be designed and tested in order to build a database with which to train any sort of neural control structure that would further improve the real-time capabilities of the system.

References

1. Fernández Llorca D, Gómez E (2021) Trustworthy autonomous vehicles. EUR 30942 EN, Publications Office of the European Union, Luxembourg, JRC127051, Tech. Rep.

2. Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E (2016) A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans Intell Veh 1(1):33–55

3. González D, Pérez J, Milanés V, Nashashibi F (2016) A review of motion planning techniques for automated vehicles. IEEE Trans Intell Transp Syst 17(4):1135–1145

4. Waymo O (2021) Waymo safety report. Waymo LLC, Tech Rep

5. Baidu Apollo team (2017) Apollo: open source autonomous driving, howpublished = https://github.com/apolloauto/apollo. note = Accessed: 2019-02-11

6. Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V (2017) CARLA: an open urban driving simulator. In: Proceedings of the 1st annual conference on robot learning, pp 1–16

7. Shah S, Dey D, Lovett C, Kapoor A (2017) Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics. [Online]. Available: https://arxiv.org/abs/1705.05065

8. Åkerberg O, Svensson H, Schulz B, Nugues P (2003) Carsim: an automatic 3d text-to-scene conversion system applied to road accident reports. In: Proceedings of the research notes and demonstrations of the 10th conference of the european chapter of the association of computational linguistics. Association of computational linguistics, pp 191–194, 10th conference of the European chapter of the association of computational linguistics; Conference date: 12-04-2003 Through 17-04-2003

9. Koenig N, Howard A (2004) Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS) (IEEE Cat. no.04CH37566), vol 3, pp 2149–2154

10. Kang Y, Yin H, Berger C (2019) Test your self-driving algorithm: an overview of publicly available driving datasets and virtual testing environments. IEEE Trans Intell Veh 4(2):171–185

11. Parra Alonso I, Izquierdo Gonzalo R, Alonso J, García Morcillo A, Fernández Llorca D, Sotelo MA (2018) The experience of drivertive-driverless cooperative vehicle-team in the 2016 gcdc. IEEE Trans Intell Transp Syst 19(4):1322–1334

12. Zhao W, Queralta JP, Westerlund T (2020) Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE Symposium series on computational intelligence (SSCI), pp 737–744

13. Salvato E, Fenu G, Medvet E, Pellegrino FA (2021) Crossing the reality gap: a survey on sim-to-real transferability of robot controllers in reinforcement learning. IEEE Access 9:153171–153187

14. Valassakis E, Ding Z, Johns E (2021) Crossing the gap: a deep dive into zero-shot sim-to-real transfer for dynamics. IEEE, https://doi.org/10.1109/IROS45743.2020.9341617

15. Chu Y-JR, Wei T-H, Huang J-B, Chen Y-H, Wu I-C (2020) Sim-to-real transfer for miniature autonomous car racing. arXiv:2011.05617

16. Stocco A, Pulfer B, Tonella P (2021) Mind the Gap! a study on the transferability of virtual vs physical-world testing of autonomous driving systems. arXiv:/2112.11255

17. Wurts J, Stein JL, Ersal T (2018) Collision imminent steering using nonlinear model predictive control. In: 2018 Annual American control conference (ACC), pp 4772–4777

18. Palmieri G, Barbarisi O, Scala S, Glielmo L (2009) A preliminary study to integrate ltv-mpc lateral vehicle dynamics control with a slip control. In: Proceedings of the 48h IEEE conference on decision and control (CDC) held jointly with 2009 28th Chinese control conference, pp 4625–4630

19. Kim D, Kim H, Huh K (2017) Local trajectory planning and control for autonomous vehicles using the adaptive potential field. In: 2017 IEEE Conference on control technology and applications (CCTA), pp 987–993

20. Lu B, Li G, Yu H, Wang H, Guo J, Cao D, He H (2020) Adaptive potential field-based path planning for complex autonomous driving scenarios. IEEE Access 8:225294–225305

21. Lattarulo R, Pérez Rastelli J (2021) A hybrid planning approach based on mpc and parametric curves for overtaking maneuvers. Sensors 21(2). [Online]. Available: https://www.mdpi.com/1424-8220/21/2/595

22. Hidalgo C, Lattarulo R, Pérez J, Asua E (2019) Hybrid trajectory planning approach for roundabout merging scenarios. In: 2019 IEEE International conference on connected vehicles and expo (ICCVE), pp 1–6

23. Yang S, Gao T, Wang J, Deng B, Lansdell B, Linares-Barranco B (2021) Efficient spike-driven learning with dendritic event-based processing. Frontiers in Neuroscience 15. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.601109

24. Chen J, Zhan W, Tomizuka M (2017) Constrained iterative lqr for on-road autonomous driving motion planning. In: 2017 IEEE 20th International conference on intelligent transportation systems (ITSC), pp 1–7

25. Bae I, Moon J, Jhung J, Suk H, Kim T, Park H, Cha J, Kim J, Kim D, Kim S (2020) Self-driving like a human driver instead of a robocar: personalized comfortable driving experience for autonomous vehicles. arXiv:2001.03908

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Iván García Daza received the MSc and PhD degrees in Telecommunications Engineering from the University of Alcalá (UAH), Madrid (Spain), in 2004 and 2011 respectively. Since 2017, he has been an Assistant Professor with the Computer Engineering Department, University of Alcalá. Since 2007, he has been a member of the INVETT research group. He is the author of more than 25 articles. His research interests include LiDAR odometry, 3D Accuracy Maps building, Optimal Control Theory applications applied to Intelligent Transportation Systems. He is an Associate Editor of the journal *Sensors*, and holds four patents. He has collaborated on more than 20 public and private projects related to computer science techniques applied to Intelligent Transportation Systems.

Rubén Izquierdo received the bachelor's degree in electronics and industrial automation engineering, the M.S. degree in industrial engineering, and the Ph.D. degree in information and communication technologies from the Universidad de Alcalá (UAH), in 2014, 2016, and 2020, respectively. He is currently Assistant professor in the Computer Engineering Department, UAH. His research interests include the prediction of vehicle behaviors but also in control algorithms for highly automated and cooperative vehicles.

Luis Miguel Martínez was born in Alcalá de Henares, Madrid, Spain, in 1997. He recieved the B.S. degree in electronics engineering and industrial automation from the University of Alcalá, Madrid, in 2022, and is currently enrolled in a M.S. program in industrial engineering.

Since early 2021 he has been a Research Assistant with the Intelligent Vehicles and Transport Technologies in the University of Alcalá. He has cooperated in several lines of research, focusing on Model Predictive Control with Autonomous Driving Applications, as well as co-authored several articles and papers. He also presented a B.S degree project on the capabilities of metaheuristic optimizers applied to MPC conrollers.

Ola Benderius received the PhD degree in Machine and Vehicle Systems from Chalmers University of Technology, Gothenburg (Sweden), in 2014. He is Associate Professor of Autonomous Mobile Systems at Chalmers Department of Mechanics and Maritime Sciences. His research focus is mainly on biologically inspired AI for mobile autonomous vehicles, both inspired by human driving and deep neural functions connected to self-motion and machine operation. He did in 2014 find the solution to a 70-year old mystery connected to human control and the so called remnant within control theory, by using inspiration from biological studies. In 2015 he joined the Chalmers vehicle research laboratory Revere, where he is currently leading research connected to autonomous driving, computer vision, software engineering in cyberphysical systems, and next generation HMI for both road and marine vehicles.

David Fernández Llorca (Senior Member, IEEE) received the Ph.D degree in telecommunication engineering from the University of Alcalá (UAH) in 2008. He is currently Scientific Officer at the European Commission - Joint Research Center. He is also Full Professor with UAH. He has authored over 130 publications and more than 10 patents. He received the IEEE ITSS Young Research Award in 2018 and the IEEE ITSS Outstanding Application Award in 2013. He is Editor-in-Chief of the IET Intelligent Transport Systems. His current research interest includes trustworthy AI for transportation, predictive perception for autonomous vehicles, human-vehicle interaction, end-user oriented autonomous vehicles and assistive intelligent transportation systems.

Affiliations

Iván García Daza[1] 📧 **· Rubén Izquierdo[1] · Luis Miguel Martínez[1] · Ola Benderius[2] · David Fernández Llorca[1,3]**

Rubén Izquierdo
ruben.izquierdo@uah.es

Luis Miguel Martínez
lmiguel.martinez@edu.uah.es

Ola Benderius
ola.benderius@chalmers.se

David Fernández Llorca
david.fernandezl@uah.es

[1] Computer Engineering Department, University of Alcalá, Alcalá de Henares, 28805, Madrid, Spain

[2] Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden

[3] Joint Research Center, European Commission, Seville, 41092, Andalucia, Spain