Turkish Journal of Electromechanics and Energy
ISSN: 2547-975X
Vol:9, No:3, Page 114-123 (2024)
https://www.scienceliterature.com

Enhanced trajectory tracking for autonomous vehicles using a nonlinear model predictive controller

H. Belkebir[1,*], T. Belkebir[1]

[1]*Laboratory of intelligent systems geo-resources and renewable energy, University of Sidi Mohhamed Bno Abddelah, Fez, Morocco*

ARTICLE INFO

Article Type:
Research Paper

Article History:
Received: 03 December 2024
Revised: 29 December 2024
Accepted: 30 December 2024
Published: 31 December 2024

Editor of the Article:
M. E. Şahin

Keywords:
Autonomous driving, Path tracking, Nonlinear optimization, Direct collocation

ABSTRACT

This study investigates the application of advanced control and optimization techniques to improve the trajectory-tracking performance of autonomous vehicles. A nonlinear model predictive controller framework is utilized, incorporating a direct collocation method enhanced with multiple shooting techniques to address the challenges of nonlinear dynamics and constraints. The trajectory-tracking problem is formulated using the bicycle model. The reference path is generated by the Carla Simulator which provides realistic and varied test scenarios for evaluating the proposed approach. The problem is transformed into a nonlinear optimization task using the Julia modelling package and solved with an interior point algorithm to ensure computational efficiency and precision. The results demonstrate that the method achieves trajectory-tracking accuracy under 1 meter in the longitudinal direction and a control response time of fewer than 100 milliseconds, validating its capability to operate in real-time under dynamic conditions.

Cite this article: H. Belkebir, T. Belkebir, "Enhanced trajectory tracking for autonomous vehicles using a nonlinear model predictive controller," *Turkish Journal of Electromechanics & Energy*, 9(3), pp. 114-123, 2024.

1. INTRODUCTION

Driver inattention, the primary cause of accidents [1-3], presents a significant challenge that can effectively be mitigated by driving assistance systems [4]. These advanced systems have the potential to significantly reduce the incidence of accidents caused by human error in both urban and rural environments, offering a promising outlook for road safety [5]. The foundation of the advanced driver assistance systems (ADAS) solutions lies in selecting the appropriate controller tailored to the level of automation that we plan to integrate into the vehicle [6, 7]. Trajectory tracking is critical to modern driving assistance systems [8]. It provides two essential functions: planning a feasible trajectory and providing the necessary control input that keeps the vehicle on the trajectory while meeting safety and comfort requirements [9].

Two kinds of approaches are used to produce a reliable and efficient trajectory-tracking algorithm [10]:

• End-to-end framework that combines the planning and control tasks in one function using reinforcement learning techniques.

• Hierarchical framework that proposes planning and control tasks in separate modules. The former module focuses on global and strategic decisions, while the latter specializes in operative manoeuvres.

In the Hierarchical approach, we will assume that the reference trajectory is given by the path planning module, it includes complete information about the desired vehicle state based on vehicle and environment limitations at every time frame. Therefore, the controller module of the ADAS system takes over and calculates a series of actions that the vehicle system should execute to reach the goals set by the high-level module. These operations are repeated during the vehicle's motion until it attains its destination [11-14].

In this process, the choice of the controller is a critical task since it's supposed to take the vehicle's state as input and produce the necessary and adequate actions to ensure that the car follows the local path accurately and safely.

Various approaches have been developed to address the task of trajectory tracking, including proportional-integral-derivative (PID), linear quadratic regulator (LQR), and model predictive control (MPC). These control strategies are well-regarded for their robustness and efficiency in solving trajectory-tracking problems [15, 16]. However, PID controllers are unsuitable for multi-input multi-output (MIMO) systems, and LQR controllers struggle to handle systems operating under constraints or exhibiting nonlinear behaviour [17, 18].
In contrast, MPC utilizes a mathematical model of the system to predict its future states. It selects the first control input from the calculated sequence before iterating this process until the desired final state is achieved. A pivotal element of the MPC framework is the trajectory tracking optimization (TTO) method, which determines optimal control inputs at each iteration. This method

**Corresponding author's e-mail:*hicham.belkebir@usmba.ac.ma

addresses challenges in complex, high-dimensional systems by optimizing a performance index while ensuring compliance with system constraints, enabling dynamically feasible motion [19, 20].

To solve the continuous TTO problem, both direct and indirect methods have been proposed, particularly in the context of autonomous vehicles (AVs) [21]. This study focuses on a direct transcription approach, which involves discretizing state and control trajectories over time. This process reformulates the original problem into an augmented representation based on trajectory values at discrete points (knots), with constraints imposed to enforce system dynamics between these knots. A nonlinear programming (NLP) solver is then employed to solve the transcribed optimization problem.

In this work, we will evaluate the effectiveness of direct methods, particularly the transcription scheme used to convert the continuous TTO formulation into a discrete one. Specifically, we analyze the performance of two widely used methods: direct multiple shooting (DMS) and direct orthogonal collocation (DOC), within realistic scenarios simulated in Carla [22]. Typically, DOC is implemented using a derivative formulation with non-uniform collocation points across each time step of the MPC horizon.

Furthermore, this study aims to enhance the DOC method by employing a cumulative integral form combined with a uniform distribution of collocation points, low-order polynomial interpolation, and shorter MPC time steps. These adjustments aim to reduce instability and align the performance with the classical DOC and DMS methods. We named the resulting method the implicit multiple shooting direct collocation method (IMSDOC).

The primary goal of this study is to enhance the trajectory-tracking performance of autonomous vehicles using a nonlinear MPC framework. The trajectory tracking problem is modelled using the kinematic bicycle model, which captures key vehicle dynamics with moderate complexity. Additionally, we will investigate the effectiveness of a nonlinear optimization algorithm, specifically the interior-point method implemented in the interior-point solver [23], to resolve the nonlinear problem resulting from the transcription process.

While direct transcription offers significant advantages, it also presents notable challenges. One major drawback is the increased size of the resulting NLP problem, which leads to high computational demands. Additionally, the solution's accuracy is critical, as the system's behaviour is only approximated between knots. This limitation can render the generated plans infeasible in real-world applications. Thus, evaluating the impact of dynamic constraints, the number of discretization nodes, and the choice of NLP solvers on the solution's accuracy and quality is crucial. However, this study doesn't claim that it will conduct a detailed evaluation of the selected methods. Instead, it will demonstrate the effectiveness of the IMSDOC method in realistic driving scenarios using the Carla simulator, focusing on computational time, solution feasibility, and tracking error metrics.

To cover all these topics, the paper is organized into several sections. The material and methods section will focus on formulating the tracking trajectory problem as a continuous-time TTO problem. Then, delve into the transcription process by covering the DMS, DOC, and IMSDOC methods. It will end by presenting the ecosystem used in this study, namely the modelling language, optimization algorithm, and simulation environment configuration and setup. The result section will graphically show the simulation outcomes and validation steps reinforced by the necessary comments and analysis. Finally, we will conclude with an overall assessment of this work and outline future perspectives.

2. MATERIAL AND METHODS

2.1. Trajectory Tracking Problem Formulation

To build a mathematical model for the tracking problem in the context of autonomous driving, we must choose the model of the vehicle from several possibilities including a two or four-wheel model with kinematic or dynamic behaviour. Assuming a soft driving behaviour, we have selected the kinematic bicycle model.

2.1.1. Kinematic bicycle model

As illustrated in Figure 1, the directional wheel of the vehicle is located at point A in the middle of the front axle.

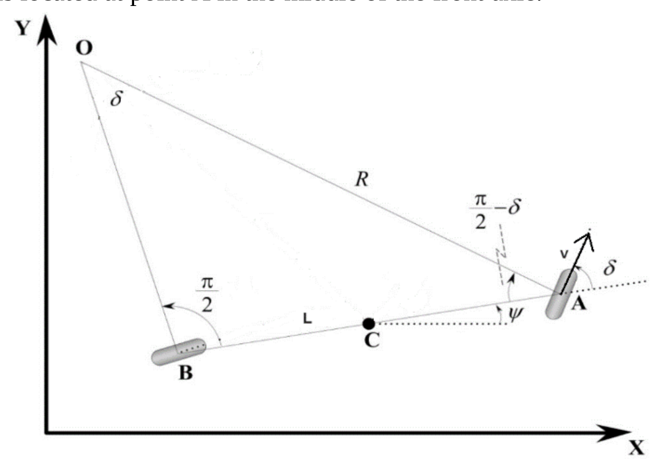


Fig. 1. Kinematic bicycle model.

The centre of this wheel plays also the role of the origin of the local referential frame attached to the vehicle's body. We limited the car displacement to a planar and non-holomorphic motion described by global coordinates (x, and y) at every time frame. Let v be the longitudinal speed, ψ the heading angle, δ the steering angle controlling the orientation of the fronted wheel with respect to the baseline direction, and a longitudinal acceleration of the vehicle. We also supposed that for a moderate rolling speed, the dynamic effects of external forces, such as aerodynamic friction and road friction, acting on the car's body, particularly on its wheels, would become negligible.

2.1.2. Vehicle dynamics

Based on all the above and by applying the basic principle of kinematic, we obtained the system of equations that describes the dynamic of the vehicle as in Equation (1).

$$\dot{X} = f(X,U) = \begin{cases} \dot{x} & = & v\cos(\psi+\delta) \\ \dot{y} & = & v\sin(\psi+\delta) \\ \dot{\psi} & = & \dfrac{v}{L}\sin(\delta) \\ \dot{v} & = & a \end{cases} \tag{1}$$

With $X = [x,\ y,\ \psi,\ v]^T$ the vehicle state vector, $U = [\delta,\ a]^T$ the control input vector, and $f(X,\ U)$ a multivariate nonlinear function representing the car's dynamic during longitudinal and lateral movement. L is the baseline length of the vehicle.

2.1.3. Continuous tracking trajectory problem

Let $\mathcal{T}$ be the reference trajectory generated by the path planning module. This trajectory is a continuous set of targeted states $X_r(t)$ which the ego vehicle must reach with a minimal tracking error defined by Equation (2).

$$\epsilon \ = \ X(t) \ - \ X_r(t) \tag{2}$$

We also used a Lagrange form of the cost function in Equation (3) that quantifies the trajectory's tracking performance:

$$\mathrm{J} = \int_{t_0}^{t_f} \epsilon(t)^T Q \epsilon(t) + \mathrm{U}(t)^T \mathrm{R} \mathrm{U}(t) \mathrm{d}t \tag{3}$$

With Q a positive definite matrix and R a semi-definite positive matrix. The tracking problem can be now formulated as a continuous optimization problem in which the goal is to calculate efficiently the most favourable sequence of viable states and admissible control inputs $(X^*,\ U^*)$ by solving the problem described by Equations (4) to (7).

$$\min_{X,U} \ J \tag{4}$$

Subject to:

$$X(0) \ = X_0 \tag{5}$$

$$\dot{X}(t) = f(X(t), U(t), t) \tag{6}$$

$$X(t) \in \mathcal{X},\ U(t) \in \mathcal{U} \tag{7}$$

$\mathcal{X}$ is the set of feasible states and $\mathcal{U}$ is the set of admissible control inputs.

2.2. Transcription of the Continuous Optimization Problem

To tackle the problem Equations (4) to (7), we utilize direct transcription, which involves discretizing the trajectory-tracking problem before optimizing control inputs. We explore two standard methods: direct multiple shooting (DMS) and direct orthogonal collocation (DOC). Both techniques have been widely used in continuous optimization problems. Direct multiple shooting divides the time horizon into shoots and defines control inputs and state variables at shoot boundaries. The optimizer then calculates optimal control inputs and state trajectories that adhere to system dynamics and constraints. Direct orthogonal collocation, on the other hand, uses collocation points to enforce system dynamics. The optimizer determines optimal inputs and trajectories that satisfy constraints at these points. Both approaches are thoroughly investigated and used in transcribing analogous continuous optimization problems. In this section, we'll outline the essential implementation steps for these methods before addressing the optimization problem.

2.2.1. Direct multiple shooting techniques

The direct shooting method (DMS) is a numerical technique commonly used to solve optimal control and state estimation problems in dynamic systems. When applied to trajectory tracking problems using MPC, the process involves:

a) *Discretization of the horizon T*: The horizon T is equally segmented into N shoots (subintervals) $[t_n, t_{n+1}]_{n=0,\dots,N-1}$ at each MPC iteration. On each shoot n, we define the following first-order ordinary differential equation (ODE) $\dot{X}_n(t) = f\big(X_n(t),\ U_n(t)\big)$

b) *Solving the ODE:* We then solve these ODEs on each shoot independently, using the explicit fourth-order Runge-Kutta method to obtain the state of the system $X_n(t)$. The initial conditions are treated as free variables determined by the optimization process.

c) *Imposing continuity at shoot's boundaries*: To ensure continuity of the system state between successive shoots, we impose matching conditions such that $X_n(t_{n+1}) = X_{n+1}(t_{n+1}) \quad \forall; n = 0, \dots, N-1$. This transformation turns the initial problem into a large-scale optimization problem, where we simultaneously optimize the state and control variables for all shoots

d) *Discrete optimization problem formulation*: This involves converting the optimal control problem into minimizing a cost function evaluated at specific time points. Constraints are imposed at these discrete time points to ensure adherence to system dynamics, continuity conditions, and feasibility requirements, which leads to the problem described below:

$$\min_{X_0,U_0,\dots,X_N,U_{N-1}} \sum_{n=0}^{N-1} (\epsilon_n^T Q \epsilon_n + U_n^T R U_n) + X_N^T Q_f X_N \tag{8}$$

Subject to:

$$X_0 = X_{r,0} \tag{9}$$

$$X_{n+1} - X_n = RK_4(X_n, U_n), \quad \forall n = 0, \dots, N-1 \tag{10}$$

$$X_n \in \mathcal{X}, \quad \forall n = 0, \dots, N \tag{11}$$

$$U_n \in \mathcal{U}, \quad \forall n = 0, \dots, N-1 \tag{12}$$

2.2.2. Direct orthogonal collocation method

The direct orthogonal collocation approach begins by dividing the MPC horizon into N particular instants called knots. Then it introduces an M collocation point between successive knots and finally approximates the vehicle's state and control input using an orthogonal polynomial function, Lagrange polynomial in our cases. The details of its implementation are listed below:

a) Segmentation of the horizon T: The MPC horizon is divided into N disjointed segments, and at each of them, we introduce M collocation points and define a list of vehicle's state $\tilde{X}_n$ that belongs to this subinterval, such that $\tilde{X}_n = \big[X_{n,0},\ X_{n,1}, \dots,\ \ X_{n,M-1}\big]$. Considered a control input on each segment and is represented by the vector U_n for $n = 0, .., N-1$.

b) *Elaborating collocation equations:* Firstly, we interpolate the vehicle state with Lagrange polynomial $\ell_{n,m}(t)$ and use a piecewise constant input control:

$$X_n(t) \approx p_n(t, \tilde{X}_n) = \sum_{n=0}^{M} X_{n,m}\, \ell_{n,m}\left(\frac{t-t_n}{t_{n+1}-t_n}\right) \tag{13}$$

$$\mathrm{U_n} = cte \quad \forall\, t \in [t_n, t_{n+1}] \tag{14}$$

Then we build the collocation equations as a constraint set:

$$\dot{p}_n(t_{n,m}, \tilde{X}_n) = f(X_{n,m}, U_n)\ \forall\,(n,m) \tag{15}$$

c) *Development of the discrete optimization problem*: The optimal control problem is reformulated into a nonlinear optimization problem (NLP), where the objective is to minimize an objective function like that used in DMS under the constraints imposed by the collocation equations and possibly additional constraints on the system states or control inputs that leads us to the discrete nonlinear optimization problem presented in Equations (16) to (20):

$$\min_{X,U} \sum_{n=0}^{N-1}\left(\sum_{m=0}^{M-1} \epsilon_{n,m}^T Q\, \epsilon_{n,m} + U_n^T R U_n\right) + X_N^T Q_f X_N \tag{16}$$

Subject to:

$$X_0 = X_{r,0} \tag{17}$$

$$f(X_{n,m}, U_n) = \dot{p_n}(t_{n,m}, X_{n,m}), \quad \forall (n,m) \tag{18}$$

$$X_{n,m} \in \mathcal{X}, \quad (n,m) \in [0,\dots,N-1] \times [0,\dots,M-1] \tag{19}$$

$$U_n \in \mathcal{U}, \quad \forall n = 0,\dots,N-1 \tag{20}$$

2.2.3. Direct orthogonal collocation with implicit multiple shooting method

This technique used the same polynomial as his precedent to interpolate the decision variables of the continuous optimization problem. However, instead of approximating the vehicle state, it estimates the vehicle dynamics according to Equation (21).

$$\dot{X}_n(t) = \sum_{m=0}^{M} \{f(X\}_{n,m}, U_n)\mathrm{l}_{n,m}\left(\frac{t-t_n}{t_{n+1}-t_n}\right) \tag{21}$$

The collocation set of equations becomes:

$$X_{n,m} = X_{n,0} + \sum_{k=0}^{M} a_{n,k} f(X_{n,k}, U_n) \quad \forall m = 1,\dots,M \tag{22}$$

With $a_{n,k} = \int_{t_{n,0}}^{t_{n,m}} \ell_{n,k}\left(\frac{t-t_n}{t_{n+1}-t_n}\right) dt.$

The Equation (22) seems like using an implicit multiple-shooting technique. The resulting discrete optimization problem becomes as described by Equations (23) to (27):

$$\min_{X,U} \left\{\sum_{n=0}^{N-1}\left(\sum_{m=0}^{M-1} \epsilon_{n,m}^T Q\, \epsilon_{n,m} + U_n^T R U_n\right) + X_N^T Q_f X_N\right\} \tag{23}$$

Subject to:

$$X_{0,0} = X_{r,0} \tag{24}$$

$$X_{n,m} = X_{n,0} + \textstyle\sum_{k=0}^{M} \alpha_k f(X_{n,k}, U_n), \quad \forall (n,m) \tag{25}$$

$$X_{n,m} \in \mathcal{X}, \quad \forall (n,m) \in [0, N-1] \times [0, M-1] \tag{26}$$

$$U_n \in \mathcal{U}, \quad \forall n \in [0,\dots,N-1] \tag{27}$$

3. SIMULATION ECOSYSTEM AND SETUP

Solving the optimization problem generated by the transcription methods requires specialized numerical software and hardware.

3.1. Hardware and Software Configuration

3.1.1 Hardware

The simulation will be run on a computer equipped with a quad-core Intel processor clocked at 2.5 GHz, with 24 GB of RAM and an Nvidia MX230 GPU.

3.1.2 Software Configuration

We conducted the simulation for the transcribed TTO problem using the Julia programming language [24] due to its speed, efficient resource management, and optimization capabilities. We utilized the JuMP framework to symbolically formulate the optimization TTO problem [25] and employed the Ipopt.jl package, which provides a wrapper for the interior point algorithm implemented in the interior point C++ solver, to solve it. For a graphical representation of the simulation outcomes, we used the Plots.jl and CairoMakie.jl libraries [26, 27]. Additionally, to use Carla within the Julia programming environment, we incorporated the PyCall.jl package [28].

3.2. Simulation

To solve the tracking trajectory problem, we have conducted the simulation process in two stages.

3.2.1. Numerical simulation stage

At this stage, we have conducted a numerical simulation using the NMPC algorithm. We established a reference horizon and performed an optimization process to determine the optimal control inputs, which we then applied to a model plant using the Runge-Kutta method. This allowed us to update the vehicle state iteratively until we reached the end of the reference trajectory. The results of this simulation are presented in the first part of the results section.

3.2.2. Realistic simulation in Carla

To validate and test the robustness of our model, we used the Carla simulator for the same trajectory as in the numerical simulation. The results of these simulations are presented in the second part of the results section. It is important to note that we changed the architecture of the controller to align with the input controls of the Carla vehicle. We implemented a hierarchical low-level controller consisting of two integrated modules. The first module is a nonlinear model predictive controller, which computes the desired control inputs, such as the steering angle and longitudinal acceleration. The second module uses the Ackermann PID controller implemented in the Carla Python application programming interface, which operates at the level of the Carla vehicle actuators, takes the outputs from the NMPC controller, along with the desired vehicle velocity calculated from the dynamics of the plant model, and generates the necessary commands to ensure the motion of the Carla vehicle within the simulation environment.

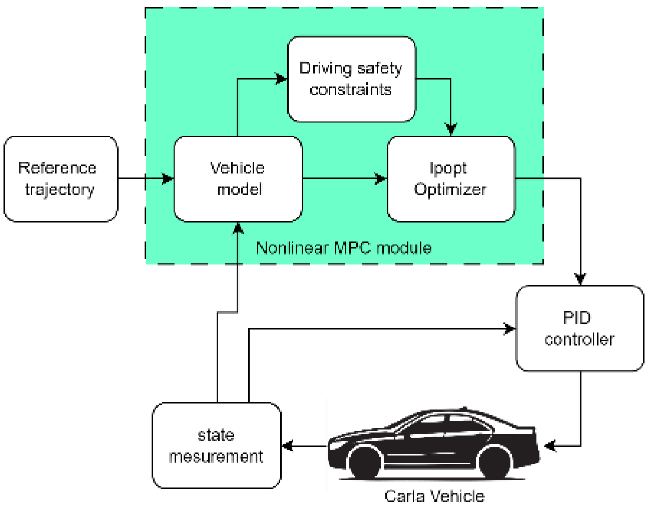


Fig. 2. Hierarchical low-level controller general structure.

3.3. Environment Setup

3.3.1. Reference trajectory extraction

Employed Carla simulator to extract the reference trajectory as presented in Figure 3.

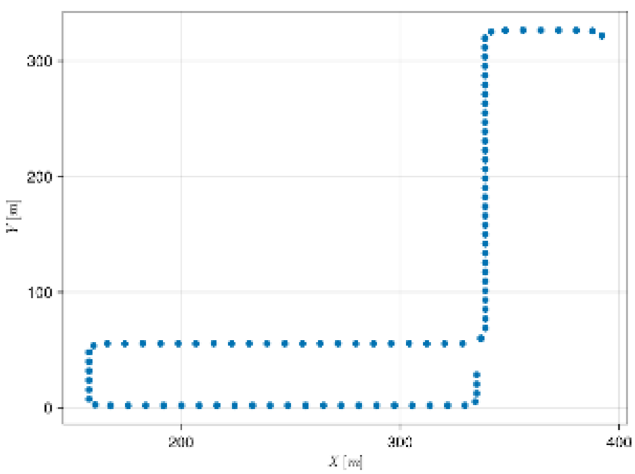


Fig. 3. Reference trajectory waypoints extracted from Carla.

3.3.2 Simulation setup

Utilized the listed parameters in Table 1 to configure the MPC algorithm and the interior point solver.

Table 1. MPC and solver parameters configuration.

MPC Algorithm Parameters	*Value*
Number of shoots/knots (N)	11
MPC Time step	0.05/0.1 seconds
Vehicle baseline length (L)	2.51 m
Maximal steering wheel angle	70 degrees
Maximal acceleration	4.9 m/s^2
Horizon	0.55 s / 1.1 s
Number of collocation points	3
Solver parameters	*Value*
Maximum number of iterations per MPC step	100

4. RESULTS

Throughout the first simulation stage, we ran a total of six simulations to assess the performance of the transcription schemes outlined in this paper. Figures 4 to 15 offer a comprehensive analysis of each transcription approach for time horizons of 0.55 seconds and 1.1 seconds, considering an average cruising speed of 10 m/s.

4.1 Numerical Simulation Results

The simulation results for the three transcription schemes previously introduced are organized into five categories:

- The first category is Figures 4, 8 and 12 addresses the error in the vehicle's position and speed after applying the first control element calculated by the model predictive control (MPC) framework.
- The second category is Figures 5, 9 and 13 evaluates the mean square error of the MPC predictions over the whole considered time horizon.
- The third category is Figures 6, 10 and 14 focuses on the performance of the nonlinear solver used to optimize the trajectory that the vehicle must follow in accordance with the reference trajectory.
- The fourth category is Figures 7, 11 and 15 presents the control elements applied to the vehicle to ensure it remains on the reference trajectory.
- The final category Figure 16 (a, b, c) show how well the tracking trajectory is achieved in the case of MPC horizon $T = 1.1s$ for the three transcriptions schemes.

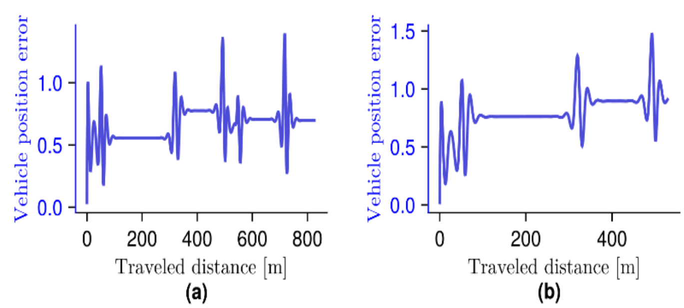
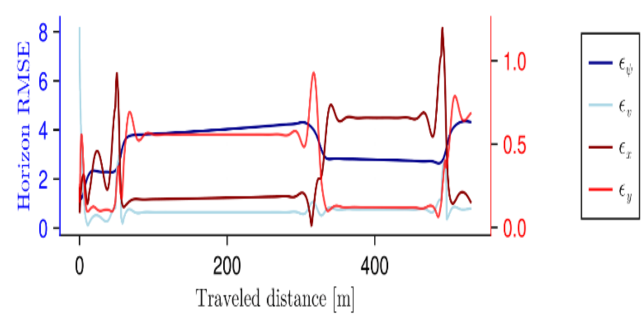
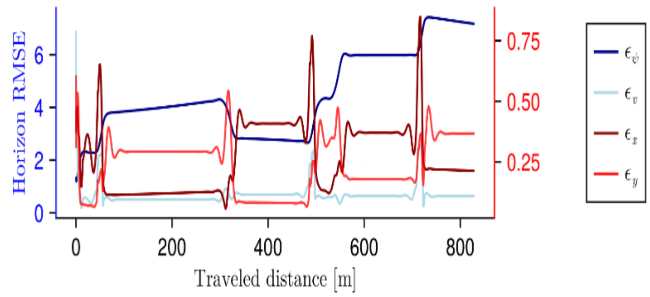


Fig. 4. Position error in the longitudinal direction for DMS transcription scheme: (a) horizon $T = 1.1\ s$, (b) horizon $T = 0.55\ s$.



(a)



(b)

Fig. 5. States vehicle error on the whole horizon $T = 0.55\ s$ for the DMS transcription scheme: : (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.

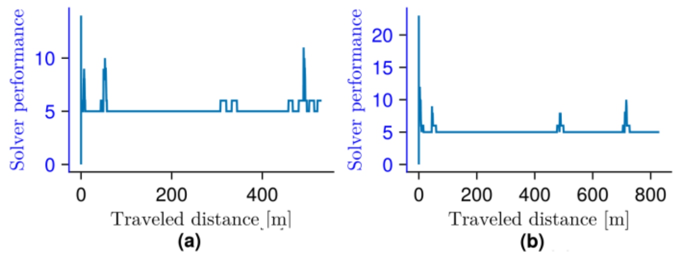
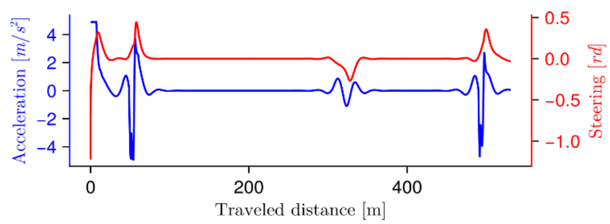
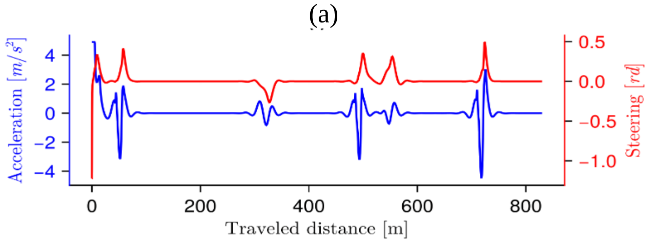


Fig. 6. Interior point solver iterations / nonlinear MPC step for DMS transcription scheme: (a) horizon $T = 1.1\ s$, (b) horizon $T = 0.55\ s$.



(a)



(b)

Fig. 7. Controller performance for DMS transcription scheme: (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.

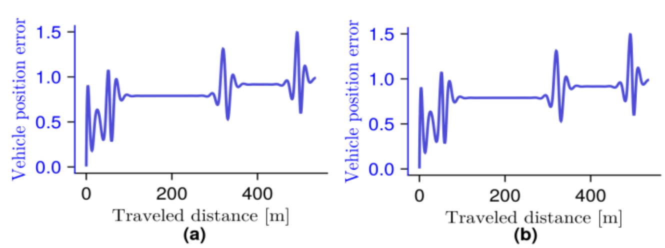
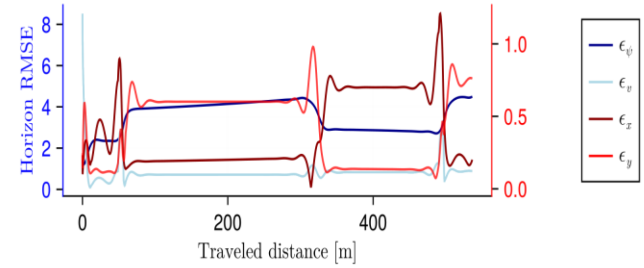
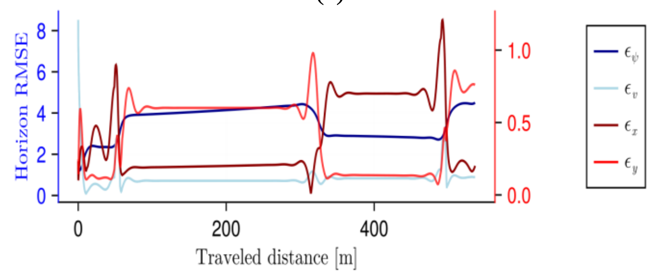


Fig. 8. Position error in the longitudinal direction for DOC transcription scheme: (a) horizon $T = 1.1\ s$, (b) horizon $T = 0.55\ s$.



(a)



(b)

Fig. 9. States vehicle error on the whole horizon $T = 0.55\ s$ for the DOC transcription scheme: (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.

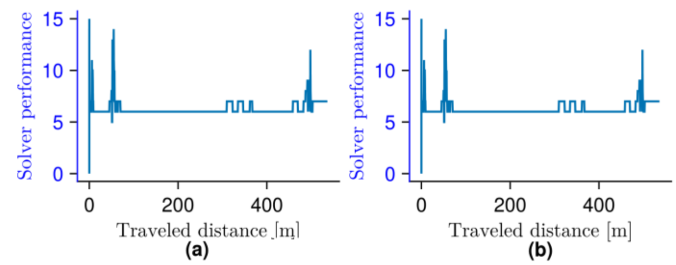
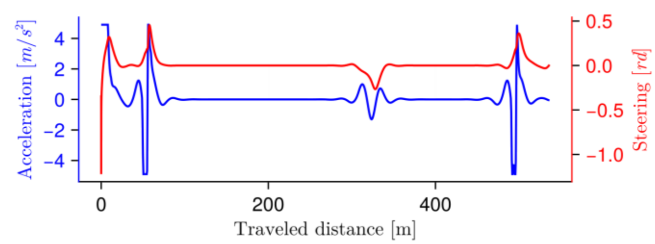
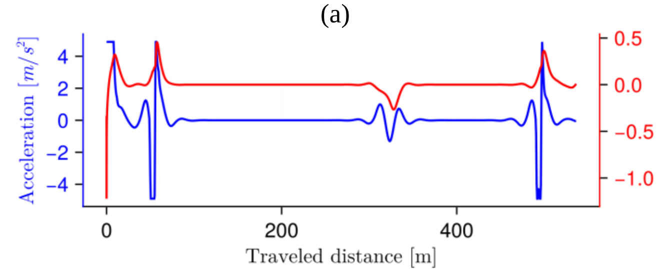


Fig. 10. Interior point solver iterations / nonlinear MPC step for DOC transcription scheme: (a) horizon $T = 1.1\ s$, (b) horizon $T = 0.55\ s$.



(a)



(b)

Fig. 11. Controller performance for DOC transcription scheme: (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.

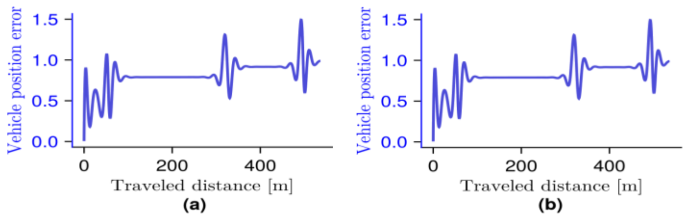
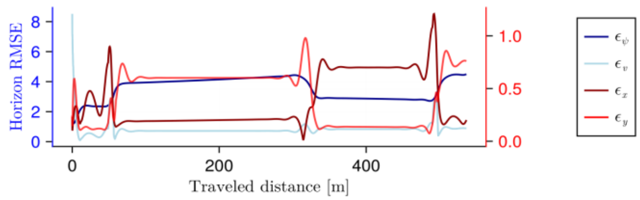
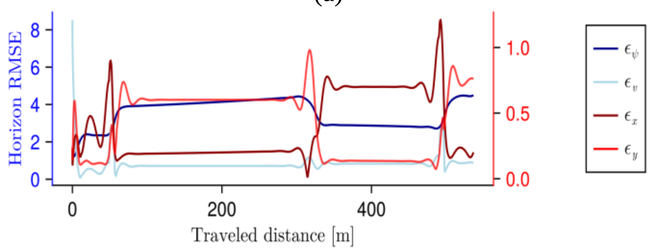


Fig. 12. Position error in the longitudinal direction for IMSDOC transcription scheme: (a) horizon $T = 1.1\ s$, (b) horizon $T = 0.55\ s$.



(a)



(b)

Fig. 13 States vehicle error on the whole horizon $T = 0.55\ s$ for the IMSDOC transcription scheme: (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.

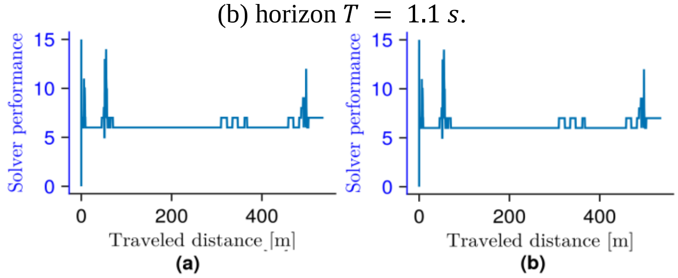
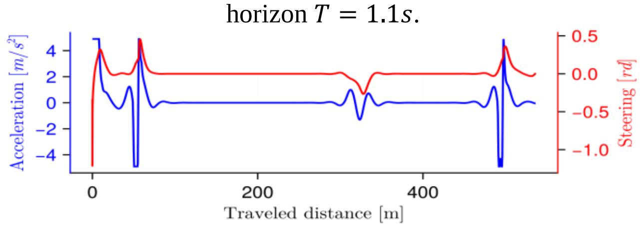
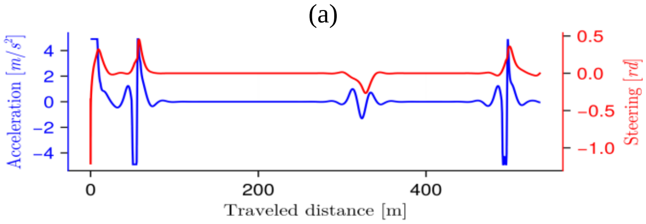


Fig. 14. Interior point solver iterations / nonlinear MPC step for IMSDOC transcription scheme: a horizon $T = 0.55s$., b. horizon $T = 1.1s$.

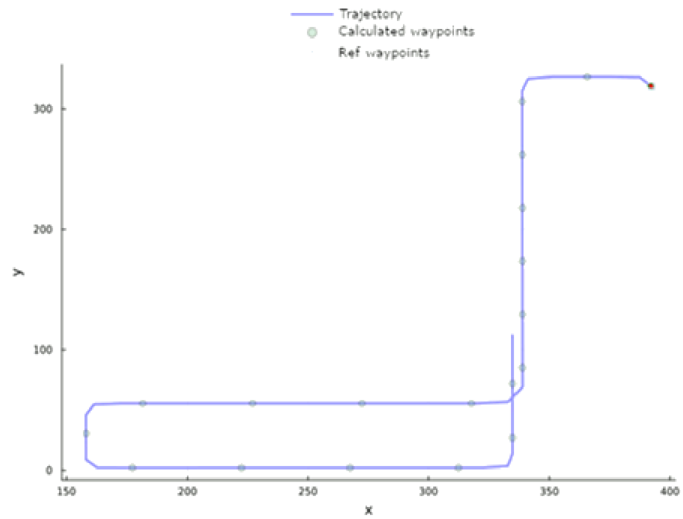


(a)

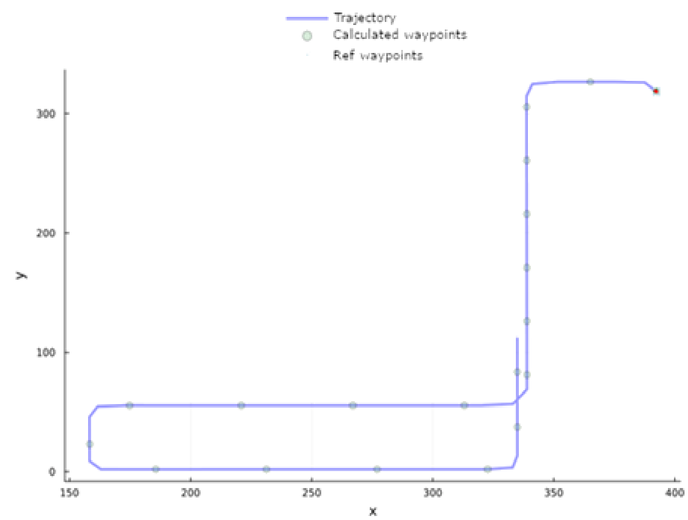


(b)

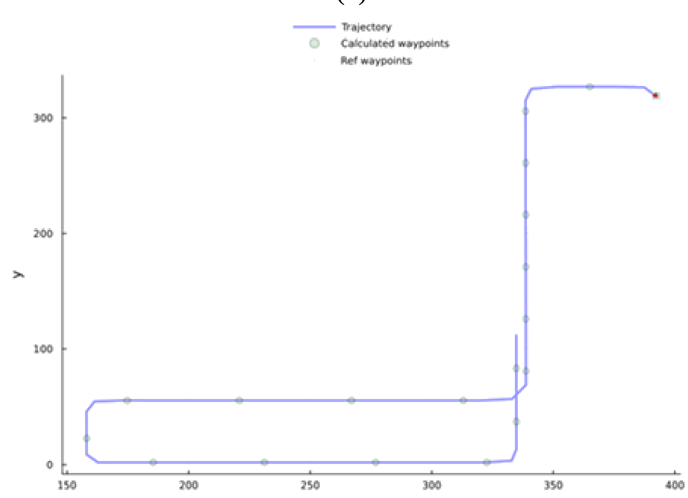
Fig. 15. Controller performance for IMSDOC transcription scheme: (a) horizon $T = 0.55\ s$, (b) horizon $T = 1.1\ s$.



(a)



(b)



(c)

Fig 16. Trajectory tracking for horizon $T = 1.1s$: (a) DMS transcription scheme, (b) DOC transcription scheme, (c) IMSDOC transcription scheme.

4.2. Analysis and Discussion

The numerical simulations conducted for the three transcription schemes mentioned above have successfully achieved their objectives. Each scheme effectively minimizes the discrete optimization problem with an average of 5 solver iterations per MPC step, as shown in Figures 4, 6, 8, 10, 12 and 14. Collocation methods exhibit greater precision in determining the control input, which is evident in Figures 11 and 15. However, these methods are relatively slower than the DMS method, which can be attributed to the differences in problem dimensionality between the two formulations. All three methods perform well in tracking the reference trajectory, as illustrated in Figures 16 (a, b) and (c). The root mean square error calculated across the entire horizon Figures 5, 9 and 13 between the targeted and predicted states for the three schemes is low and consistent, demonstrating the effectiveness of using a uniform distribution of collocation points.

4.3 Results Validation

The presented simulation results conducted in Carla for the same trajectory and simulation setup, focusing on the longest horizon simulation case and using DMS and IMSDOC transcription schemes respectively seen in Figure 17 to 19.

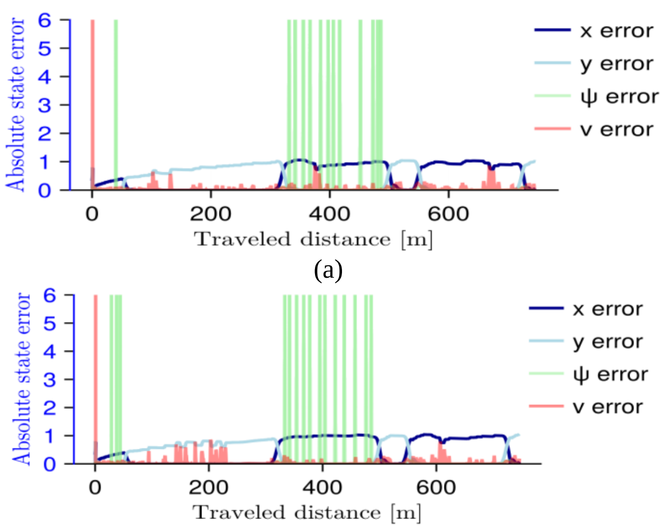


Fig. 17. State error in the Carla simulator for horizon $T = 1.1\ s$: (a) DMS transcription scheme, (b) IMSDOC transcription scheme.

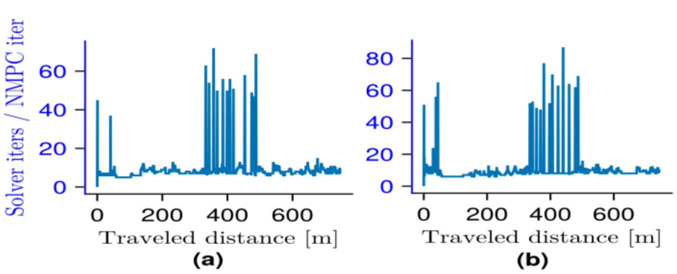


Fig. 18. Performance of the interior point solver in Carla simulator for horizon $T = 1.1s$: (a) DMS transcription, (b) IMSDOC transcription.

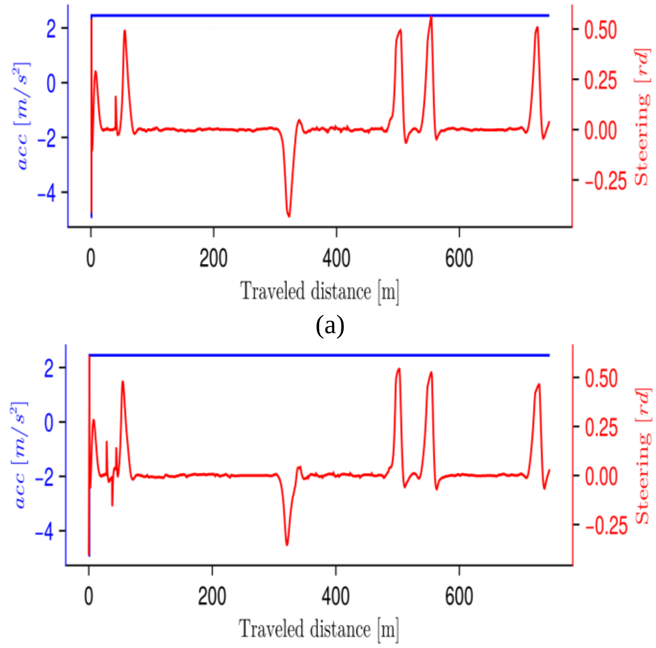


Fig. 19. Controller performance in Carla simulator horizon T=1.1s: (a) DMS transcription, (b) IMSDOC transcription.

4.4 Analysis and Discussion

In Figure 17(b), absolute error in vehicle positioning using the Carla simulator demonstrates consistent performance, comparable to that of the DMS transcription scheme as shown in Figure 17(a). The longitudinal position errors remain below 1 meter, indicating a reliable tracking capability. Additionally, the absolute errors in heading angle and speed are maintained at low levels.

Figure 18(b) shows the performance of the nonlinear solver optimizer for the IMSDOC transcription scheme, highlighting its consistent efficiency as demonstrated in numerical simulations. However, the optimizer has an average overhead of approximately 60 iterations in certain segments of the desired trajectory. A similar observation applies to the DMS transcription scheme, as illustrated in Figure 18(a). This indicates the presence of instabilities that must be addressed and corrected in the kinematic bicycle model used during these simulations.

Furthermore, the acceleration in Figure 19(a) and (b) is kept constant along the vehicle trajectory. This can be explained by the fact that the PID has shifted control over this parameter to the throttle and brake inputs. Consequently, the steering control inputs exhibit remarkable smoothness, with minimal oscillations across the two inspected transcription schemes in the Carla simulator.

These findings strongly support our decision to implement a uniform distribution of collocation points paired with lower-order interpolation polynomials, which proves to be a justified and effective approach to optimizing vehicle control.

4. CONCLUSION

In this paper, we addressed a trajectory tracking problem for a vehicle equipped with a driver assistance system. Our primary objective was to develop a nonlinear model predictive controller. To achieve this, we first selected as the vehicle model the kinematic bicycle model on which the controller would operate.

Next, we elaborated on the trajectory tracking problem, which enabled us to formulate a continuous nonlinear optimization problem. We then explored various transcription methods to generate a discrete version of this problem that could be implemented and solved on a digital computer. We focused on three techniques: the first was direct multiple shooting, the second was direct orthogonal collocation, and the third combined direct orthogonal collocation with implicit multiple shooting for which we employed a uniform collocation point distribution with low-order Lagrange polynomial interpolation for and a small MPC timestep in the latter method.

For each technique, we generated an equivalent discrete optimization problem and subsequently entered the simulation phase, utilizing a solver based on the interior point method. The results indicated that all three methods were well-suited for this problem.

To validate our findings, we also tested our controller within the CARLA simulation environment, which confirmed the efficiency of the MPC controller for this application. Although this work is still ongoing, we plan to implement this type of controller on embedded computers to evaluate their performance further. Additionally, we intend to compare the performance of these techniques with other orthogonal and non-orthogonal collocation methods, including trapezoidal collocation and Hermite-Simpson collocation.

References

[1] K. Kircher and C. Ahlstrom, "Minimum Required Attention: A Human-Centered Approach to Driver Inattention," *Human Factors*, 59(3), pp. 471–484, 2017.

[2] A. J. Khattak, N. Ahmad, B. Wali, and E. Dumbaugh, "A taxonomy of driving errors and violations: Evidence from the naturalistic driving study," *Accid. Anal. Prev*., vol. 151, p. 105873, Mar. 2021

[3] P. Sârbescu and A. Rusu, "Personality predictors of speeding: Anger-Aggression and Impulsive-Sensation Seeking. A systematic review and meta-analysis," *J. Safety Res*., vol. 77, pp. 86–98, Jun. 2021.

[4] I. Kotseruba and J. K. Tsotsos, "Attention for Vision-Based Assistive and Automated Driving: A Review of Algorithms and Datasets," *IEEE Trans. Intell. Transp. Syst.*, 23(11), pp. 19907–19928, 2022.

[5] H. Hsiao, J. Chang, and P. Simeonov, "Preventing Emergency Vehicle Crashes: Status and Challenges of Human Factors Issues," *Human Factors*, 60(7), pp. 1048–1072, 2018.

[6] P. Pathrose, "ADAS and automated driving: a practical approach to verification and validation," *SAE International*, 2022.

[7] D. O. Aihe and A. J. Gonzalez, "Correcting flawed expert knowledge through reinforcement learning," *Expert Syst. Appl.*, 42(17–18), pp. 6457–6471, 2015.

[8] W. Xu, R. Sainct, D. Gruyer, and O. Orfila, "Safe Vehicle Trajectory Planning in an Autonomous Decision Support Framework for Emergency Situations," *Appl. Sci.*, 11(14), p. 6373, 2021.

[9] N. V. Thanh and T. T. M. Linh, "Real-time Trajectory Planning for Autonomous Vehicles in Dynamic Traffic Environments: A Survey of Modern Algorithms and Predictive Techniques," *J. Intell. Connect. Emerg. Technol*., 7(12), pp. 1–25, 2022.

[10] Z. Huang, "Learning-enabled decision-making for autonomous driving: framework and methodology," *Ph.D. dissertation*, Nanyang Technological University, 2023.

[11] N. Dong, S. Chen, Y. Wu, Y. Feng, and X. Liu, "An enhanced motion planning approach by integrating driving heterogeneity and long-term trajectory prediction for automated driving systems: A highway merging case study," *Transportation Research Part C: Emerging Technologies*, vol. 161, p. 104554, Apr. 2024.

[12] M. Margherita, "Development of a hierarchical architecture for real-time autonomous vehicle control," *Ph.D. dissertation*, PhD, Degli Studi di Firenze University, 2021.

[13] L. Chen et al., "Milestones in Autonomous Driving and Intelligent Vehicles Part I: Control, Computing System Design, Communication, HD Map, Testing, and Human Behaviors," *IEEE Trans. Syst. Man Cybern. Syst.*, 53(9), pp. 5831–5847, 2023.

[14] L. Chen et al., "Milestones in Autonomous Driving and Intelligent Vehicles—Part II: Perception and Planning," *IEEE Trans. Syst. Man Cybern. Syst*., 53(10), pp. 6401–6415, 2023.

[15] Q. Wang, Q. Chen, C. Liu, and L. Li, "A Model Predictive Control Algorithm for Path Tracking Based on Multi-point Preview Dynamics and Safety Guaranteed Constraint," *IEEE Trans. Veh. Technol.*, Early Access, pp. 1–15, 2024.

[16] S. Hong and G. Park, "Trajectory Optimization and Robust Tracking Control for Off-Road Autonomous Vehicle," *IEEE Access*, vol. 12, pp. 82205–82219, 2024.

[17] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Review and performance evaluation of path tracking controllers of autonomous vehicles," *IET Intell. Transp. Syst.*, 15(5), pp. 646–670, 2021.

[18] A. Ortiz, S. Garcia-Nieto, and R. Simarro, "Comparative Study of Optimal Multivariable LQR and MPC Controllers for Unmanned Combat Air Systems in Trajectory Tracking," *Electronics*, 10(3), p. 331, 2021.

[19] F. Liu, P. Qin, Z. Guo, Y. Shang, and Z. Li, "Research on Robust Model Predictive Control Based on Neural Network Optimization for Trajectory Tracking of High-Speed Autonomous Vehicles," *J. Transp. Res. Board*, Early access, Sep. 2024.

[20] M. H. Badiei, S. M. Dashtaki, N. Vafamand, and Md. J. Piran, "Uncertainty-Aware Path Planning with Multiple Constraints for Autonomous Vehicles Based on Nonlinear Reduced-Order Model Predictive Control," *IEEE Trans. Consum. Electron.*, Early access, pp. 1–1, 2024.

[21] R. Bordalba, T. Schoels, L. Ros, J. M. Porta, and M. Diehl, "Direct Collocation Methods for Trajectory Optimization in

Constrained Robotic Systems," *IEEE Trans. Robot.*, 39(1), pp. 183–202, 2023.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," *Proceedings of the 1st Annual Conference on Robot Learning*, PMLR vol. 78, pp. 1-16, 2017

[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, 106(1), pp. 25–57, 2006.

[24] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A Fast Dynamic Language for Technical Computing," arXiv:1209.5145[cs], 2012.

[25] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A Modeling Language for Mathematical Optimization," *SIAM Rev.*, 59(2), pp. 295–320, 2017.

[26] S. Christ, D. Schwabeneder, C. Rackauckas, M. K. Borregaard, and T. Breloff, "Plots.jl -- a user extendable plotting API for the Julia programming language," arXiv:2204.08775, Jun. 17, 2022,

[27] S. Danisch and J. Krumbiegel, "Makie.jl: Flexible high-performance data visualization for Julia," *JOSS*, 6(65), p. 3349, 2021.

[28] Steven G. Johnson, "PyCall.jl: A package to call Python functions from Julia," *GitHub repository*, v1.95.1, 2025.

Biographies

Hicham Belkebir was born on April 9, 1973, in Meknes, Morocco, from a young age, he showed a keen interest in science and technology. In 1979, he began his schooling and obtained his baccalaureate in 1990 with honours in industrial chemistry from Lycée Moulay Youssef in Tangier. Pursuing his passion for science, he enrolled at the Faculty of Sciences in Meknes, where he earned a bachelor's degree in physical sciences, specializing in electronics, in 1996. His professional career began in 1998 when he was appointed as a laboratory preparer at ENSAM in Meknes. In 2003, he made a significant leap in his career by becoming an application engineer. Always eager to learn, Hicham enrolled in a postgraduate program in 2005 and obtained an advanced studies diploma in 2007 in the field of optical information processing. In 2008, he embarked on a national doctorate and defended his thesis in 2015, focusing on the contribution of planar photonic crystals to improving the performance of MZI-type optical modulators in SOI technology. In 2016, he left ENSAM to join the National School of Applied Sciences in Fez, where he continues to work to this day. Passionate about new technologies, programming, artificial intelligence, cinema, and travel, he remains dedicated to his fields of interest, contributing to the advancement of science and technology.

Taoufik Belkebir was born on January 02, 1988, in Meknes, Morocco. From an early age, he demonstrated a strong inclination toward technology and problem-solving. In 1995, Taoufik began his schooling journey and excelled academically, earning his baccalaureate in 2006 with honours in mathematical sciences from Lycée Imam El Ghazali in Meknes. His passion for innovation led him to pursue engineering studies at the National School of Applied Sciences (ENSA) in Oujda, where he graduated in 2012 with an engineering degree in electronics and industrial computing.

Taoufik's professional career began in 2012, with a role at Zodiac Aerospace, where he developed expertise in model-based development and verification for aeronautic safety-critical embedded systems. Over the years, he advanced to leadership positions, including his current role as R&D Manager at ALTEN Morocco, where he oversees multidisciplinary projects in aerospace and embedded software. Committed to lifelong learning, he obtained a master's degree in project management in 2016 and earned certifications in Safe Agile and ISTQB testing standards.

In 2021, he embarked on a PhD journey at FST Sidi Mohamed Ben Abdellah University, focusing on optimal control and reinforcement learning in autonomous vehicles and robotics. His research emphasizes intelligent planning and robust control strategies, aiming to contribute to the advancement of autonomous systems. Outside his professional and academic pursuits, he is passionate about mentoring, teaching project management, and exploring the intersections of technology, artificial intelligence, and innovation.