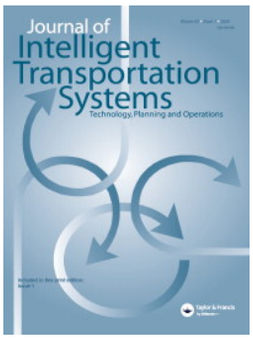


Taylor & Francis
Taylor & Francis Group

Journal of Intelligent Transportation Systems
Technology, Planning, and Operations

ISSN: 1547-2450 (Print) 1547-2442 (Online) Journal homepage: www.tandfonline.com/journals/gits20

Design of a mixed proportional integral - nonlinear model predictive control strategy for automated vehicles: application to mechanically connected convoys

Michael Franci, Luca Pugi, Michelangelo-Santo Gulino & Adriano Alessandrini

To cite this article: Michael Franci, Luca Pugi, Michelangelo-Santo Gulino & Adriano Alessandrini (18 May 2026): Design of a mixed proportional integral - nonlinear model predictive control strategy for automated vehicles: application to mechanically connected convoys, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2026.2669494

To link to this article: https://doi.org/10.1080/15472450.2026.2669494

Published online: 18 May 2026.

JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS
https://doi.org/10.1080/15472450.2026.2669494

Design of a mixed proportional integral - nonlinear model predictive control strategy for automated vehicles: application to mechanically connected convoys

Michael Franci, Luca Pugi, Michelangelo-Santo Gulino, and Adriano Alessandrini

Department of Industrial Engineering, University of Florence, School of Engineering (DIEF), Florence, Italy

ABSTRACT

Recent advances in sustainable mobility have accelerated autonomous driving technologies, particularly for public transport convoys of mechanically connected vehicles, improving the mobility and safety of urban mobility. This work develops a control architecture for such convoys, focusing on trajectory tracking and vehicle coordination. Conducted within a UniFi research project on automated electric minibuses, the study evaluates both mechanical coupling and conventional platooning strategies to optimize the reliability of urban transport. A Nonlinear Model Predictive Controller (NMPC) is designed for vehicle lateral and longitudinal control, ensuring precise tracking while considering actuator constraints and vehicle dynamics. The approach is first applied to a single vehicle and then extended to a two-vehicle convoy, where a Proportional-Integral (PI) controller manages the distance between vehicles. The follower replicates the leader's path while avoiding corner cutting and maintaining a predefined gap. The simulation results demonstrate excellent performance under ideal conditions, achieving precise trajectory tracking with a distance error limited to $\pm 0.05\ m$. However, Model-In-the-Loop (MIL) tests reveal a larger deviation (up to 0.80 *m*) due to trajectory characteristics and dynamic constraints. The MATLAB/Simulink ® implementation, validated with GNSS-acquired (Global Navigation Satellite System), confirms the viability of the proposed strategy, paving the way for further optimisations and experimental validations.

ARTICLE HISTORY
Received 31 March 2025
Revised 17 February 2026
Accepted 30 April 2026

KEYWORDS
distance; lateral control; longitudinal control; trajectory tracking; vehicle dynamics

1. Introduction

In recent years, autonomous vehicle research has made significant progress, leading to a substantial evolution of this technology. Early experiments date back to the 1980s, but it is particularly in recent decades that this field has experienced significant development. The ever-increasing interest in this area of study is primarily justified by the numerous advantages it offers: active devices on board and, more prominently, autonomous vehicles promise improved driving conditions and increased road safety by reducing accidents caused by human error (Chelbi et al., 2022; Gulino et al., 2024; Lyu et al., 2021; Vangi et al., 2019; Yao et al., 2021), as well as improved driving comfort (Mohtavipour & Mollajafari, 2021; Tapani, 2012), reduced energy consumption in transportation, and consequently lower costs (Shladover, 2018).

The application of autonomous driving has also been extended to vehicle convoys in the context of the so-called "convoying"[1], where a fleet of vehicles travels in alignment, following the same trajectory. In this case as well, autonomous convoying offers exciting opportunities, improving logistic efficiency and road safety while simultaneously reducing fuel consumption and emissions (Nahavandi et al., 2022). The connection among automated vehicles also allows their circulation in any mixed traffic environment (Arvin et al., 2021) without the need for specific permissions: if the leading vehicle is driven by humans, the mechanical connection serves as a safety device that enables the following automated vehicles to replicate its trajectory even in the event of an electrical failure of the convoy.

This paper aims to propose a controller architecture for a convoy of automated vehicles, specifically for urban public transport applications. It focuses on

CONTACT Michael Franci michael.franci@unifi.it Department of Industrial Engineering, University of Florence, School of Engineering (DICEA), Florence, Italy.

electric buses of the type "Tecnobus Gulliver" in Figure 1, currently used for experimentation by the University of Florence (Adriano et al., 2021). The objective is to allow convoys of mechanically connected electric vehicles to move in an automated manner and follow a predefined route, where the mechanical connection also exploits the possibility of an electrical connection to perform Vehicle-to-Vehicle (V2V) power sharing between vehicles described in Alessandrini et al. (2023) and Alessandrini et al. (2024), also called shared dynamic charging.

The development of this type of technology fits into a wider framework for the advancement and optimization of transportation systems, which has attracted increasing interest in recent years. These objectives are further enabled by the simultaneous advancements in energy management, the development of control systems, and increasingly sophisticated communication technologies. The primary objective of this work is to develop a control system for a convoy of two or more automated vehicles using a mixed PI-NMPC approach, where the two-vehicle setup is a necessary baseline before scaling to N-vehicle convoys due to the mechanical stress propagation in rigid couplings. Initially, the vehicle dynamics model was derived, which can be approximated by the simplified bicycle model. This model was implemented in the MATLAB/Simulink environment to facilitate the subsequent development of the controller. To evaluate the performance of the control architecture, it was validated on different test trajectories with specific characteristics and then extended to a trajectory registered *via* GNSS to evaluate its performance and robustness in a Model-In-the-Loop (MIL validation).

2. Related works

Trajectory tracking allows a vehicle to follow a predefined reference trajectory, which consists of a sequence of waypoints and an associated speed profile. This technique must satisfy not only spatial (position and orientation) constraints but also the temporal constraints established by the route Li et al. (2021) and Dixit et al. (2018). The trajectory can be generated offline or determined online *via* a planning module. Numerous control methodologies have been developed for this application, with many solutions existing for both single-vehicle autonomous driving and multi-vehicle convoys, particularly in the context of platooning Nahavandi et al. (2022). Convoy control strategies vary depending on the objectives and the sensing/communication methods. Generally, each vehicle follows the path of the one in front while maintaining a safe distance.

Several approaches in the literature address the challenge of convoy control, especially the "cutting-corners" problem that is common in trailers. For example, Vasconcelos Filho et al. (2020) proposes a PI Look-Ahead Controller based on Vehicle-to-Vehicle (V2V) communication, which implements separate lateral and longitudinal PI controllers and includes an orientation adjustment mechanism. Similarly, Bayuwindra et al. (2020) presents an Extended Look-Ahead controller with an orientation error observer, using an extended reference point to mitigate corner-cutting. Other kinematic approaches, such as in Petrov (2008), use adaptive control with virtual reference points to manage the follower's position relative to the leader. These strategies often rely on V2V communication to transfer the necessary tracking information. Another common approach, seen in Pauca et al. (2020), combines a trajectory planner with a tracker, where each follower receives a target position and solves a local optimization problem, while a Cooperative Adaptive Cruise Control (CACC) mechanism manages the distance. Recent studies in 2024 and 2025 have further expanded this domain by exploring "virtual track" technologies, which aim to mimic rail-like behavior without physical infrastructure. For instance, Hou and Giannopoulos (2024) proposed a virtual track model that converts autonomous vehicle movement into simulated train dynamics to improve management in mixed traffic.

In the field of predictive control, controller designs vary between decoupled and combined systems for longitudinal and lateral dynamics. In Kianfar et al. (2014), the platooning application is addressed by jointly managing both aspects. A headway controller using frequency-domain methods ensures string stability, while a Model Predictive Control (MPC) controller handles lateral lane-keeping. Conversely, Gao (2014) proposes a hierarchical MPC framework. This structure uses a high-level planner with a simplified model to compute obstacle-avoiding trajectories and a low-level MPC with a high-fidelity model to track them, allowing for longer prediction horizons and faster sampling times. This hierarchical approach remains highly relevant in the recent literature on complex vehicle topologies. Notably, Feng et al. (2025) developed a layered coordinated control strategy for articulated vehicles, employing an upper-layer MPC for trajectory planning and a lower-layer control for steering. Furthermore, Huang et al. (2025) demonstrated a multi-objective MPC for electric-driven buses that dynamically allocates stability weights.

Figure 1. Tecnobus Gulliver Minibus employed for the scopes of the research.

Beyond traditional predictive control, recent advancements have increasingly explored data-driven methods to manage complex non-linear traffic dynamics. For instance, a study by Marin et al. (2025) introduces a Deep Reinforcement Learning (DRL) framework based on the Deep Deterministic Policy Gradient (DDPG) algorithm for autonomous car-following. Their approach utilizes a novel reward function to simultaneously optimize energy efficiency, safety, and passenger comfort across various vehicle powertrains, demonstrating high adaptability compared to traditional MPC. In a related effort to stabilize mixed traffic flows, Dayi et al. (2022) proposed a trajectory-based coordinated speed control strategy that leverages connected vehicle and roadside sensor data to predict the trajectories of downstream human-driven vehicles. This prediction allows upstream automated vehicles to proactively plan their speed distributions and smoothly merge into traffic, effectively suppressing backward-propagating traffic oscillations. However, while DRL models and electronic coordination excel in continuous-space adaptivity, the rigorous constraint handling required for exact path replication makes NMPC highly suitable for physical linkages.

While the reviewed work has significantly advanced the control of single automated vehicles or *electronically* platooned fleets (which often use CACC based on real-time leader-following), a specific gap remains in the control of *mechanically connected* convoys. This application presents a fundamentally different challenge: to prevent "cutting-corners," the follower vehicle must not track the leader's current position but must instead precisely replicate the path that the leader has previously traversed. To address this gap, this paper contributes a novel control architecture. The core of our strategy is the implementation of a FIFO (First-In, First-Out) buffer that records the leader's trajectory, including position, orientation, and velocity. This stored path is then supplied to the follower as a delayed reference signal. The delay in this buffer is dynamically managed to correspond to the desired convoy distance (i.e., the length of the physical connection). This path-replication mechanism is then governed by our mixed PI-NMPC control strategy, where the NMPC handles the high-level task of tracking this delayed path, and the PI controller manages the low-level distance adjustments. This specific architecture, which uses a FIFO buffer to enable precise path replication for mechanically linked vehicles, has not been addressed in the reviewed literature.

3. Materials and methods

3.1. Case study

The case study involves vehicles equipped with a mechanical coupling system[2], designed primarily as a safety feature. This system ensures continued operation even

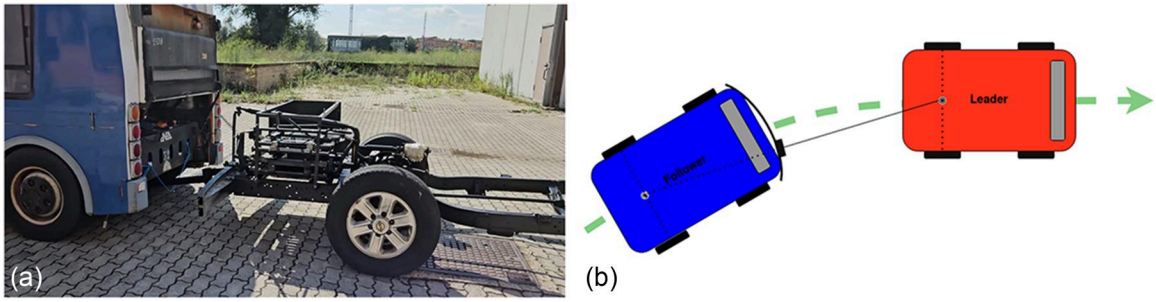


Figure 2. Convoy prototype in which the follower integrates a patented steering delaying mechanism that enables it to replicate the steering of the leader without the use of electric/electronic components.

Figure 3. Automated golf car-type vehicle.

in the event of a controller malfunction or power failure; since a human operator is always present in the leading vehicle, the following automated vehicles are mutually tied and can circulate in any urban environment without the special permissions that would otherwise be required. The convoy takes the form of a tram that moves independently of a rail-based infrastructure. The design incorporates a sliding cursor fixed to the follower vehicle, which maintains a consistent distance between vehicles while simultaneously transmitting commands. These commands are appropriately delayed to ensure that the follower accurately replicates the leader's trajectory. To better explain the regulating concepts of the idea, Figure 2a presents a prototype of the follower that has no electrical components, while Figure 2b highlights the physical principles behind the mechanism. Since the research focusses on electric buses, but the initial experimental tests will be conducted on a golf-cart-type vehicle (Figure 3), equipped for autonomous operation due to reasons of availability and management, the simulations were performed using data from the latter. This approach allows testing the control architecture on a more accessible system before implementing it on a bus.

3.2. Vehicle model

The control strategy employs a dynamic bicycle model, which offers a more accurate representation of vehicle behavior at higher speeds and complex dynamics

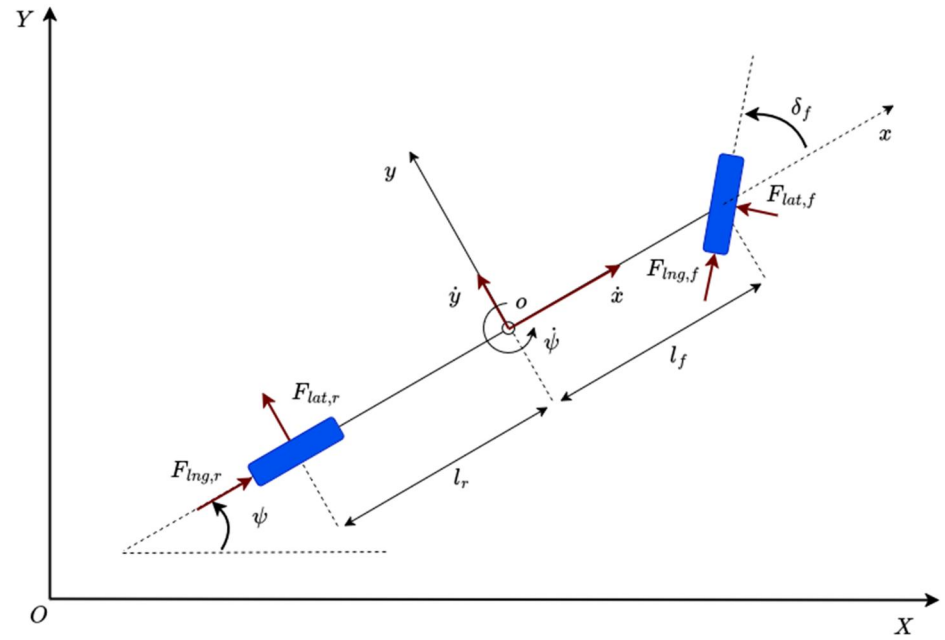


Figure 4. Dynamic bicycle model.

Table 1. Symbols and definitions of the dynamic bicycle model.

Symbol	Definition
m	Vehicle mass
δ_f	Front wheel steering angle
l_f, l_r	Center of mass distance from front/rear axes
$\dot{x}, \dot{y}$	Vehicle longitudinal and lateral speed in *xoy* reference
$\psi, \dot{\psi}$	Yaw angle and Yaw rate
$F_{lat,i}, F_{lng,i}$	Lateral and Longitudinal forces on $i \in \{f, r\}$ tyres

compared to kinematic models Polack et al. (2017, 2018). As shown in Figure 4, the vehicle is approximated as a rigid body with three degrees of freedom: longitudinal motion x, lateral motion y, and yaw rotation ψ. We assume that the vehicle mass is equally distributed, and effects such as aerodynamic drag, suspension dynamics, and road gradient are negligible.

The symbols used in the model are defined in Table 1.

By applying Newton's second law of motion to the vehicle frame *xoy*, we obtain the following dynamic equations governing the longitudinal, lateral, and rotational motions Rajamani (2011) and Wang et al. (2019):

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{x,f} + 2F_{lng,r} \tag{1}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{y,f} + 2F_{lat,r} \tag{2}$$

$$I_z\ddot{\psi} = 2l_fF_{y,f} - 2l_rF_{lat,r} \tag{3}$$

where $F_{x,f}$ and $F_{y,f}$ represent the projections of the front tyre forces onto the vehicle coordinate system.

The lateral forces are modeled using a linearized tyre model Pacejka and Bakker (1992), assuming the lateral force is proportional to the tyre side-slip angle α. Given the operational speeds ($v_{\max} \approx 9\ m/s$) and the absence of aggressive maneuvers, we assume small steering angles ($\delta_f \ll 1$) and neglect longitudinal slip rates ($\gamma \approx 0$). Consequently, the lateral forces and slip angles are defined as:

$$\begin{cases} F_{lat,f} = C_{lat,f}\alpha_f \approx C_{lat,f}\left(\dfrac{\dot{y} + l_f\dot{\psi}}{\dot{x}} - \delta_f\right) \\ F_{lat,r} = C_{lat,r}\alpha_r \approx C_{lat,r}\left(\dfrac{l_r\dot{\psi} - \dot{y}}{\dot{x}}\right) \end{cases} \tag{4}$$

where $C_{lat,f}$ and $C_{lat,r}$ are the cornering stiffness coefficients. Substituting Equation (4) into Equations (1)–(3) yields the final nonlinear state-space model used for the controller design. To improve the realism of the longitudinal dynamics and account for actuator delays (e.g., vehicle inertia and wheel dynamics), we model the longitudinal acceleration using a first-order jerk equation:

$$\ddot{x} = \frac{1}{\tau}(-\ddot{x} + a_x) \tag{5}$$

where a_x is the commanded acceleration and τ is the time constant representing the system response delay [cite: 204-210]. This acts as a low-pass filter,

preventing unrealistic instantaneous acceleration changes. Consequently, by combining the lateral dynamics derived in Equations (1)–(3) with the longitudinal jerk model, we define the state vector $\chi = [\dot{y}, \dot{\psi}, \dot{x}, \ddot{x}]^T$ and control input $u = [a_x, \delta_f]^T$. The resulting continuous-time state-space model is:

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \begin{bmatrix} \frac{2(C_{lat,f}-C_{lat,r})}{m\dot{x}} & -\dot{x} + \frac{2(C_{lat,f}l_f - C_{lat,r}l_r)}{m\dot{x}} & 0 & 0 \\ \frac{2(C_{lat,f}l_f - C_{lat,r}l_r)}{I_z\dot{x}} & \frac{2(C_{lat,f}l_f^2 - C_{lat,r}l_r^2)}{I_z\dot{x}} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \chi + \begin{bmatrix} 0 & -\frac{2C_{lat,f}}{m} \\ 0 & -\frac{2l_f C_{lat,f}}{I_z} \\ 0 & 0 \\ \frac{1}{\tau} & 0 \end{bmatrix} u \tag{6}$$

3.3. Tracking errors model

The proposed control architecture combines lateral and longitudinal dynamics to create an NMPC controller for the automated driving of a vehicle, which minimizes the error with respect to the trajectory to be followed (defined a *priori*) by regulating the longitudinal acceleration and the steering angle. The objective is therefore to minimize the lateral error e_{lat} and the orientation error e_{yaw}, while maintaining the reference speed defined a priori for each waypoint of the trajectory. A vehicle-based model was used to create this control system based on the lateral and longitudinal dynamics equations defined in the previous paragraph, in terms of lateral e_{lat}, orientation e_{yaw}, and velocity e_{vel} errors with respect to the reference trajectory. The lateral error e_{lat} is defined as the lateral deviation of the center of mass of the vehicle with respect to the reference path, while the orientation error e_{yaw} is given by the difference between the yaw angle of the vehicle and the reference angle of the trajectory. The speed error e_{vel} is obtained as the difference between the current speed of the vehicle and the desired speed supplied to the control system. In this case, the reference speed will not be constant but will vary along the trajectory. The equations that define the errors are as follows:

$$e_{lat} = -(x - x_{ref})\sin(\alpha) + (y - y_{ref})\cos(\alpha) \tag{7}$$

$$e_{yaw} = \psi - \psi_{ref} \tag{8}$$

$$e_{vel} = \dot{x}_{ref} - \dot{x} \tag{9}$$

where α is the angle between the current position of the vehicle and the reference position of the trajectory; in particular, (x_i, y_i) represents the current position of the rear axle of the vehicle and (x_{ref}, y_{ref}) denotes the position of the reference waypoint. α is hence defined as:

$$\alpha = tan^{-1}\frac{y_i - y_{ref}}{x_i - x_{ref}} - \psi \tag{10}$$

Therefore, by deriving the previous equations over time, the following equations representing the errors over time are obtained:

$$\dot{e}_{lat} = \dot{x}e_{yaw} + \dot{y} \tag{11}$$

$$\dot{e}_{yaw} = \dot{\psi} - \dot{\psi}_{ref} \tag{12}$$

$$\dot{e}_{vel} = \ddot{x}_{ref} - \ddot{x} \tag{13}$$

where the derivative of ψ_{ref} is obtained from the product of longitudinal velocity V_x and trajectory curvature ρ as follows:

$$\dot{\psi}_{ref} = \frac{\dot{x}}{R} = \dot{x}\rho \tag{14}$$

where R is the radius of curvature of the road. In particular, ρ can be calculated using the following expression:

$$\rho = \frac{|x'y'' - x''y'|}{\left[x'^2 + y'^2\right]^{\frac{3}{2}}} \tag{15}$$

where x', x'', y', y'' are, respectively, the first and second derivatives of the x and y coordinates of the trajectory. At this point, Equations (7)–(9) are included in the previous state model, Equation (6), resulting in the following augmented state in the vehicle frame:

$$\chi_{aug} = \begin{bmatrix} \dot{y} & \dot{\psi} & \dot{x} & \ddot{x} & e_{lat} & e_{yaw} & e_{vel} \end{bmatrix}^T \tag{16}$$

obtaining the following augmented state-space model:

$$\dot{\chi}_{aug} = \begin{bmatrix} \ddot{y} \\ \dot{\psi} \\ \ddot{x} \\ \dddot{x} \\ \dot{e}_{lat} \\ \dot{e}_{yaw} \\ \dot{e}_{vel} \end{bmatrix} = \begin{bmatrix} \frac{2}{m}\frac{C_{lat,f}-C_{lat,r}}{\dot{x}} & -\dot{x}+\frac{2}{m}\frac{C_{lat,f}l_f+C_{lat,r}l_r}{\dot{x}} & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{I_z}\frac{C_{lat,f}l_f+C_{lat,r}l_r}{\dot{x}} & \frac{2}{I_z}\frac{C_{lat,f}l_f^2-C_{lat,r}l_r^2}{\dot{x}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & e_{yaw} & -\frac{1}{\tau} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\psi} \\ \dot{x} \\ \ddot{x} \\ e_{lat} \\ e_{yaw} \\ e_{vel} \end{bmatrix} + \begin{bmatrix} 0 & -\frac{2}{m}C_{lat,f} \\ 0 & -\frac{2}{I_z}C_{lat,f}l_f \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_x \\ \delta_f \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\dot{x} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \rho \\ \ddot{x}_{ref} \end{bmatrix} \tag{17}$$

4. Control architecture

4.1. Model Predictive control (MPC)

Model Predictive Control (MPC) is an advanced optimal control technique, particularly suitable for managing multiple-input and multiple-output (MIMO) systems. It is based on a mathematical model of the system, in our case on the dynamic bicycle model of the vehicle, and involves the optimization of a cost function with respect to a finite prediction horizon to calculate the optimal control actions for the plant being considered. At each iteration, the controller acquires the system state variables and determines the sequence of control actions that minimize the cost function along the prediction horizon, solving a constrained optimization problem. Only the first calculated control action is then applied to the system, after which the process is started to repeat iteratively. An interesting aspect of MPC is the possibility of including constraints and limitations on the inputs, states, and outputs of the system to be controlled, facilitating their management compared to other control techniques. Furthermore, the resulting control action also considers future inputs to the system, depending on the chosen prediction horizon N_p. This is made possible by the predictive approach integrated into the controller, which helps to optimize the behavior of the system. However, compared to other control techniques already known, MPC is more complex and computationally more expensive, since it must solve an optimization problem in real time at each iteration. Figure 5 represents the classic structure of an MPC control system applied to a controlled system. At each iteration, the controller receives input measurements of the outputs and disturbances acting on the system, using them internally as part of the prediction model to estimate the state of the system and calculate the optimal control actions over the defined prediction horizon.

4.2. Nonlinear model Predictive control (NMPC)

In this work, Nonlinear Model Predictive Control (NMPC) is implemented due to the inherently nonlinear nature of vehicle dynamics, as described in Section 3.2. Unlike standard MPC, NMPC is specifically designed to handle nonlinear models, making it well suited to optimizing vehicle control by predicting its future evolution. This predictive capability is crucial in autonomous driving, where anticipating

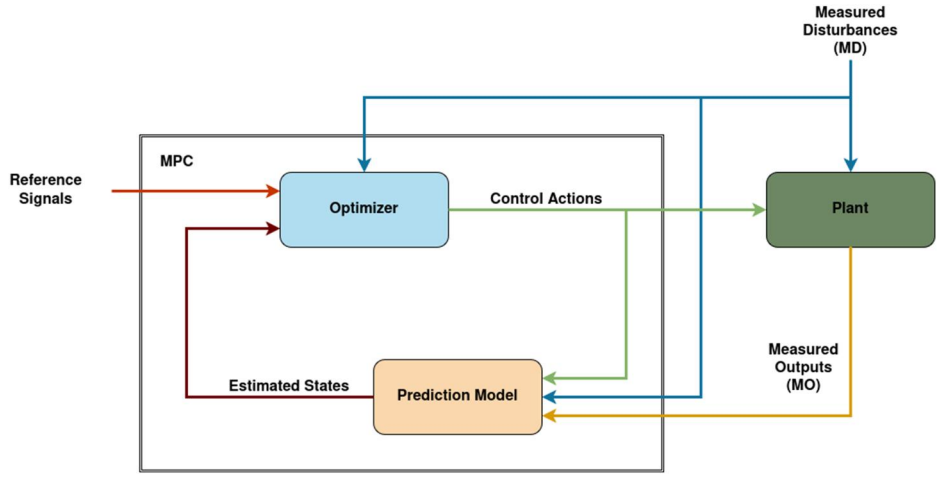


Figure 5. MPC structure.

complex trajectory characteristics is necessary to ensure accurate tracking while simultaneously respecting physical and operational constraints. These constraints are on acceleration, deceleration, and steering angle, which must be enforced to maintain safe and feasible driving conditions. Furthermore, NMPC allows for the simultaneous optimization of multiple objectives, such as minimizing both lateral and longitudinal tracking errors while ensuring that the vehicle maintains its desired orientation and speed. As mentioned above, predictive control operates by solving an optimization problem at each time step, minimizing a cost function that includes terms related to tracking errors and the smoothness of the control actions. However, unlike linear MPC, NMPC relies on a more complex optimization process that directly incorporates the nonlinear vehicle model, ensuring higher accuracy in trajectory tracking. Despite its higher computational complexity, the NMPC provides a more accurate and adaptive control strategy, which makes it particularly effective for autonomous driving applications. As described in the previous section, the seven-state space variable vector can be considered as:

$$\chi_{aug} = \begin{bmatrix} \dot{y} & \dot{\psi} & \dot{x} & \ddot{x} & e_{lat} & e_{yaw} & e_{vel} \end{bmatrix}^T \tag{18}$$

by choosing $u_{mv} = \begin{bmatrix} a_x & \delta_f \end{bmatrix}^T$ as the vector of the manipulated variables (that is, the control variables) and $u_{dist} = \begin{bmatrix} \rho & \ddot{x}_{ref} \end{bmatrix}^T$ as the vector of the measured disturbances. The state equations are based on previous dynamic Equations (2) and (3), the jerk Equation (5), and error Equations (7)–(9), which are obtained as follows:

$$\begin{cases} \dot{\chi}_{aug}(t) = A_t \chi_{aug}(t) + B_t u_{mv}(t) + D_t u_{dist}(t) \\ Y(t) = C_t \chi_{aug}(t) \end{cases} \tag{19}$$

where:

$$A_t = \begin{bmatrix} \frac{2}{m}\frac{C_{lat,f}-C_{lat,r}}{\dot{x}} & -\dot{x} + \frac{2}{m}\frac{C_{lat,f}l_f + C_{lat,r}l_r}{\dot{x}} & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{I_z}\frac{C_{lat,f}l_f + C_{lat,r}l_r}{\dot{x}} & \frac{2}{I_z}\frac{C_{lat,f}l_f^2 - C_{lat,r}l_r^2}{\dot{x}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau} & 0 & 0 & 0 \\ 1 & 0 & e_{yaw} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

$$B_t = \begin{bmatrix} 0 & \frac{2}{m}C_{lat,f} \\ 0 & \frac{2}{I_z}C_{lat,f}l_f \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{21}$$

$$D_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\dot{x} & 0 \\ 0 & 1 \end{bmatrix} \tag{22}$$

$$Y(t) = \begin{bmatrix} e_{lat} \\ e_{yaw} \\ e_{vel} \end{bmatrix} \quad C_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{23}$$

Therefore, using the trapezoidal rule (i.e., Crank–Nicolson method), it is possible to discretise the continuous-time expression of the state space in Equation (19) obtaining its discrete-time version as follows:

$$\begin{cases} \chi_{aug}(k+1) = \left(I - \frac{T}{2}A_t\right)^{-1}\left[\left(I + \frac{T}{2}A_t\right)\chi_{aug}(k) \right. \\ \qquad\qquad \left. + \mathrm{T}(B_t u_{mv}(k) + D_t u_{dist}(k))\right] \\ Y(k) = C_t \chi_{aug}(k) \end{cases} \tag{24}$$

where T represents the sampling time and I is the identity matrix. To improve computational efficiency, it is also important to define an analytic Jacobian for the state function, with which the controller can calculate changes in the function more quickly and more precisely the objective and constraints, avoiding rounding errors and ensuring better convergence of the optimization algorithm. Therefore, the state and input Jacobians, respectively $J(\chi_{aug}, u, t)$ and $G(\chi_{aug}, u, t)$, of the state function $f(\chi_{aug}(t), u(t), t)$ described in Equation (17) can be calculated as:

$$\begin{aligned} J(\chi_{aug}, u, t) &= \frac{\partial f(\chi_{aug}(t), u(t), t)}{\partial \chi_{aug}} \\ G(\chi_{aug}, u, t) &= \frac{\partial f(\chi_{aug}(t), u(t), t)}{\partial u} \end{aligned} \tag{25}$$

where $\chi_{aug} = \begin{bmatrix} \dot{y} & \dot{\psi} & \dot{x} & \ddot{x} & e_{lat} & e_{yaw} & e_{vel} \end{bmatrix}^T$ and $u = \begin{bmatrix} a_f & \delta_f \end{bmatrix}^T$, obtaining:

$$J = \begin{bmatrix} -\frac{2}{m}\frac{C_{lat,f}-C_{lat,r}}{\dot{x}^2} & -1-\frac{2}{m}\frac{C_{lat,f}l_f+C_{lat,r}l_r}{\dot{x}^2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{2}{I_z}\frac{C_{lat,f}l_f+C_{lat,r}l_r}{\dot{x}^2} & -\frac{2}{I_z}\frac{C_{lat,f}l_f^2-C_{lat,r}l_r^2}{\dot{x}^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{26}$$

$$G = \begin{bmatrix} 0 & \frac{2}{m}C_{lat,f} \\ 0 & \frac{2}{I_z}C_{lat,f}l_f \\ 0 & 0 \\ \frac{1}{\tau} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{27}$$

4.3. Cost function

The goal of the MPC-based tracking strategy is to minimize the error between the predicted output variables and the reference values. This ensures that vehicles can accurately follow the predefined trajectory while maintaining both lateral and longitudinal stability. The generic quadratic cost function commonly used in NMPC can be written as follows:

$$J_{cost} = J_{tracking} + J_{control\ effort} + J_{control\ rate} + J_{penalise} \tag{28}$$

where:

- $J_{tracking}$ is related to minimization of tracking errors compared to reference values:

$$J_{tracking} = \sum_{j=1}^{n_y}\sum_{i=1}^{p}\left\{\frac{\omega_{i,j}^y}{s_j^y}\left[r_j(k+i|k) - y_j(k+i|k)\right]\right\}^2 \tag{29}$$

  where $y_j(k+i|k)$ is the predicted output, $r_j(k+i|k)$ is the reference value, $\omega_{i,j}^y > 0$ and $s_j^y > 0$ are the weights and the scaling factors of the output, n_y is the number of tracked outputs and $p > 0$ is the prediction horizon of the controller.
- $J_{control\ effort}$ limits the control effort by optimizing the control variables while keeping them close to the target values:

$$J_{control\ effort} = \sum_{j=1}^{n_u}\sum_{i=0}^{p-1}\left\{\frac{\omega_{i,j}^u}{s_j^u}\left[u_j(k+i|k) - u_{j,target}(k+i|k)\right]\right\}^2 \tag{30}$$

  where $u_j(k+i|k)$ is the control input, $u_{j,target}(k+i|k)$ is the desired target, $\omega_{i,j}^u > 0$ and $s_j^u > 0$ are the weights and the scaling factors of the control input and n_u is the number of control inputs.
- $J_{controlrate}$ limits the width of the variation of the control actions in successive steps ensuring smooth control actions:

$$J_{control\ rate} = \sum_{j=1}^{n_u}\sum_{i=0}^{p-1}\left\{\frac{\omega_{i,j}^{\delta_u}}{s_j^u}\left[u_j(k+i|k) - u_j(k+i-1|k)\right]\right\}^2 \tag{31}$$

  where $\omega_{i,j}^{\delta_u} > 0$ is the weight of the control action variation and $u_j(k+i-1|k)$ is the control action at the previous step.
- $J_{penalise}$ is associated with penalizing constraints if the problem is unsolvable by introducing a soft constraint mechanism:

$$J_{penalise} = \rho_\epsilon \epsilon_k^2 \tag{32}$$

  where ρ_ϵ is the penalizing weight and ϵ_k is a slack variable.

In our case, the cost function J^{cost}_{leader} for the leading vehicle can be constructed as follows:

$$J^{cost}_{k,leader} = \sum_{k=0}^{N_p} (Y_k^T Q Y_k + u_k^T R u_k) = \sum_{k=0}^{N_p} \left[(C_t x_k - y_{ref,k})^T Q (C_t x_k - y_{ref,k}) + u_k^T R u_k \right] \tag{33}$$

where $Q \in \mathbb{R}^{3\times3}$ and $R \in \mathbb{R}^{2\times2}$ are diagonal positive definite weighting matrices of the controlled outputs and inputs, which are presented in the following equations, respectively, Equations (34) and (35):

$$Q_i = \begin{bmatrix} Q_{e_{lat}} & 0 & 0 \\ 0 & Q_{e_{yaw}} & 0 \\ 0 & 0 & Q_{e_{vel}} \end{bmatrix}, \quad with \quad i = 1, 2, ..., N_p \tag{34}$$

$$R_i = \begin{bmatrix} R_{a_x} & 0 \\ 0 & R_{\delta_f} \end{bmatrix}, \quad with \quad i = 1, 2, ..., N_c - 1 \tag{35}$$

where N_p represents the prediction horizon, i.e. the number of future control intervals that the controller considers in solving the optimization problem, predicting how the system will evolve in the following steps, and N_c represents the control horizon, i.e. the number of time steps for which the control actions must be optimized to control the evolution of the system. The value of N_c must be such that $N_c \ll N_p$, both to have a lower computational cost and to guarantee the internal stability of the controller. In particular, we decided to handle the constraints separately because they must be rigorously applied, ensuring the feasibility and efficiency of the solver. Consequently, the optimization problem can finally be defined as follows:

$$\begin{aligned} \min_{u=[a_x \delta_f]^T} \quad & J^{cost}_{k,leader} \\ \text{s.t.} \quad & a_{x,\min} \le a_x \le a_{x,\max} \\ & \delta_{f,\min} \le \delta_f \le \delta_{f,\max} \end{aligned} \tag{36}$$

where $a_x[m/s^2] \in [-2, 1]$ and $\delta_f[rad] \in [-0.4, 0.4]$. Following the previous steps, we can define the follower vehicle cost function $J_{follower}$ in the same way. The only difference consists in the definition of the errors used for the cost function; in particular, we will have:

$$e_{lat} = -(x_{follower} - x_{leader})\sin(\theta) + (y_{follower} - y_{leader})\cos(\theta) \tag{37}$$

$$e_{yaw} = \psi_{follower} - \psi_{leader} \tag{38}$$

$$e_{vel} = \dot{x}_{leader} - \dot{x}_{follower} \tag{39}$$

where in this case the reference values of the (x, y, ψ) coordinates are the current ones of the leader vehicle and the angle θ is calculated as the difference between the current position of the follower vehicle and that of the leader. In particular, to simulate the mechanical connection between vehicles, the system calculates the current distance between the two vehicles at each time step and uses a PI control to evaluate the reference speed to be sent to the follower NMPC based on the distance error. In this way, the reference speed value $\dot{x}_{leader}$ for the follower vehicle is calculated as follows:

$$\dot{x}_{leader}(t) = K_p e_{dist}(t) + K_i \int_0^t e_{dist}(t) d\tau \tag{40}$$

where K_p and K_i are, respectively, the proportional and integral gains of the PI controller and $e_{dist} = d_{curr} - d_0$ is the distance error between the two vehicles with d_{curr} current distance and d_0 reference distance (in this case $d_0 = 6\ m$).

5. Results and discussion

To verify the effectiveness of the proposed controller, numerous simulations were conducted in the MATLAB/Simulink environment, examining different vehicle trajectories and conditions. The controller was first tested in the case of a single vehicle to verify its effectiveness in the trajectory tracking control application, the general architecture of which is highlighted in Figure 6.

Therefore, two different scenarios were considered: In the first scenario (Figure 7a), an ideal trajectory generated by parametric equations is considered. However, in the second scenario (Figure 7b), the actual path was recorded by a GNSS device (25 *Hz*) mounted on the vehicle used as a reference, whose parameters can be seen in Table 2 and were then used as the desired path. For brevity and given the focus of the article, only the scenarios for the application of the convoy will be shown.

As regards the convoy application between two vehicles, a second vehicle has been introduced with the same characteristics as the first and at a fixed distance d_0 from it in a leader-follower logic. The overall control architecture is schematized in Figure 8, where the leader vehicle communicates to the follower information relating to its status, such as position, longitudinal speed, orientation, and curvature of the traveled trajectory.where: (x_{ref}, y_{ref}) and ρ_{ref} denote the reference path and the trajectory curvature, respectively; v_{ref} and a_{ref} are the reference speed and acceleration; $e_{lat_i}, e_{vel_i}, e_{yaw_i}$ are the lateral, speed, and yaw errors of

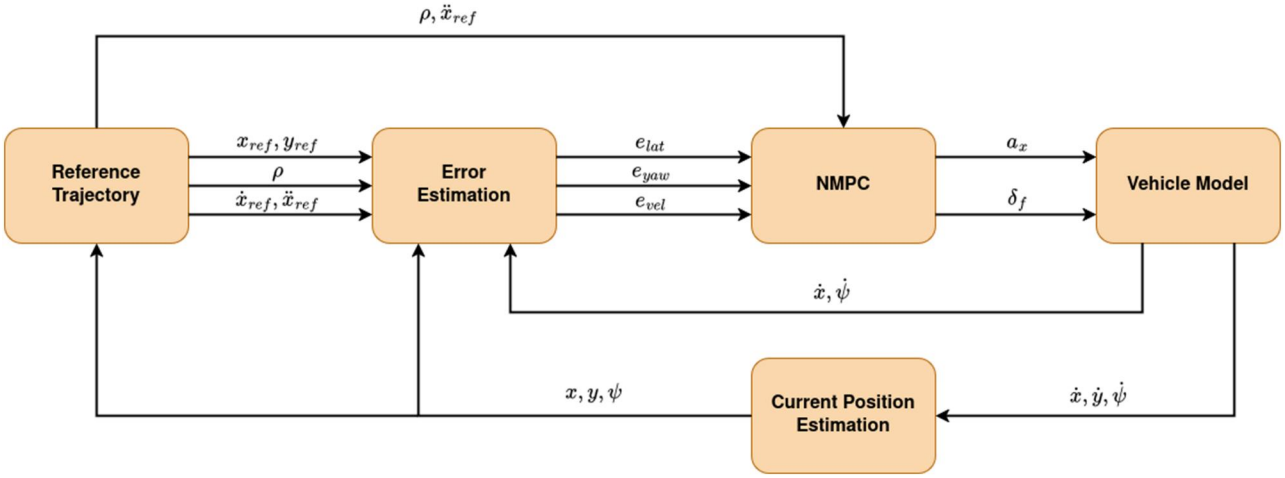


Figure 6. Overall control architecture for single vehicle trajectory tracking.

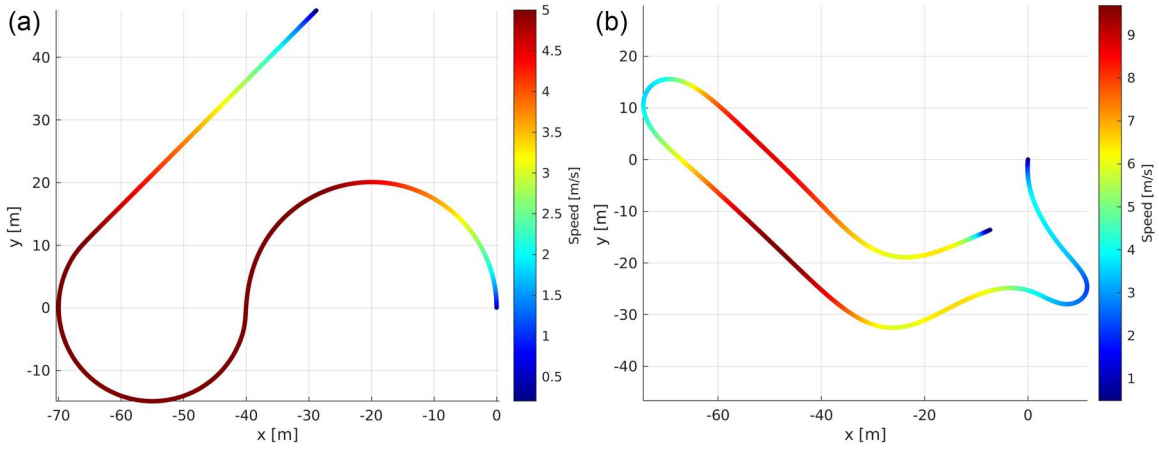


Figure 7. Tested trajectories.

Table 2. Vehicle model parameters.

m [kg]	l_f [m]	l_r [m]	I_z [kg · m²]	v_{max} [m/s]	$C_{lat,f}$ [N/rad]	$C_{lat,r}$ [N/rad]
450	0.8	0.8	270	7	10000	9000

vehicle $i \in \{l, f\}$ (leader, follower) with respect to the trajectory reference; a_{x_i} and δ_i represent the throttle/brake and steering commands of vehicle i; (x_i, y_i) and $(v_{x_i}, v_{y_i}, \dot{\psi}_i)$ are the current position, linear, and angular velocities of vehicle i; ρ_{est} is the estimated curvature of the leader's trajectory; and e_{dist} is the inter-vehicle distance error.

The follower vehicle stores this data in a memory buffer, from which it can subsequently extract reference values according to its current location, using them for tracking. Consequently, the second vehicle has no a priori knowledge of the route or the speed profile to follow. The control logic is structured in such a way that, based on the reference data taken from the buffer and the errors detected, the follower determines the curvature and the reference speed necessary to minimize such errors. These values are then used as input to the NMPC control. The same controller designed for the leader was used for the follower vehicle. This choice was motivated by the need to allow vehicles to operate in variable configurations in the future, enabling them to assume the roles of leader and follower indifferently at any time. Therefore, the effectiveness of the system was evaluated in both scenarios, with the objective of considering the future implementation of different control strategies depending on the specific function of the vehicle, switching between the controls based on the assigned task. The operation of the control architecture can be divided into two main phases:

Phase 1: Initialization – The convoy is initialized with the vehicles aligned and physically connected at the pre-established target distance. In this phase, the leader begins to move, and its trajectory data (e.g., position, orientation, and curvature) are recorded in sequence into the FIFO buffer. The follower's controller is active, but its objective is simplified: it travels in

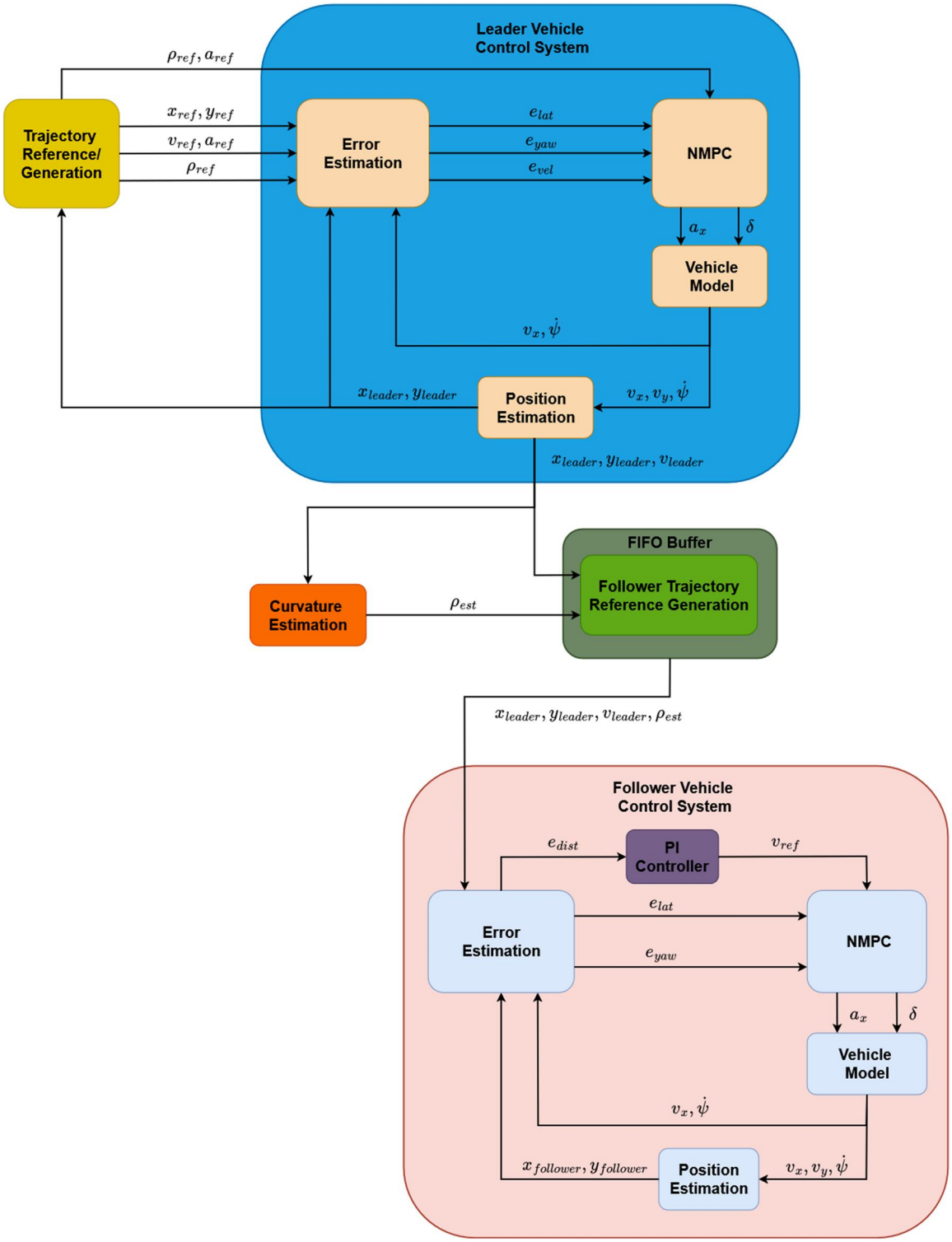


Figure 8. Overall control architecture for convoy of two vehicles.

a straight line (target curvature set to zero), with the PI controller managing its speed to maintain the initial distance. This phase continues until the follower's position is within a predefined threshold of the leader's starting point, at which time the system transitions to Phase 2.

Phase 2: Path Replication and Distance Control – Once the follower reaches the leader's starting point, it transitions to full path-following mode. The NMPC controller now receives two distinct reference signals sourced from different components of our architecture:

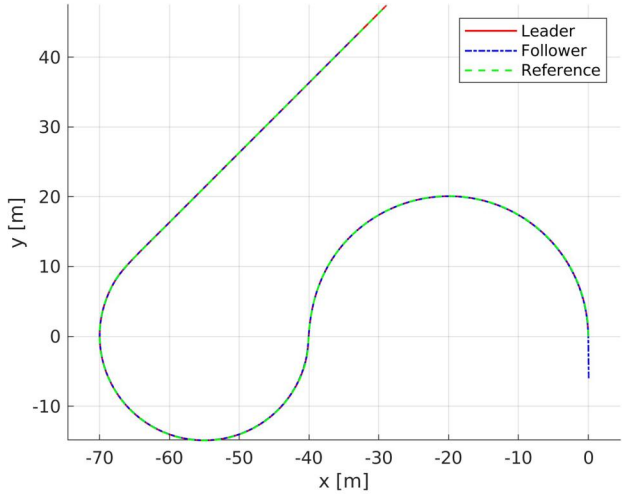
- **Path Reference (from FIFO Buffer):** The follower tracks the leader's geometric path, which is read from the FIFO buffer. The specific reference waypoint and target curvature are selected from the buffer's stored data by finding the path point that minimizes the distance to the follower's current position. This "path replication" ensures the follower precisely follows the leader's track, preventing "cutting-corners."
- **Speed Reference (from PI Controller):** The target speed for the follower is not taken from the buffer. Instead, it is calculated in real-time by the Proportional-Integral (PI) controller. This controller's input is the current distance error (the difference between the desired fixed connection length and the actual measured distance). The PI controller's output provides the reference speed for the NMPC.

These two reference values—the path/curvature from the buffer and the reference speed from the PI—are used to calculate the tracking errors that the NMPC controller is tasked with minimizing. This mixed strategy effectively decouples the control problem: the FIFO buffer ensures lateral (path) fidelity, while the PI controller ensures longitudinal (distance) stability.

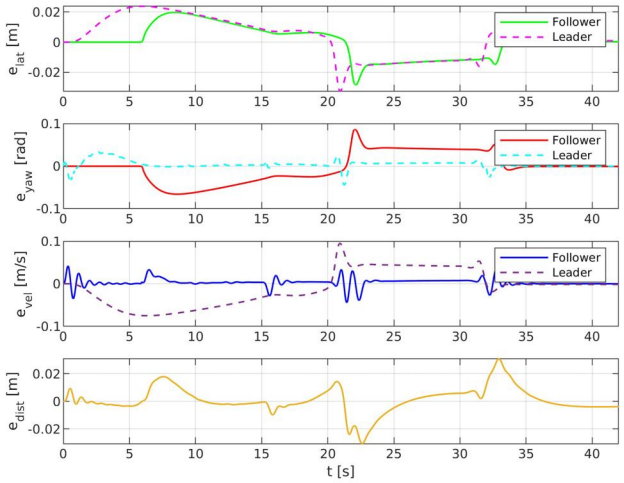
Finally, the proposed control architecture was tested in both previous scenarios.

5.1. Scenario 1

In Scenario 1 (SIL), the proposed control architecture demonstrates excellent performance in accurately following the reference trajectory for both vehicles, as shown in Figure 9a.



(a) Scenario 1 trajectory.



(b) Lateral, Yaw, Speed and Distance errors in Scenario 1.

Figure 9. Scenario 1.

In Figure 9b, the lateral, yaw, speed, and inter-vehicle distance errors are shown. Both vehicles exhibit excellent trajectory-following performance, maintaining near-zero lateral and yaw errors throughout the scenario, indicating precise adherence to the planned path. Their velocity variations are relatively small and likely reflect the adjustments needed to navigate the trajectory, particularly at discontinuities. During continuous trajectory segments, both vehicles maintain small lateral and yaw errors, typically within the range of $\pm 0.02\ m$ and $\pm 0.01\ rad$, respectively, demonstrating effective path following. Speed errors also remain low, generally within $\pm 0.01\ m/s$. However, discontinuities in the trajectory (due to the connection points between different routes), particularly around $t = 20\ s$, induce transient deviations in the convoy's performance but are still relatively small in absolute terms (for example, lateral errors with a peak around $\pm 0.02\ m$, yaw errors around $\pm 0.08\ rad$, and distance errors around $\pm 0.02\ m$), demonstrating the robustness of the proposed architecture.

Figure 10 presents the control inputs generated by the NMPC during the ideal Software-in-the-Loop (SIL) test. As shown, both the longitudinal acceleration (a_x) and the steering angle (δ_f) are exceptionally smooth. This behavior is consistent with the task of tracking a perfect, mathematically defined trajectory that lacks any noise or sudden disturbances. The commands are gradual and precise, demonstrating the controller's stable and effective performance under ideal conditions. Table 3 summarizes the results obtained in Scenario 1 for both vehicles in terms of the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE), and the Mean Absolute Error (MAE).

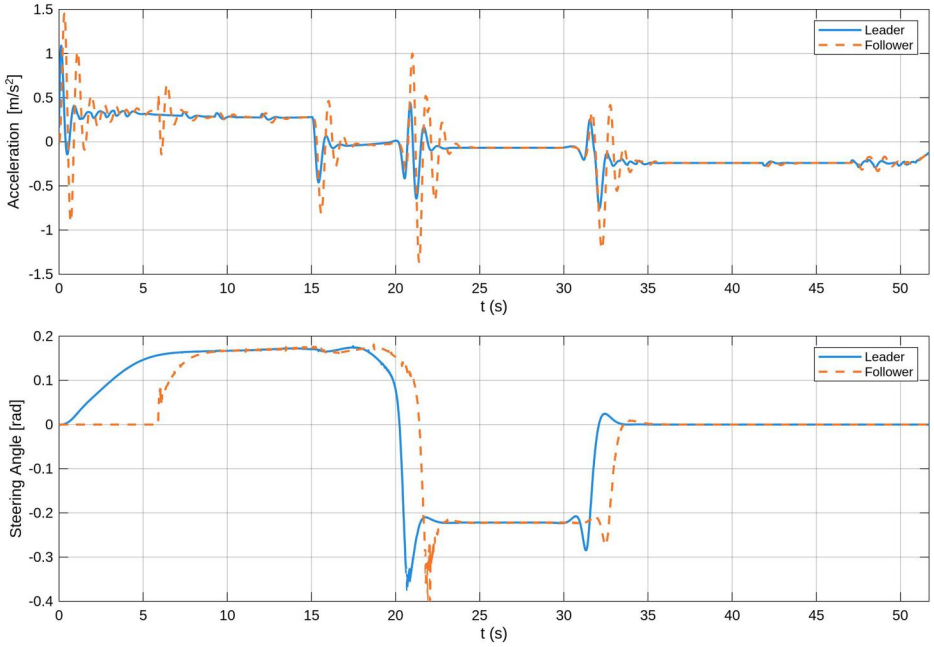


Figure 10. Control inputs in Scenario 1.

Table 3. MSE, RMSE and MAE for follower and leader in Scenario 1.

Errors	MSE	RMSE	MAE
Follower errors			
Lateral (m)	0.0001	0.0094	0.0064
Yaw (rad)	0.0010	0.0313	0.0217
Speed (m/s)	0.0001	0.0091	0.0051
Distance (m)	0.0001	0.0085	0.0057
Leader errors			
Lateral (m)	0.0001	0.0118	0.0088
Yaw (rad)	0.0001	0.0094	0.0054
Speed (m/s)	0.0014	0.0380	0.0282

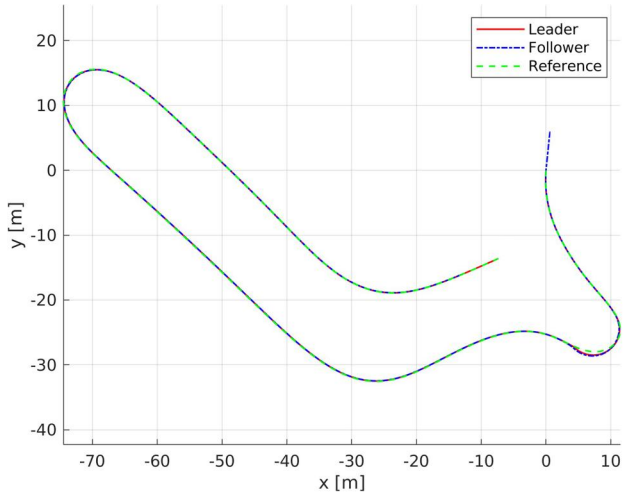
5.2. Scenario 2

In Scenario 2 (MIL), the proposed control architecture continues to exhibit solid performance; however, factors from the real world introduce slightly higher errors compared to the ideal case, as shown globally in Figure 11a.

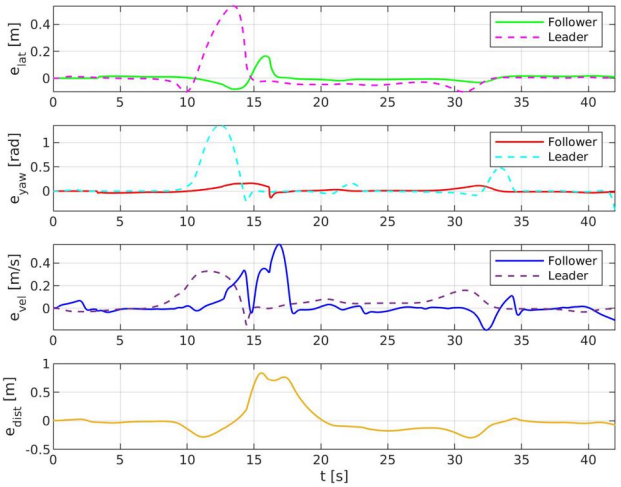
Figure 11b presents the lateral, yaw, speed, and inter-vehicle distance errors for both vehicles. Although the control strategy effectively maintains trajectory tracking, external disturbances, unmodeled dynamics, and measurement noise lead to transient deviations, particularly during trajectory discontinuities. As shown in Figure 11b, both vehicles maintain small lateral and yaw errors for most of the scenario.

The follower typically exhibits lateral deviations within $\pm 0.03\ m$, while the leader experiences slightly larger deviations, particularly around $t = 12 - 15\ s$, where it reaches peaks of approximately $0.4\ m$. Yaw errors follow a similar trend, remaining within $\pm 0.05\ rad$ for most of the scenario but showing transient peaks of approximately $1.0\ rad$ around $t = 12 - 15\ s$ and up to $0.5\ rad$ at $t = 32 - 33\ s$ during abrupt trajectory changes. The velocity errors, illustrated in the third subplot of Figure 11b, remain low under steady-state conditions but increase around discontinuities, particularly near $t = 15\ s$, where the follower shows oscillations reaching $0.4\ m/s$. This suggests that the controller effectively manages speed fluctuations but must compensate for sudden trajectory transitions. The inter-vehicle distance error, shown in the last subplot, remains relatively small but exhibits noticeable variations around $t = 15\ s$, peaking at approximately $0.5\ m$. This behavior indicates a temporary deviation in convoy coordination but demonstrates the control system's ability to rapidly stabilize and maintain safe vehicle spacing. In general, while disturbances induce higher transient errors compared to the ideal scenario, the control architecture remains robust, ensuring that both vehicles quickly recover and maintain stable performance.

In contrast to the SIL case, the control inputs for the Model-in-the-Loop (MIL) test, shown in Figure 12, are visibly more active. The controller is making continuous and more aggressive adjustments to both a_x and δ_f to compensate for the higher-frequency components and imperfections present in the prerecorded GNSS trajectory. This demonstrates the effectiveness of NMPC's in a challenging, realistic scenario. Crucially, the controller is shown to use the full



(a) Scenario 2 trajectory.



(b) Lateral, Yaw, Speed and Distance errors in Scenario 2.

Figure 11. Scenario 2.

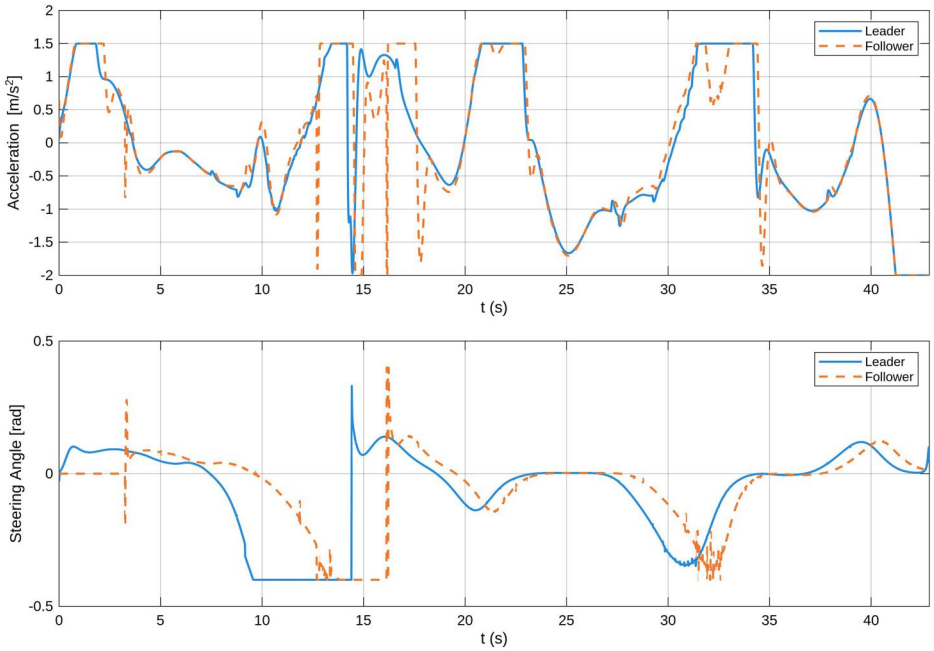


Figure 12. Control inputs in Scenario 2.

Table 4. MSE, RMSE and MAE for follower and leader in Scenario 2.

Errors	MSE	RMSE	MAE
Follower errors			
Lateral (m)	0.0010	0.0311	0.0179
Yaw (rad)	0.0026	0.0510	0.0321
Speed (m/s)	0.0140	0.1183	0.0577
Distance (m)	0.0569	0.2385	0.1446
Leader errors			
Lateral (m)	0.0144	0.1199	0.0548
Yaw (rad)	0.1565	0.3957	0.1459
Speed (m/s)	0.0102	0.1012	0.0616

available control authority, with the longitudinal acceleration a_x clearly reaching its predefined constraint of 2 m/s^2. The steering angle δ_f also operates over a much wider range, approaching its own limit of 0.4 *rad*. This confirms that the NMPC is successfully and robustly handling the task by optimizing its commands right up to the physical limits of the actuators without ever exceeding them. The signals remain stable and do not exhibit chattering, validating the controller's design.

Table 4 summarizes the error metrics (MSE, RMSE, and MAE) for both vehicles in the real trajectory scenario, providing a quantitative assessment of the controller's effectiveness using Model-In-the-Loop (MIL) validation.

Table 5 summarizes the results obtained for both scenarios and vehicles.

Table 5. Error metrics comparison for follower and leader in both scenarios.

Errors	MSE	RMSE	MAE
Ideal trajectory			
Follower			
Lateral (m)	0.0001	0.0094	0.0064
Yaw (rad)	0.0010	0.0313	0.0217
Speed (m/s)	0.0001	0.0091	0.0051
Distance (m)	0.0001	0.0085	0.0057
Leader			
Lateral (m)	0.0001	0.0118	0.0088
Yaw (rad)	0.0001	0.0094	0.0054
Speed (m/s)	0.0014	0.0380	0.0282
Real trajectory			
Follower			
Lateral (m)	0.0010	0.0311	0.0179
Yaw (rad)	0.0026	0.0510	0.0321
Speed (m/s)	0.0140	0.1183	0.0577
Distance (m)	0.0569	0.2385	0.1446
Leader			
Lateral (m)	0.0144	0.1199	0.0548
Yaw (rad)	0.1565	0.3957	0.1459
Speed (m/s)	0.0102	0.1012	0.0616

6. Conclusion

In this paper, a mixed PI-NMPC control strategy was proposed for the trajectory tracking control of a mechanically connected automated vehicle convoy. The proposed mixed PI-NMPC control strategy ensures coordinated motion in a convoy of two vehicles, where both the leader and the follower implement NMPC for trajectory tracking. The NMPC predicts future states and optimizes steering and throttle/brake commands while accounting for the nonlinear vehicle dynamics and constraints. Additionally, a PI controller regulates the inter-vehicle distance, ensuring that the follower accurately tracks the leader. This approach improves stability, adherence to the trajectory, and adaptability to varying driving conditions. The effectiveness of the control architecture was verified in the MATLAB/Simulink® environment by performing simulations on ideal and real datasets, representative of different operational scenarios. The simulation results demonstrated excellent performance in the ideal case, guaranteeing accurate tracking of the reference trajectory and a distance error between the two vehicles limited to an interval of $\pm 0.05\ m$. However, in the real scenario, the distance error turns out to be larger (up to approximately $0.80\ m$), mainly due to constraints imposed on the dynamics and characteristics of the trajectory. The results obtained from the simulations in the real scenario therefore highlight the limits and problems that a possible mechanical coupling between the vehicles would impose on the system. A rigid connection mechanism would, in fact, be incompatible with the presence of distance errors, although small, due to the large forces that can be exchanged between the vehicles and the coupling. It therefore becomes necessary to rethink the type of coupling for a future real application, evaluating the adoption of an elastic rather than rigid connection between the vehicles, in order to tolerate the inevitable deviations in the desired distance between the vehicles, due both to the natural delays of the mechanical actuation systems of the vehicles and to the need for the controller to adapt in real time to variations in the longitudinal speed of the leader, which presents a profile that varies over time and is not known a priori.

Nomenclature

CACC	Cooperative Adaptive Cruise Control
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
FIFO	First Input First Output
GNSS	Global Navigation Satellite System
MAE	Mean Absolute Error
MD	Measured Disturbances
MIL	Model-In-the-Loop
MIMO	Multi-Input-Multi-Output
MPC	Model Predictive Control
MO	Measured Outputs
MSE	Mean Squared Error
MV	Manipulated Variables
NMPC	Nonlinear Model Predictive Control
OV	Output Variables
PI	Proportional–Integral
RMSE	Root Mean Squared Error
V2V	Vehicle-To-Vehicle

Notes

1. From now on, the term "convoying" will be used to describe a convoy of mechanically connected automated vehicles.
2. Patented by Università degli Studi di Firenze, priority number IT201900012120A1 Alessandrini and Cignini (2021)

Acknowledgements

The authors would like to express their gratitude to ENEA (National Agency for New Technologies, Energy and Sustainable Economic Development) for their valuable support in the experimentation phase of this work. Their assistance and resources have been instrumental in the successful implementation and validation of our research.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The project has received funding from the Sustainable Mobility Center (MOST) within the framework of the Italian PNRR (CUP B13C22001000001).

References

Adriano, A., Berzi, L., Cignini, F., Favilli, T., Genovese, A., Lidozzi, A., & Staffa, D. (2021). Design of a new on-board energy storage and conversion system for a fast charging urban transport electric bus. In *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)* (pp. 1–6). https://doi.org/10.1109/EEEIC/ICPSEurope51590.2021.9584573

Alessandrini, A., & Cignini, F. (2021). *Mechanical coupling device between vehicles, in particular for convoys of automatic vehicles.* https://patents.google.com/patent/IT201900012120A1

Alessandrini, A., Ortenzi, F., Berzi, L., Gulino, M.-S., Cignini, F., & Pugi, L. (2023). An innovative convoying and power management system for public transportation. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)* (pp. 1–5). https://doi.org/10.1109/VTC2023-Spring57618.2023.10200691

Alessandrini, A., Ortenzi, F., Cignini, F., Gulino, M. S., Franci, M., Berzi, L., Pugi, L. (2024). Preliminary design and simulation of a return current collector for an innovative trambus system. In *International Symposium on Industrial Engineering and Automation* (pp. 85–92).

Arvin, R., Khattak, A. J., Kamrani, M., & Rio-Torres, J. (2021). Safety evaluation of connected and automated vehicles in mixed traffic with conventional vehicles at intersections. *Journal of Intelligent Transportation Systems*, *25*(2), 170–187. https://doi.org/10.1080/15472450.2020.1834392

Bayuwindra, A., Lefeber, E., Ploeg, J., & Nijmeijer, H. (2020). Extended look-ahead tracking controller with orientation-error observer for vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, *21*(11), 4808–4821. https://doi.org/10.1109/TITS.2019.2947348

Chelbi, N. E., Gingras, D., & Sauvageau, C. (2022). Worst-case scenarios identification approach for the evaluation of advanced driver assistance systems in intelligent/autonomous vehicles under multiple conditions. *Journal of Intelligent Transportation Systems*, *26*(3), 284–310. https://doi.org/10.1080/15472450.2020.1853538

Dayi, Q., Yanfeng, J., Tao, W., Bin, L., & Lewei, H. (2022). Research on coordinated control of vehicle's speed in new mixed traffic flow. *Journal of Intelligent Transportation Systems*, *26*(6), 704–716. https://doi.org/10.1080/15472450.2021.1973897

Dixit, S., Fallah, S., Montanaro, U., Dianati, M., Stevens, A., Mccullough, F., & Mouzakitis, A. (2018). Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control*, *45*, 76–86. https://www.sciencedirect.com/science/article/pii/S136757881730130X https://doi.org/10.1016/j.arcontrol.2018.02.001

Feng, J., Pi, D., Wu, Y., & Zhang, C. (2025). Research on coordinated control method of trajectory tracking for wheel corner module articulated vehicle. In *International Conference on Computer Vision and Augmented Reality (CVAR 2025)* (Vol. 13801, pp. 391–401). https://doi.org/10.1117/12.3076915

Gao, Y. (2014). *Model predictive control for autonomous and semiautonomous vehicles*. University of California.

Gulino, M.-S., Damaziak, K., Fiorentino, A., & Vangi, D. (2024). Handling inevitable collision states by advanced driver assistance systems functions: software-in-the-loop performance assessment of an injury risk-based logic in a "lane departure" scenario. *Journal of Intelligent Transportation Systems*, *28*(6), 1011–1031. https://doi.org/10.1080/15472450.2023.2277713

Hou, K., & Giannopoulos, G. (2024). Modeling the deployment and management of large-scale autonomous vehicle circulation in mixed road traffic conditions considering virtual track theory. *Future Transportation*, *4*(1), 215–235. https://doi.org/10.3390/futuretransp4010011

Huang, Y., Zheng, X., Han, T., & Tan, W. (2025). Coordinated control of trajectory tracking and lateral stability for distributed electric-driven buses. *World Electric Vehicle Journal*, *16*(10), 576. https://doi.org/10.3390/wevj16100576

Kianfar, R., Ali, M., Falcone, P., & Fredriksson, J. (2014). Combined longitudinal and lateral control design for string stable vehicle platooning within a designated lane. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 1003–1008).

Li, L., Li, J., & Zhang, S. (2021). Review article: State-of-the-art trajectory tracking of autonomous vehicles. *Mechanical Sciences*, *12*(1), 419–432. https://ms.copernicus.org/articles/12/419/2021/ https://doi.org/10.5194/ms-12-419-2021

Lyu, N., Duan, Z., Ma, C., & Wu, C. (2021). Safety margins–a novel approach from risk homeostasis theory for evaluating the impact of advanced driver assistance systems on driving behavior in near-crash events. *Journal of Intelligent Transportation Systems*, *25*(1), 93–106. https://doi.org/10.1080/15472450.2020.1795846

Marin, A. L., Martínez-Plumed, F., Makridis, M. A., Tansini, A., Pulvirenti, L., Suarez Corujo, J., Komnos, D., Ramírez-Quintana, M. J., Monserrat, C., & Fontaras, G. (2025). Follow the leader: a deep reinforcement learning framework for safe and efficient autonomous car-following. *Journal of Intelligent Transportation Systems*. Advance online publication. https://doi.org/10.1080/15472450.2025.2576907

Mohtavipour, S. M., & Mollajafari, M. (2021). An analytically derived reference signal to guarantee safety and comfort in adaptive cruise control systems. *Journal of Intelligent Transportation Systems*, *25*(1), 1–20. https://doi.org/10.1080/15472450.2019.1619559

Nahavandi, S., Mohamed, S., Hossain, I., Nahavandi, D., Salaken, S. M., Rokonuzzaman, M., Ayoub, R., & Smith, R. (2022). Autonomous convoying: A survey on current research and development. *IEEE Access*, *10*, 13663–13683. https://doi.org/10.1109/ACCESS.2022.3147251

Pacejka, H. B., & Bakker, E. (1992). The magic formula tyre model. *Vehicle System Dynamics*, *21*(1), 1–18. https://doi.org/10.1080/00423119208969994

Pauca, O., Caruntu, C. F., & Maxim, A. (2020). Trajectory planning and tracking for cooperative automated vehicles in a platoon. In *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 769–774). https://doi.org/10.1109/ICSTCC50638.2020.9259780

Petrov, P. (2008). A mathematical model for control of an autonomous vehicle convoy. *WSEAS Transactions on Systems and Control*, *3*(9), 835–848.

Polack, P., Altché, F., d'Andréa Novel, B., de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?

In *2017 IEEE Intelligent Vehicles Symposium (iv)* (pp. 812–818).

Polack, P., Altché, F., D'Andrea-Novel, B., & de La Fortelle, A. (2018). Guaranteeing consistency in a motion planning and control architecture using a kinematic bicycle model. In *2018 Annual American Control Conference (ACC)* (pp. 3981–3987). https://doi.org/10.23919/ACC.2018.8430886

Rajamani, R. (2011). *Vehicle dynamics and control*. Springer Science & Business Media.

Shladover, S. E. (2018). Connected and automated vehicle systems: Introduction and overview. *Journal of Intelligent Transportation Systems*, *22*(3), 190–200. https://doi.org/10.1080/15472450.2017.1336053

Tapani, A. (2012). Vehicle trajectory effects of adaptive cruise control. *Journal of Intelligent Transportation Systems*, *16*(1), 36–44. https://doi.org/10.1080/15472450.2012.639641

Vangi, D., Virga, A., & Gulino, M.-S. (2019). Combined activation of braking and steering for automated driving systems: adaptive intervention by injury risk-based criteria. *Procedia Structural Integrity*, *24*, 423–436. https://doi.org/10.1016/j.prostr.2020.02.039

Vasconcelos Filho, E., Severino, R., Koubaa, A., & Tovar, E. (2020). An integrated lateral and longitudinal look ahead controller for cooperative vehicular platooning. In *International Conference on Intelligent Transport Systems* (pp. 142–159).

Wang, H., Liu, B., Ping, X., & An, Q. (2019). Path tracking control for autonomous vehicles based on an improved MPC. *IEEE Access*, *7*, 161064–161073. https://doi.org/10.1109/ACCESS.2019.2944894

Yao, Y., Zhao, X., Wu, Y., Zhang, Y., & Rong, J. (2021). Clustering driver behavior using dynamic time warping and hidden Markov model. *Journal of Intelligent Transportation Systems*, *25*(3), 249–262. https://doi.org/10.1080/15472450.2019.1646132