Control Engineering Practice 154 (2025) 106153

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Enhancing truck platooning efficiency and safety—A distributed Model Predictive Control approach for lane-changing manoeuvres

Beatriz Lourenço [d], Daniel Silvestre [a,b,c,*,1]

[a] School of Science and Technology from the NOVA University of Lisbon (FCT/UNL), Largo da Torre, Caparica, 2829-516, Lisbon, Portugal
[b] COPELABS, Lisbon, Portugal
[c] Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
[d] Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

ARTICLE INFO

Keywords:
Truck platooning
Nonlinear Model Predictive Control
Cooperative Adaptive Cruise Control
Obstacle avoidance

ABSTRACT

The advent of autonomous driving technologies has paved the way for notable advancements in the realm of transportation systems. This paper explores the dynamic field of truck platooning, focusing on the development of a Nonlinear Model Predictive Control (NMPC) approach within a Cooperative Adaptive Cruise Control (CACC) framework. The research tackles the critical challenges in obstacle avoidance and lane-changing manoeuvres. The core contribution of this work lies in the development and implementation of a novel NMPC algorithm tailored to platoon control. This framework integrates a penalty soft constraint to guarantee obstacle avoidance and maintain platoon coherence while optimising control inputs in real-time. Several experiments, including static and dynamic obstacle avoidance scenarios, validate the efficacy of the proposed approach. In all experiments, the vehicles closely follow one another, resulting in smooth trajectories for all system states and control input signals. Even in the event of abrupt braking by the ego vehicle, the platoon remains cohesive. Moreover, the proposed NMPC proves to be computationally efficient when compared to the state-of-the-art.

1. Introduction

Autonomous driving is revolutionising how we interact with vehicles, with potentially far-reaching impacts on transportation. While much attention is on automation, there is a broader impact on transportation systems, particularly in the realm of road-based freight transport. Truck platooning, where a coordinated convoy of vehicles travels closely, emerges as a promising solution (Porfiri, Roberson, & Stilwell, 2006). It offers reduced aerodynamic drag, improved fuel efficiency, lower emissions, enhanced safety, and reduced traffic congestion. Moreover, these advancements in truck platooning hold significant relevance in the context of the freight and distribution sector, which is increasingly grappling with the critical issue of a shortage of qualified professional drivers (Ji-Hyland & Allen, 2020).

The European Truck Platooning Challenge in 2017 revealed a potential 15% reduction in fuel consumption, dispelling safety concerns and envisioning multi-brand platooning technology by 2025 (Challenge, 2016). Studies have delved into platoon dynamics and communication protocols. One critical challenge is navigating platoons during dynamic scenarios like obstacle avoidance and lane-changing. Effective control strategies are needed for real-world deployment.

The literature regarding safety in platooning has considered several aspects that are relevant for a full implementation, including platoon coordination and communication reliability. In the pursuit of refining existing strategies, studies have delved into the intricacies of platoon shape, rearrangement, and formation dynamics, as exemplified by the work of Maiti, Winter, Kulik, and Sarkar (2020). Furthermore, a parallel line of research has studied communication protocols, striving to engineer seamless and reliable mechanisms capable of addressing the challenges posed by the variable range between vehicles, as demonstrated in the investigation by Won (2022). These studies collectively highlight some of the intricate factors that must be managed to successfully deploy truck platooning.

Another alternative for the coordination of platoons can be to employ Model Predictive Control (MPC) given its ability to handle constraints that can translate safety during dynamics maneuvers of overtaking and lane changing (Hernandez, Desideri, Ionescu, Keyser,

* Corresponding author at: School of Science and Technology from the NOVA University of Lisbon (FCT/UNL), Largo da Torre, Caparica, 2829-516, Lisbon, Portugal.
E-mail addresses: beatriz.b.lourenco@tecnico.ulisboa.pt (B. Lourenço), dsilvestre@isr.tecnico.ulisboa.pt (D. Silvestre).

[1] This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through project FirePuma (https://doi.org/10.54499/PCIF/MPG/0156/2019), through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UIDB/50009/2020 (10.54499/UIDB/50009/2020), and through COPELABS, University Lusófona project UIDB/04111/2020 (10.54499/UIDB/04111/2020).

https://doi.org/10.1016/j.conengprac.2024.106153
Received 19 February 2024; Received in revised form 6 October 2024; Accepted 23 October 2024
Available online 12 November 2024

Lemort, & Quoilin, 2016). Using a model of the dynamics, MPC generates solutions that minimise a cost function using both the future control actions and states. In a similar direction, the MPC techniques to achieve navigation of autonomous vehicles in cluttered environments are related to this study. In particular, the work in Silvestre and Ramos (2023) replaces the non-convex constraints associated with the obstacles by a set of linear inequalities added to keep the vehicle position from reaching the obstacle.

In Thormann, Schirrer, and Jakubek (2022) a distributed model predictive control (MPC) concept was proposed, enabling safe dense spacing with minimal communication requirements and robustness against communication loss. This approach introduces a safety extension that separates safety constraints from tracking control goals, allowing for agreed-upon behaviour during limited decelerations. Utilising driving corridors based on position errors, the system selects appropriate control modes or triggers prediction updates for following vehicles. The authors tested using realistic vehicle dynamics simulations for a platoon in scenarios such as emergency braking and maneuver tracking amid traffic disturbances, demonstrating effectiveness despite model errors, implicit collision safety, and string stability with low communication demands. The main difference with the proposed technique in this paper is that hard constraints can render the problem unfeasible and generate aggressive maneuvers, which we aim to mitigate.

Other works have considered different maneuvers. The work in Chen, Tang, Johansson, and Mårtensson (2024) introduces a formation control design for safe platooning and merging of vehicle groups in multi-lane road scenarios. With the leader vehicle independently controlled, the aim is to manage the follower vehicles to maintain a desired lane and constant distance behind the preceding vehicle while avoiding collisions with neighbouring vehicles and road edges. The proposed method uses Control Barrier Functions (CBFs), which is equivalent to the method that we use for comparison as the representative of the state-of-the-art. The idea in Chen et al. (2024) is to have each follower vehicle have two control components: a nominal controller for tracking the neighbouring vehicle, and a collision avoidance mechanism using divergent flow as a dissipative term to reduce relative velocity towards the neighbouring vehicle and road edges without compromising nominal control performance.

In a different field but a similar application, the work in Taborda, Matias, Silvestre, and Lourenço (2024) has shown that the binary decision variables can be incorporated into the MPC to define a mixed integer program but then solved following a heuristic that governs instantiation of these variables to achieve very fast implementations with performance close to the actual solution of the mixed integer program. Such a technique can also be leveraged for the truck platooning problem to decide when to start the maneuver.

The paper (Zhou, Tian, Sheng, Duan, Qu, Zhao, Cao, & Shen, 2022) uses a more traditional technique to incorporate the collision avoidance into the MPC. Namely, by formulating the problem as a minimisation-maximisation, the disturances and other vehicles are treated as an opponent that is attempting to collide whereas the control actuation is competing with those adversarial signals. The main drawback of such a solution is the high computational complexity given that min–max optimisation problems are harder to solve in the general case. Along those lines, Johansson, Nekouei, Johansson, and Mårtensson (2023) formulates the platooning problem as a game and shows how algorithms to compute best strategies in games can be used for platooning maneuvers. Nonetheless, these controllers typically result in control laws that are not optimised with respect to the cost function like in an MPC framework. Similarly, Sidorenko, Thunberg, Sjöberg, Fedorov, and Vinel (2022) have studied the automatic breaking whereas in this paper we are interested in allowing the solver for the controller to decide whether to break or shift lanes depending on which action has a better cost. The interested reader is referred to the recent survey in Lesch, Breitbach, Segata, Becker, Kounev, and Krupitzer (2022) for a more complete picture regarding platoon control and different aspects of automatic controllers for trucks.

Building upon this landscape of autonomous driving and platooning challenges, this paper addresses the problem of truck platooning navigation involving obstacle avoidance and lane-changing maneuvers when only the ego vehicle is equipped with a LiDAR. Through the lens of an Nonlinear Model Predictive Control (NMPC) approach within a Cooperative Adaptive Cruise Control (CACC) framework on a leader-follower topology, this work seeks to achieve safe and efficient platoon navigation.

One particularity of our proposed solution is the processing of LiDAR data using a set representation and the incorporation of a soft constraint within the MPC that translates that representation of the obstacle. Thus, the current solution is also related to the computation of safe sets like in Alam, Gattami, Johansson, and Tomlin (2014) where the authors are capable of showing that 1.2 m is a safe distance given the typical disturbances and dynamics equations for commercial trucks. Moreover, this estimation can encompass other instances like the use of range and bearing measurements like in Silvestre (2022a) or uncertain parameters as in Silvestre (2022b). Therefore, the soft constraint proposed in this paper can be tunned to match this distance value or be larger if the engineer wants to account for more extreme conditions.

Given the above discussion, the main contributions of this paper can be summarised as follows:

- The paper proposes a distributed Model Predictive Controller that is able to perform lane-changing manoeuvres or breaking in response to the actions of the ego vehicle. The behaviour is not explicitly coded in the solution but rather what the solver can compute to minimise the cost function.
- In order to avoid feasibility issues of the MPC algorithm, it is proposed a soft constraint with an exponential weight. This is a major difference with the state-of-the-art techniques employing solely Control Barrier Functions (CBFs) for safety as the problem can become infeasible either due to conflicting constraints (namely maximum velocity and acceleration, turning rate, etc.) or due to the incorrect selection of the desired rate.

2. Background overview

This section provides the essential background and benchmarks needed to understand and evaluate the research at hand.

2.1. Vehicle modelling

The effectiveness of model-based control is greatly influenced by the accuracy of the models employed. In the context of modelling trucks and heavy-duty vehicles, a kinematic model known as the general n-trailer model is commonly employed Moradi (2022) and Ljungqvist (2020). This model provides a comprehensive representation of the motion and dynamics of these vehicles, taking into account the complex interactions between the tractor and multiple trailers. The system consists of $n+1$ vehicle segments, which include a leading car-like tractor connected to n passive trailers by passive rotary joints as depicted in Fig. 1.

The kinematic model of the general n-trailer system results in the nonholonomic model, i.e. the wheels of the vehicle are assumed to be rolling without slipping, given by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{\upsilon}_0 \\ \dot{\delta}_0 \\ \dot{\theta}_0 \\ \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} = \begin{bmatrix} \upsilon_0 \cos\theta_0 \\ \upsilon_0 \sin\theta_0 \\ u_0 \\ u_1 \\ \upsilon_0 \dfrac{\tan\left(\alpha_0\right)}{l_0} \\ g_1\left(\beta_1, \upsilon_0, \delta_0\right) \\ \vdots \\ g_n\left(\boldsymbol{\beta}, \upsilon_0, \delta_0\right) \end{bmatrix} \tag{1}$$

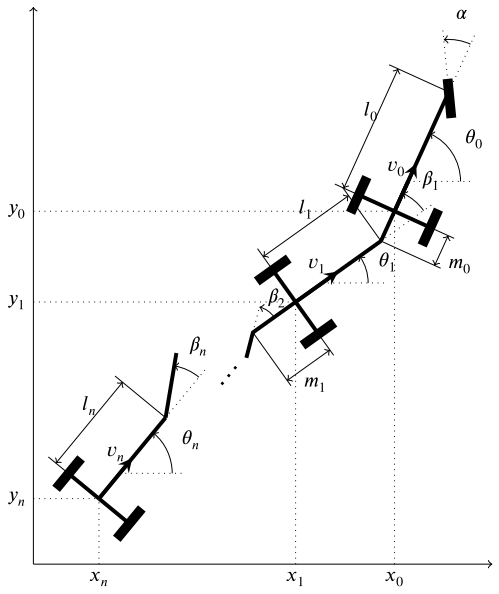


Fig. 1. A schematic description of the general n-trailer with a car-like tractor. *Source:* Inspired and adapted from Lukassek, Völz, Szabo, and Graichen (2021).

where $\boldsymbol{x} = \left[x_0, y_0, v_0, \delta_0, \theta_0, \dots, \theta_n\right] \in \mathbb{R}^3 \times \left(\mathbb{S}^1\right)^{n+2}$, with $\mathbb{S} = (-\pi, \pi]$, represents the state, and $\boldsymbol{u} = \left[u_0, u_1\right] \in \mathbb{R}^2$ the control input with acceleration $\dot{v}_0 = u_0$ and steering rate $\dot{\delta}_0 = u_1$.

The global position $\left[x_0, y_0\right]^\top$ represents the Cartesian coordinates of the vehicle's rear axle midpoint in the fixed world frame. The longitudinal vehicle velocity is denoted as v_0 and the steering angle as δ_0. The states of the trailers are provided by the heading angles $\theta_i, i = 1, \dots, n$ and each segment position can be calculated by

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - \sum_{j=1}^{i} l_j \begin{bmatrix} \cos\theta_j \\ \sin\theta_j \end{bmatrix} - \sum_{j=0}^{i-1} m_j \begin{bmatrix} \cos\theta_j \\ \sin\theta_j \end{bmatrix} \tag{2}$$

with respect to the vehicle-tractor position $\left[x_0, y_0\right]^\top$. Given the dynamics of the trailer heading angles $\theta_i, i = 1, \dots, n$, the difference angles between each segment, i.e., hitching angles can be depicted as

$$\beta_i \triangleq \theta_{i-1} - \theta_i \in \mathcal{B}_i = [-\bar{\beta}_i, \bar{\beta}_i], \quad \bar{\beta}_i \in (0, \pi), \quad i = 1, \dots, n \tag{3}$$

where $\boldsymbol{\beta} = \left[\beta_1, \beta_2, \dots, \beta_n\right]^\top$. The recursive formula for the transformation of the angular velocity formulates the velocity transformation as follows and introduces $\boldsymbol{w} = \left[\dot{\theta}_0, v_0\right]^\top$ and $\boldsymbol{c} = [1, 0]^\top$ together with the transformation matrices

$$\boldsymbol{J}_i\left(\beta_i\right) = \begin{bmatrix} -\dfrac{m_{i-1}\cos\beta_i}{l_i} & \dfrac{\sin\beta_i}{l_i} \\ m_{i-1}\sin\beta_i & \cos\beta_i \end{bmatrix} \quad i = 1, \dots, n, \tag{4}$$

the functions $g_i\left(\boldsymbol{\beta}, v_0, \delta_0\right) : \mathbb{R} \to \mathbb{R}$ are computed recursively

$$\begin{aligned} g_1\left(\beta_1, v_0, \delta_0\right) &= \boldsymbol{c}^\top \boldsymbol{J}_1\left(\beta_1\right) \boldsymbol{w} \\ &\vdots \\ g_n\left(\boldsymbol{\beta}, v_0, \delta_0\right) &= \boldsymbol{c}^\top \boldsymbol{J}_n\left(\beta_n\right) \boldsymbol{J}_{n-1}\left(\beta_{n-1}\right) \dots \boldsymbol{J}_1\left(\beta_1\right) \boldsymbol{w}. \end{aligned} \tag{5}$$

2.2. Nonlinear model predictive control

NMPC is generally formulated as an online optimisation problem for a system with nonlinear dynamics while satisfying a set of both linear and nonlinear constraints (Allgöwer, Findeisen, & Nagy, 2004). Given that a controller is formally implemented through a digital computer that samples the variables of the system and transmits the control action back at discrete time steps, it is advantageous to consider the system specified in discrete time as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{6}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ denote the state and the control input vector at instant k, respectively. The discrete-time model is, however, only an approximation of the continuous-time model.

The objective function in this case is often composed of the sum of a staged cost $\boldsymbol{q}$ and a final cost $\boldsymbol{p}$ such that

$$J\left(\boldsymbol{x}, \boldsymbol{u}\right) = \boldsymbol{p}(\boldsymbol{x}_N) + \sum_{k=0}^{N-1} \boldsymbol{q}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \tag{7}$$

where N denote the discrete-time horizon. Thus, the discrete NMPC optimisation problem can be posed as

$$\begin{aligned} \min_{\boldsymbol{u}(\cdot)} \quad & \boldsymbol{p}(\boldsymbol{x}_N) + \textstyle\sum_{k=0}^{N-1} \boldsymbol{q}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), && k = 0, \dots, N-1, \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0, \\ & \boldsymbol{x}_N \in \mathcal{X}_f, \\ & \boldsymbol{u}_k \in \mathcal{U}, && k = 0, \dots, N-1, \\ & \boldsymbol{x}_k \in \mathcal{X}, && k = 0, \dots, N-1 \end{aligned} \tag{8}$$

here $\boldsymbol{x}_0$ is the starting measured state and $\mathcal{X}_f \subseteq \mathbb{R}^n$ is a *terminal region* that we require the system states to reach at the end of the horizon.

2.2.1. Control barrier functions

A Control Barrier Function (CBF) is a function employed to enforce safety constraints on the states of a dynamical system (Zeng, Li, & Sreenath, 2021). Consider the discrete-time control system intruduced in (6), For safety-critical control, considering a set $\mathcal{C}$ defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$,

$$\begin{aligned} \mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}, \\ \partial\mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) = 0\}, \\ \mathrm{Int}(\mathcal{C}) &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) > 0\}, \end{aligned} \tag{9}$$

$\mathcal{C}$ can be referred to as the safe set and can be regarded as the ensemble of states satisfying distance constraints

$$h(\boldsymbol{x}) \geq 0. \tag{10}$$

In a stricter manner, the function h becomes a CBF in the discrete-time domain if it satisfies the following relation,

$$\Delta h\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \geq -\gamma_k h\left(\boldsymbol{x}_k\right), \quad 0 < \gamma_k \leq 1, \tag{11}$$

where $\Delta h\left(\boldsymbol{x}_k, \boldsymbol{x}_k\right) := h\left(\boldsymbol{x}_{k+1}\right) - h\left(\boldsymbol{x}_k\right)$. Satisfying such constraint implies $h\left(\boldsymbol{x}_{k+1}\right) \geq \left(1 - \gamma_k\right) h\left(\boldsymbol{x}_k\right)$, i.e, the lower bound of control barrier function $h(\boldsymbol{x})$ decreases exponentially at time k with the rate $1 - \gamma_k$. The CBFs are thus designed to guarantee the forward invariance of the safe set $\mathcal{C}$.

If $\gamma(x)$ is close to 1, the system converges to $\partial\mathcal{C}$ slowly but can easily become infeasible. On the other hand, if $\gamma(x)$ is close to 0, the constraint is feasible in a larger domain but can approach $\partial\mathcal{C}$ quickly and become unsafe. In this context, we draw insights from the research conducted by Thirugnanam, Zeng, and Sreenath in Thirugnanam et al. (2022). Their proposed formulation rewrites the CBF constraint in (11) such that

$$h(\boldsymbol{x}_{k+1}) \geq \boldsymbol{\omega}_k \boldsymbol{\gamma}_k h(\boldsymbol{x}_k), \quad 0 \leq \gamma(\boldsymbol{x}) < 1, \tag{12}$$

where the relaxing variable ω resolves the trade-off between feasibility and safety and is optimised with other variables inside an optimisation formulation. A NMPC formulation with the CBF integration can be posed as follows,

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{\omega}} \quad & \sum_{k=0}^{N-1} \left[\boldsymbol{q}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \boldsymbol{\psi}\left(\boldsymbol{\omega}_k\right)\right] \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \\ & \boldsymbol{u}_k \in \mathcal{U}, && k = 0, \dots, N-1, \\ & \boldsymbol{x}_k \in \mathcal{X}, && k = 0, \dots, N-1, \\ & h(\boldsymbol{x}_{k+1}) \geq \boldsymbol{\omega}_k \boldsymbol{\gamma}_k h(\boldsymbol{x}_k) \quad \boldsymbol{\omega}_k \geq 0 && k = 0, \dots, N-1 \end{aligned} \tag{13}$$

The proposed construction of CBFs involves formulating a non-convex optimisation problem to ensure safety guarantees and constraint satisfaction. Non-convexity introduces the challenge of multiple local minima, making the efficient discovery of the global solution arduous.

2.3. Distributive predictive control

In a platooning scenario, the communication and computational cost of implementing a centralised MPC grows with the number of vehicles. Thus, it is attractive to produce a distributed scheme of MPC that both enables autonomy of the individual vehicles and improves tractability (Mishra, Wang, Gazzola, & Chowdhary, 2020).

This segment introduces the fundamental concepts of algebraic graph theory, which are pivotal for comprehending the upcoming problem statement. Additionally, it sets the stage for discussing control strategies within a multi-agent framework and provides an overview of the leader-follower multi-agent topology.

2.3.1. Graph theory

Graph theory provides a structured framework for representing and analysing the interactions between the individual vehicles within the platoon.

Consider a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprising a set of n vertices $\mathcal{V} := \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E} = \left\{(i, j) \in \mathcal{V} \times \mathcal{V} \mid j \in \mathcal{N}_i\right\}$ (Bullo, 2022). Each edge $(i, j) \in \mathcal{E}$ signifies a communicative link between vertices i and j. Here, $m = |\mathcal{E}|$ is the number of edges and $\mathcal{N}_i$ denotes the agents in the neighbourhood of agent i that can communicate with i.

The adjacency matrix $\mathbb{A}$ of $\mathcal{G}$ is the $n \times n$ matrix whose elements a_{ij} are given by $a_{ij} = 1$, if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$, otherwise. $\mathbb{A}$ is symmetric for undirected graphs. The degree of vertex i is defined as $d_i = \sum_{j \in \mathcal{N}_i} a_{ij}$. Then the degree matrix is $\Delta = \operatorname{diag}\left(d_1, d_2, \dots, d_n\right)$. The graph Laplacian of $\mathcal{G}$ is $L = \Delta - \mathbb{A}$, this holds particular importance for continuous time systems.

In truck platooning, distributed control proves advantageous over both centralised and decentralised approaches due to its real-time adjustments based on local information. This strategy balances computational complexity from centralised control and potential synchronisation issues from decentralised control, leading to improved efficiency, safety, and adaptability in platoons. Within the distributed scheme, the control inputs may be contingent on the states of individual agents, such as

$$\boldsymbol{u}_i = \boldsymbol{k}\left(\boldsymbol{x}_i, \{\boldsymbol{x}_j, j \in \mathcal{N}_i\}\right) . \tag{14}$$

Here, the index i represents the specific agent, $\mathcal{N}_i$ denotes its set of respective neighbouring agents and $\boldsymbol{k}$ is the controller function.

Within the scope of cooperative network communication, a critical consideration arises concerning choosing an appropriate topology. While inter-vehicle communication has long been explored and utilised in various vehicular applications, the concept of a leader-follower communication topology has emerged as a preferred and promising approach in the context of platooning (Boulu-Reshef, Holt, Rodgers, & Thomas-Hunt, 2020). The rationale behind this preference lies in the inherently less computationally demanding and simpler hierarchical structure it offers, while still achieving consensus within the network. When relying solely on inter-vehicle communication this introduces significant complexities associated with establishing and maintaining direct communication links among all vehicles within a platoon. This requirement leads to a continuous exchange of information between multiple nodes, resulting in heightened network traffic and the potential for latency issues.

2.3.2. CACC

The core of the CACC system is rooted in its control structure, which governs the coordination and interaction between the ego vehicle and the followers within a network, or a platoon. The platoon comprises a leader vehicle and n_f follower vehicles, and communication between adjacent vehicles is facilitated through V2V techniques. When an obstacle appears in front of the ego vehicle within the same lane, maintaining an appropriate distance and a slower speed than the platoon, the CACC system orchestrates the speed reduction of all platoon members and assigns new cruising speeds. This control strategy, outlined in more detail in Ma, Chu, Guo, Wang, and Guo (2020), ensures seamless platoon integration while epitomising traffic flow.

3. Method

3.1. Problem statement

Platooning refers to a group of vehicles that move in close proximity, coordinated by an ego vehicle. The single ego vehicle serves as the leader of the formation while the followers align closely behind it. A highway-like environment is to be considered, defined by a set of L constitutional lanes denoted as

$$\mathcal{E} = \left\{l_0, l_1, \dots, l_{L-1}\right\} , \tag{15}$$

assuming a left-hand driving configuration. The primary constraints in this environment are the driving direction and lane-specific speed limits, which apply to all vehicles within the platoon. All vehicles considered are modelled as trucks using the general n-trailer model described in (1). The dynamics of the ego agent are then given by

$$\dot{\boldsymbol{x}}_{\text{ego}} = \boldsymbol{f}_{\text{ego}}(\boldsymbol{x}_{\text{ego}}, \boldsymbol{u}_{\text{ego}}) , \tag{16}$$

while the dynamics of the follower by

$$\dot{\boldsymbol{x}}_{\text{follower}} = \boldsymbol{f}_{\text{follower}}(\boldsymbol{x}_{\text{follower}}, \boldsymbol{u}_{\text{follower}}) . \tag{17}$$

In this setup, only the ego vehicle is equipped with a sensor, specifically a LiDAR. To achieve a simulation that closely mirrors reality, we must not assume that the entire environment is constantly known. Instead, we should account for the limited range of perception for the sensor. This necessitates the introduction of the following concept:

Definition 1 (*Field of View (FoV)*). $\text{FoV}_k \subset \mathbb{R}^2$ is defined as the set of points at time step k which are within direct line of sight from the sensor resolution, restricted to the ego vehicle's current lane l_i. The index i auxiliaries the lane designation present in the road segment considered, ranging from the leftmost to the rightmost lane.

This research is centred on the development of a NMPC strategy integrated within a CACC framework capable of effectively controlling all the agents in the network, and the platoon while ensuring a safe and feasible trajectory at all time steps. An emphasis also lies on real-time computation and decision-making capabilities, with a specific aim to render the strategy computationally manageable when compared to state-of-the-art solutions.

3.2. Ego vehicle

To enable the transition from an Optimal Control Problem to a Non-Linear Programming formulation, a discrete model is required, thus, the continuous-time dynamics are discretised through the explicit Runge–Kutta 4th Order method, yielding:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) . \tag{18}$$

Note that here the ego subscripts were omitted to simplify notation. Hence, $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$, denoting the ego state, and $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^m$ the control inputs. Set $\mathcal{X}$ bounds the state according to the vehicle limitations, $\mathcal{U}$ is a compact set and $\boldsymbol{f}$ encapsulates the system behaviour.

3.2.1. Environment interpretation

Recall the highway environment described in (15), thus its associated constraints, here denoted as C, can be formally defined as follows

$$C = \left\{v_0 \leq v_{\text{limit}} \mid l_i \in \mathcal{E}, \text{constraint on lane } l_i\right\} . \tag{19}$$

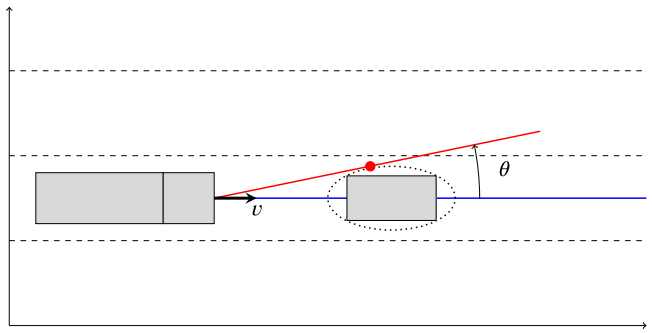


Fig. 2. Scheme depicting the modelling of an obstacle in a simulated highway setting. The ego vehicle is on the left, the obstacle is on the right bounded by an ellipsoid. The red dot indicates the leftmost boundary.

Note that v_0 represents the longitudinal velocity of the vehicle. Thus, the basic NMPC controller for the ego vehicle can be formulated as

$$\begin{aligned} \min_{\boldsymbol{u}(\cdot)} \quad & \sum_{k=0}^{N-1} \boldsymbol{q}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \\ & \boldsymbol{u}_k \in \mathcal{U}, \qquad k = 0, \ldots, N-1, \\ & \boldsymbol{x}_k \in \mathcal{X} \cap C, \qquad k = 0, \ldots, N-1, \end{aligned} \tag{20}$$

where $\boldsymbol{q}$ depicts the reference tracking penalty. i.e the stage cost such that

$$\begin{aligned} \boldsymbol{q}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) = & (\boldsymbol{x}_k - \boldsymbol{r}_k)^\top \mathbf{Q} (\boldsymbol{x}_k - \boldsymbol{r}_k) \\ & + (\boldsymbol{u}_k - \boldsymbol{u}_k^{\text{ref}})^\top \mathbf{R} (\boldsymbol{u}_k - \boldsymbol{u}_k^{\text{ref}}) \, . \end{aligned} \tag{21}$$

That is when assuming a highway-like environment, the constraints are the driving direction, $v \geq 0$, and the speed limit in the lane, v_{limit}. Taking into account the onboard used, that is the LiDAR, the FoV can be redefined as a subset of points within a certain range and angular span (Yin, Yang, & He, 2016; Yuan, Liu, Hong, & Zhang, 2021). Let the sensor's position be denoted by $(x_{\text{LiDAR}}, y_{\text{LiDAR}})$ in the 2D plane, and let $\theta_{\min}$ and $\theta_{\max}$ represent the minimum and maximum angles of the FoV relative to the LiDAR's orientation. Thus, the FoV at a time instant k can be mathematically defined as the set of points (x, y) such that:

$$\begin{aligned} \text{FoV}_k = \{(x, y) \in \mathbb{R}^2 \mid & \\ & \sqrt{(x - x_{\text{LiDAR}})^2 + (y - y_{\text{LiDAR}})^2} \leq R_{\max}, \\ & \theta_{\min} \leq \theta \leq \theta_{\max} \, \}. \end{aligned} \tag{22}$$

It is assumed that the sensor is perfectly coupled and aligned within the ego vehicle. Note that the FoV does not necessarily coincide with the predefined horizon of the NMPC. Meaning, that information outside the NMPC's horizon may not be used immediately for control decisions, even if the sensor can perceive it, which can affect system performance and safety.

3.2.2. Obstacle detection and geometric modelling

Due to their convex nature, bounded ellipsoids are well-suited for representing obstacle geometry, ensuring efficient and reliable trajectory planning. These ellipsoids are determined by reference points obtained from the sensor and are created using Khachiyan's algorithm for Minimum Volume Enclosing Ellipsoids (MVEE), D be an ellipse obtained from the algorithm with centre $\boldsymbol{c}$ and shape matrix $\mathbf{A}$. The leftmost boundary of the ellipsoidal obstacles is used as a reference point for these manoeuvres, especially for predicting a leftward overtaking strategy during obstacle avoidance. The determination of the leftmost boundary point, denoted as $\boldsymbol{b}$, is made concerning a reference frame aligned and centred on the orientation of the ego vehicle, as illustrated in Fig. 2. This process of retrieving $\boldsymbol{b}$ is explained in the Algorithm 1.

Algorithm 1 Retrieve Leftmost Boundary Point of an Ellipse

Initialise $\boldsymbol{b}_0 \leftarrow [-\infty, 0]$
Initialise angle range $[\theta_{\min}, \theta_{\max}] \leftarrow [0, \frac{\pi}{2}]$
while $\theta_{\max} - \theta_{\min} > \delta$ **do**
  Compute midpoint angle $\theta_m \leftarrow \dfrac{\theta_{\min} + \theta_{\max}}{2}$
  Construct line $\mathcal{L}$ with equation $y = \tan(\theta_m)x$
  Compute intersection points between $\mathcal{L}$ and ellipse D
  if there is exactly one boundary intersection point $\boldsymbol{p}$ **then**
    Update $\boldsymbol{b}_i \leftarrow \boldsymbol{p}$
    break
  else
    if $\boldsymbol{p}$ has 2 solutions **then** $\theta_{\min} = \theta_m$
    if $\boldsymbol{p}$ has 0 solutions **then** $\theta_{\max} = \theta_m$
    Update angle range $[\theta_{\min}, \theta_{\max}]$
Output final leftmost boundary point $\boldsymbol{b} \leftarrow \boldsymbol{b}_{\text{final}}$

3.2.3. Integration with motion planning

Here we propose a dynamic obstacle avoidance penalty method based on an NMPC problem, drawing inspiration from Artificial Potential Fields (Sheng & Wang, 2022). The integration involves incorporating an additional optimisation problem that is activated only when an obstacle is detected within the FoV of the ego vehicle, i.e.

$$\text{FoV}_k \cap \mathcal{O} \neq \varnothing \, , \tag{23}$$

where $\mathcal{O}$ represents the set of detected obstacles. Therefore, the NMPC framework is combined with the previously proposed optimisation problem in (16), allowing for adaptive obstacle avoidance during motion planning. This is summarised in Algorithm 2.

Algorithm 2 Dynamic Obstacle Avoidance with NMPC

Input: Environment $\mathcal{E}$
Output: Optimized trajectory for motion planning
Run NMPC problem stated on Eq. (16)
if $\text{FoV}_k \cap \mathcal{O} \neq \varnothing$ **then**
  Incorporate obstacle avoidance penalty terms into the NMPC objective ▷ According to Eq. (25)
Optimise the NMPC problem to obtain the trajectory for motion planning
Return Optimised trajectory for motion planning

Accordingly, to enforce the obstacle constraints to an acceptable predefined tolerance, we employ a penalty method, as shown in Algorithm 3.

Algorithm 3 Dynamic Obstacle Penalty

Input: Current time step k, ego vehicle state $\boldsymbol{x}_k^{\text{ego}}$, FoV_k, detected obstacle $\mathcal{O}_i$
Output: Cost function with an adequate penalty
initiate cost function $\mathbf{J} = 0$
for $i = 0$ to N **do**
  Predict obstacle position via KF
  Compute its leftmost boundary $\boldsymbol{b}_i$ ▷ According to Algorithm 3
  if available left lane for overtake in $\mathcal{E}$ **then**
    Define inner product $\left\langle (\boldsymbol{x}_0 - \boldsymbol{x}_k), (\boldsymbol{b}_i - \boldsymbol{x}_k) \right\rangle$
    Sum the exponential weight to cost function

The proposed reformulated NMPC problem is thus posed as follows

$$\begin{aligned} \min_{\boldsymbol{u}(\cdot)} \quad & \sum_{k=0}^{N-1} \left[q\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \boldsymbol{r}\left(\boldsymbol{x}_k, \mathcal{O}_i\right) \right] \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \\ & \boldsymbol{u}_k \in \mathcal{U}, \qquad k = 0, \dots, N-1, \\ & \boldsymbol{x}_k \in \mathcal{X} \cap C, \qquad k = 0, \dots, N-1 \end{aligned} \tag{24}$$

where q depicts the aforementioned reference tracking penalty and the term $\boldsymbol{r}$ conveys the exponential weight penalty

$$\boldsymbol{r}\left(\boldsymbol{x}_k, \mathcal{O}_i\right) = \beta^k \left\langle \left(\boldsymbol{x}_0 - \boldsymbol{x}_k\right), \left(\boldsymbol{b}_i - \boldsymbol{x}_k\right) \right\rangle \tag{25}$$

This exponential penalty forces the trajectory to change locally where the obstacle is predicted to be with rate β accounting for the growth of the penalty.

3.3. Follower vehicles

The primary objective for each follower vehicle is to execute reference tracking in line with the platoon's trajectory, resembling the formulation present initially in (16). However, an additional safety constraint comes into play: ensuring a safe distance is maintained from the preceding vehicle. To incorporate this safety consideration, the optimisation problem can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{u}(\cdot)} \quad & \sum_{k=0}^{N-1} q\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \\ & \boldsymbol{u}_k \in \mathcal{U}, \quad k = 0, \dots, N-1, \\ & \boldsymbol{x}_k \in \mathcal{X} \cap C, \quad k = 0, \dots, N-1, \\ & (x_0 - x_{\text{preceder}})^2 + (y_0 - y_{\text{preceder}})^2 \geq d^2, \\ & \quad k = 0, \dots, N-1. \end{aligned} \tag{26}$$

The cost function is formulated as in (21). The reference trajectory $\boldsymbol{r}$ for each follower vehicle should be derived from the ego vehicle's current position to maintain a cohesive and synchronised platoon trajectory. Additionally, the control inputs $\boldsymbol{u}^{\text{ref}}$ are derived from the ego's optimised steering and throttle signals.

3.4. CACC for the proposed distributed controller

In our proposal, the CACC constitutes the cornerstone of the platooning system, orchestrating the interaction between the ego vehicle and follower vehicles. We model the network interactions through the adjacency matrix, denoted as $\mathbb{A}$, which captures the connectivity between vehicles:

$$\mathbb{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{27}$$

Let the number of followers in the platoon be n_f, the degree matrix quantifies the number of connections each vehicle has

$$\Delta = \begin{bmatrix} n_f & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{28}$$

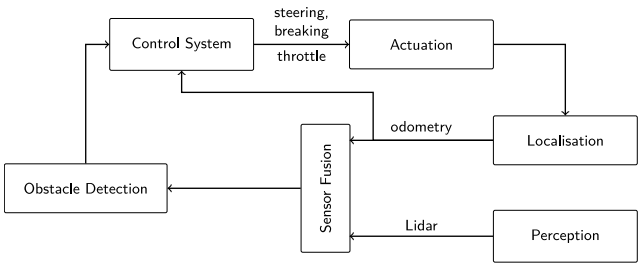


Fig. 3. Holistic view of system architecture for the ego vehicle. Illustrating the interplay between control system, actuation, localisation, perception, sensor fusion, and obstacle detection components.

Thus the Laplacian matrix is such that

$$L = \begin{bmatrix} n_f & -1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \vdots & 0 \end{bmatrix}. \tag{29}$$

We remark to the reader that the definitions presented in this subsection are given just for clarity of the communication topology. Since we are not proposing a centralised controller, in the implementation, each vehicle is computing the control law based solely on their state and the sensor readings and received data from the ego vehicle and these matrices are not used as they represent the entire network.

3.5. Implementation

This section delves into the practical implementation of the proposed algorithm, achieved through the integration of ROS with Sharifi, Chen, Pretty, Clucas, and Cabon-Lunel (2018). An overview of the ego vehicle's system architecture is presented in Fig. 3.

Here, the perception module supplies the control system with a precise description of the environment, covering moving and static objects. This module is primarily based on a LiDAR sensor, such as the Hokuyo UTM-30LX 2D laser scanner (Hokuyo Automatic, 2024). The sensor fusion module combines data from multiple sources, such as odometry and LiDAR, to estimate obstacle positions accurately. Moreover, it employs a Kalman Filter to estimate the obstacle's velocity with the NMPC horizon. The control module is built upon the NMPC formulation introduced earlier, in (20) for the ego and (26) for the follower vehicles. To find a locally optimal solution to the OCP, the state-of-the-art numerical optimal control software CasADi with ipopt solver (Andersson, Gillis, Horn, Rawlings, & Diehl, 2019) is employed. The software package allows for reformulating the continuous-time optimal control problem into an NLP problem.

The distributed control scheme implemented in this research is facilitated through ROS. One key component is the set of topics, denoted as $\mathcal{T}$, which serves as the communication channels for vehicles within the network. These topics encompass the main entities involved, including $\mathcal{T}$ ego for the ego vehicle and $\mathcal{T}$ follower $1, \dots, \mathcal{T}_{\text{follower } n_f}$ for the followers. The system is structured around specific object classes, captured in the set C, which includes C_{odom} for odometry messages, C_{control} for control-related data, and C_{sensor} for sensor information. To manage the roles and interactions within the network, the system employs labels assigned to each vertex, which can be collectively represented as $\mathcal{X}$. These labels correspond to distinct vehicle roles, with X_0 designating the ego vehicle, and $X_1, X_2, \dots, X_{n_f}$ representing the labels for the follower vehicles, sequentially. This structured approach allows us to envision the communication and coordination in a ROS-based system as a directed graph $\mathcal{G}$, where nodes and edges represent vehicles and their interactions, respectively.

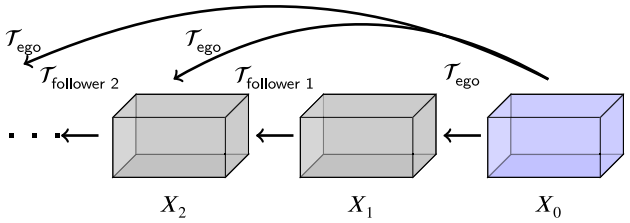


Fig. 4. A visual representation of the information flow in the platoon dynamics within ROS. The ego vehicle is represented in *blue* and the different followers in *light grey*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 3 visually captures the essence of this communication scheme, drawing inspiration from the CACC framework, and highlighting the labels and topics integral to the platoon's coordinated operation (see Fig. 4).

4. Validation

The proposed method is thoroughly assessed to validate its performance across a range of scenarios, such as lane-changing manoeuvres, obstacle avoidance and overtaking, and abrupt braking of the ego vehicle.[2]

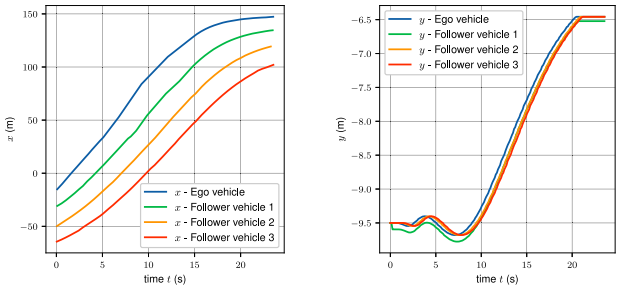
Prior, a minimal parameter tuning to optimise the performance of the proposed algorithm was conducted. The NMPC weight matrices were such that

$$\mathbf{Q} = \text{diag}(100, 150, 10, 5, 10, \underbrace{5, 5, \dots, 5}_{n\text{-times}}) \tag{30}$$

where n represents the number of trailers in the controlled vehicle model, both for the ego and followers. Additionally, the control input penalty matrix was defined as a fixed-size matrix $\mathbf{R} = \text{diag}(5, 5)$. The distance among consecutive followers or the ego vehicle was defined based on time by setting a distance reference such that a second is elapsed between the two vehicles crossing the same position. On the other hand, velocity reference is not specified and the MPC solver can select an appropriate value provided is below the legal limit of 90 km/h and the vehicle is moving towards the reference point along the lane and the inter-distance is maintained. We assumed a maximum of 1.6 ft/s^2 or 0.4877 m/s^2 as acceleration as provided in Yang, Xu, Wang, and Tian (2016). For the maximum turning rate, there are quite a variety of values depending on the manufacturer and we selected 100°/s or 1.7453 rad/s based on values often found in models for Gazebo. Furthermore, the exponential weight penalty for the soft constraint in (25) was set to $\beta = 3.7$. The prediction horizon was fixed at $N = 10$, and the controller operates at a frequency of 10 Hz. Additionally, all topics and services in the ROS system are broadcasted at uniform time intervals to ensure standardisation. All the forthcoming experiments were conducted using trucks of comparable shapes, each featuring a single trailer in a simple tractor-trailer configuration. The tractor segment has a length of $l_0 = 3.5$ m, with a hitching offset denoted as of $m_0 = 0.5$ m. The trailer's length is $l_1 = 6$ m. The LiDAR's operating range was empirically set to match the road width. Similarly, the KF parameters were tuned empirically.

In this study, the computational resources utilised include a MacBook Pro powered by an Apple M1 Pro chip with 16 GB of RAM. All experiments and simulations were conducted within a virtual machine environment, facilitated by Parallels Desktop, using Ubuntu 20.04 and ROS Noetic.

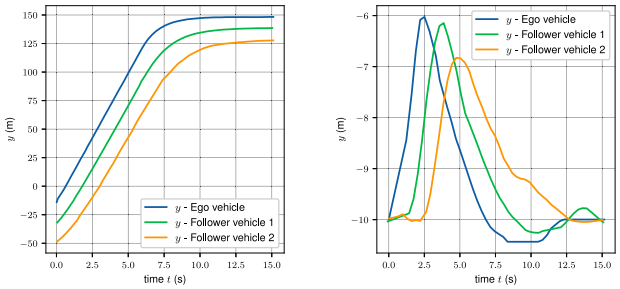
[2] The codebase and resources associated with the work presented are available in the following repository: https://github.com/blourenco217/Master-Thesis-Simulation.



(a) Evolution of the longitudinal distance of the platoon.

(b) Evolution of the lateral displacement in the road.

Fig. 5. Evolution of the position for a 4-truck platoon when changing lanes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Longitudinal position for an overtake maneuver of a static obstacle.

(b) Lateral position for an overtake maneuver of a static obstacle.

Fig. 6. Position tracking during static obstacle avoidance. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Foremost, basic lane-changing manoeuvres were executed with different platoon configurations. In Fig. 5 we present the results for a 4-truck platoon and present the evolution of the x and y coordinate for the ego vehicle and the 3 followers respectively in Fig. 5(a) and Fig. 5(b).

The x position demonstrates a synchronised behaviour, indicating that the vehicles maintain a constant inter-vehicle distance throughout the manoeuvre. Meanwhile, the y position reveals a small ripple effect originating when the ego vehicle initiates the maneuver, which is then propagated to the other vehicles as they receive the information.

Shifting the focus to the overtaking scenarios, Fig. 6 presents the evolution of the x and y coordinates over time for an overtake maneuver of another vehicle that is stationary in the road. Here, the algorithm's effectiveness in manoeuvring around the obstacle corresponds to the noticeable *jump* in the trajectory at the start of the maneuver (that was clocked to be at $t = 0$) corresponds to the activation of the obstacle avoidance component of the MPC.

In both scenarios depicted in Figs. 7 and 8, the ego vehicle, equipped with a LiDAR, leads a platoon of two following vehicles, while an obstacle in the form of a *light-grey* cube is on the rightmost lane of the road. In both instances, the obstacle is successfully avoided.

The obstacle avoidance performance of the system was compared with a baseline model, as introduced in (13). This comparison enables a robust evaluation of the system's capabilities. Both approaches successfully execute overtaking without colliding with the obstacle.

Figs. 9 and 10 provide a side-by-side comparison of the input signals acceleration and turning rate for the proposed method in comparison with the direct application of hard constraints in MPC. In the novel

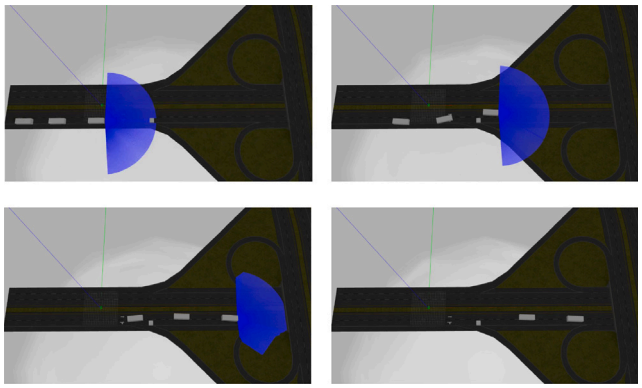


Fig. 7. Snapshots from a 13-s video clip showcasing a platoon of vehicles navigating past a static obstacle.

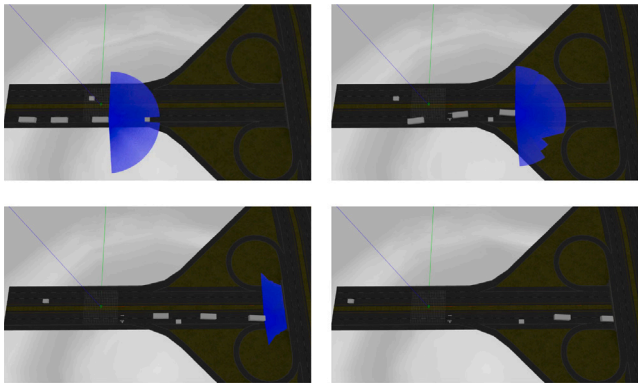
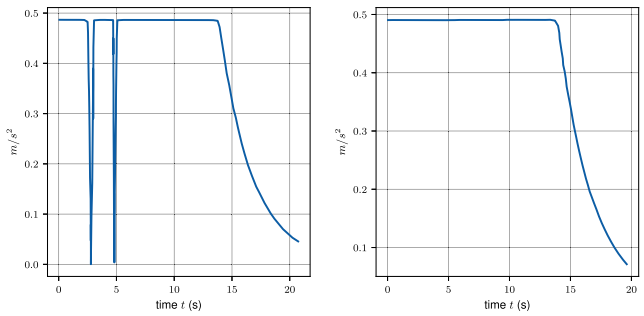
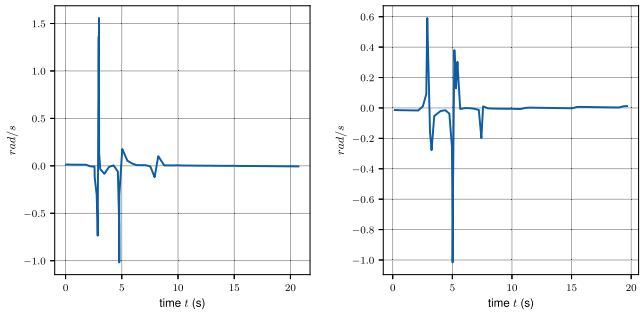


Fig. 8. Snapshot from a 16-s video clip showcasing a platoon of vehicles navigating past a dynamic obstacle.



(a) Baseline method. (b) Novel method.

Fig. 9. Comparative analysis of the acceleration for the baseline vs. the novel proposed method.



(a) Baseline method. (b) Novel method.

Fig. 10. Comparative analysis of the turning rate for the baseline vs. the novel proposed method.

method, acceleration exhibits a linear behaviour at the beginning of the ego vehicle's movement. Conversely, in the baseline approach, acceleration shows two steep valleys. These valleys are a direct result of the CBF method that initially breaks before attempting to overtake. We remark from the results that the selected cost function for the MPC resulted in the vehicle attempting to speed up to decrease the positioning error and saturating given the bound for the acceleration. On the other hand, when assessing the steering rate, both experiments showcase similarities although the proposed method has a smoother profile. This similarity suggests that the steering rate profiles in both methods are relatively consistent albeit the peaks associated with the maneuvers are less pronounced in the proposed method due to the proposed soft constraint with an exponential weight. Other choices for the cost function can be employed since one might want to smooth the turning rate by adding a state that tracks the difference of actuation for two consecutive time steps and adding a quadratic term to penalize large deviations of the turning rate.

5. Conclusion

Platoon-based transportation systems offer a promising solution to revolutionise road transport, with potential benefits in fuel efficiency, reduced emissions, and improved traffic management. However, the successful implementation of platoons hinges on their ability to navigate complex environments safely and efficiently. In this paper, we addressed these challenges by introducing a novel approach to real-time obstacle avoidance in platoon control systems. Our approach leverages NMPC to enable platoons to respond dynamically to obstacles. It integrates seamlessly with existing platoon control systems and adapts to rapidly changing road conditions. Safety is a top priority, with penalty soft constraints ensuring vehicles maintain safe distances from obstacles while preserving platoon coherence. We demonstrated the effectiveness of our approach in preventing collisions through real-time simulations, optimising trajectory planning, and ensuring minimal disruption to traffic flow.

In extensive experiments replicating real-world scenarios, our approach outperformed one established baseline model in terms of safety, efficiency, and real-time decision-making. We found that it enhances safety, improves efficiency, and offers real-time adaptability, making platooning a viable solution for modern transportation needs. Additionally, our approach excels in providing smoother velocity profiles in overtaking scenarios.

Despite our significant progress, future research should focus on scalability, integration with autonomous vehicles, and real-world testing. This includes exploring cost-coupled optimisation problems, 3D LiDAR integration, advanced communication protocols, fine-tuning MPC parameters, and GPU integration for real-time control.

CRediT authorship contribution statement

Beatriz Lourenço: Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Daniel Silvestre:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Daniel Silvestre reports financial support was provided by Foundation

for Science and Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alam, A., Gattami, A., Johansson, K. H., & Tomlin, C. J. (2014). Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations. *Control Engineering Practice*, *24*, 33–41. http://dx.doi.org/10.1016/j.conengprac.2013.11.003.

Allgöwer, F., Findeisen, R., & Nagy, Z. K. (2004). Nonlinear model predictive control: From theory to application. *Journal of the Chinese Institute of Chemical Engineers*, *35*, 299–315. http://dx.doi.org/10.6967/JCICE.200405.0299.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*, 1–36. http://dx.doi.org/10.1007/s12532-018-0139-4.

Boulu-Reshef, B., Holt, C. A., Rodgers, M. S., & Thomas-Hunt, M. C. (2020). The impact of leader communication on free-riding: An incentivized experiment with empowering and directive styles. *The Leadership Quarterly*, *31*, Article 101351. http://dx.doi.org/10.1016/j.leaqua.2019.101351.

Bullo, F. (2022). *Lectures on network systems* (1.6 ed.). Kindle Direct Publishing, URL https://fbullo.github.io/lns.

Challenge, E. T. P. (2016). European Truck Platooning Challenge 2016. URL https://www.government.nl/binaries/government/documenten/leaflets/2015/10/06/leaflet-european-truck-platooning-challenge-2016/brochure-european-truck-platooning-challenge-2016.pdf.

Chen, X., Tang, Z., Johansson, K. H., & Mårtensson, J. (2024). Safe platooning and merging control using constructive barrier feedback. *European Journal of Control*, Article 101060. http://dx.doi.org/10.1016/j.ejcon.2024.101060.

Hernandez, A., Desideri, A., Ionescu, C., Keyser, R. De., Lemort, V., & Quoilin, S. (2016). Real-time optimization of organic rankine cycle systems by extremum-seeking control. *Energies*, *9*(334), http://dx.doi.org/10.3390/en9050334.

Hokuyo Automatic (2024). UTM-30lx | products list | scanning rangefinder | distance data output | UTM-30lx.

Ji-Hyland, C., & Allen, D. (2020). What do professional drivers think about their profession? An examination of factors contributing to the driver shortage. *International Journal of Logistics Research and Applications*, *25*, 231–246. http://dx.doi.org/10.1080/13675567.2020.1821623.

Johansson, A., Nekouei, E., Johansson, K. H., & Mårtensson, J. (2023). *Platoon coordination in large-scale networks: a game theoretic approach* (pp. 79–100). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-43448-8_5.

Lesch, V., Breitbach, M., Segata, M., Becker, C., Kounev, S., & Krupitzer, C. (2022). An overview on approaches for coordination of platoons. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 10049–10065. http://dx.doi.org/10.1109/TITS.2021.3115908.

Ljungqvist, O. (2020). *Linköping studies in science and technology. dissertations*: *vol. 2070*, *Motion planning and feedback control techniques with applications to long tractor-trailer vehicles*. Linköping: Linköping University Electronic Press, http://dx.doi.org/10.3384/diss.diva-165246.

Lukassek, M., Völz, A., Szabo, T., & Graichen, K. (2021). Model Predictive Path-Following Control for General n-Trailer Systems with an Arbitrary Guidance Point. In *European control conference* (pp. 1335–1340). http://dx.doi.org/10.23919/ECC54610.2021.9654870.

Ma, H., Chu, L., Guo, J., Wang, J., & Guo, C. (2020). Cooperative adaptive cruise control strategy optimization for electric vehicles based on SA-PSO with model predictive control. *IEEE Access*, *8*, 225745–225756. http://dx.doi.org/10.1109/ACCESS.2020.3043370.

Maiti, S., Winter, S., Kulik, L., & Sarkar, S. (2020). The Impact of Flexible Platoon Formation Operations. *IEEE Transactions on Intelligent Vehicles*, *5*, 229–239. http://dx.doi.org/10.1109/TIV.2019.2955898.

Mishra, P. K., Wang, T., Gazzola, M., & Chowdhary, G. (2020). Centralized model predictive control with distributed adaptation. In *59th IEEE conference on decision and control* (pp. 697–703). [ISSN: 2576-2370] http://dx.doi.org/10.1109/CDC42340.2020.9304097.

Moradi, B. (2022). *A non-linear MPC local planner for tractor-trailer vehicles in forward and backward maneuvering* Master's thesis, Canada: University of Regina.

Porfiri, M., Roberson, D., & Stilwell, D. (2006). Environmental tracking and formation control of a platoon of autonomous vehicles subject to limited communication. In *Proceedings of IEEE international conference on robotics and automation* (pp. 595–600). [ISSN: 1050-4729] http://dx.doi.org/10.1109/ROBOT.2006.1641775.

Sharifi, M., Chen, X., Pretty, C., Clucas, D., & Cabon-Lunel, E. (2018). Modelling and simulation of a non-holonomic omnidirectional mobile robot for offline programming and system performance analysis. *Simulation Modelling Practice and Theory*, *87*, 155–169. http://dx.doi.org/10.1016/j.simpat.2018.06.005.

Sheng, Z., & Wang, S. (2022). Obstacle avoidance of mobile robots using modified artificial potential field algorithm based on vortex and PID adjustment. In *International conference on advanced sensing and smart manufacturing* (pp. 318–333). SPIE, http://dx.doi.org/10.1117/12.2652491.

Sidorenko, G., Thunberg, J., Sjöberg, K., Fedorov, A., & Vinel, A. (2022). Safety of automatic emergency braking in platooning. *IEEE Transactions on Vehicular Technology*, *71*, 2319–2332. http://dx.doi.org/10.1109/TVT.2021.3138939.

Silvestre, D. (2022a). Constrained convex generators: A tool suitable for set-based estimation with range and bearing measurements. *IEEE Control Systems Letters*, *6*, 1610–1615. http://dx.doi.org/10.1109/LCSYS.2021.3129729.

Silvestre, D. (2022b). Set-valued estimators for uncertain linear parameter-varying systems. *Systems & Control Letters*, *166*, Article 105311. http://dx.doi.org/10.1016/j.sysconle.2022.105311.

Silvestre, D., & Ramos, G. (2023). Model predictive control with collision avoidance for unknown environment. *IEEE Control Systems Letters*, *7*, 2821–2826. http://dx.doi.org/10.1109/LCSYS.2023.3288884.

Taborda, P., Matias, H., Silvestre, D., & Lourenço, P. (2024). Convex MPC and thrust allocation with deadband for spacecraft rendezvous. *IEEE Control Systems Letters*, *8*, 1132–1137. http://dx.doi.org/10.1109/LCSYS.2024.3407611.

Thirugnanam, A., Zeng, J., & Sreenath, K. (2022). Safety-Critical Control and Planning for Obstacle Avoidance between Polytopes with Control Barrier Functions. In *International conference on robotics and automation* (pp. 286–292). http://dx.doi.org/10.1109/ICRA46639.2022.9812334.

Thormann, S., Schirrer, A., & Jakubek, S. (2022). Safe and efficient cooperative platooning. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 1368–1380. http://dx.doi.org/10.1109/TITS.2020.3024950.

Won, M. (2022). L-Platooning: A Protocol for Managing a Long Platoon With DSRC. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 5777–5790. http://dx.doi.org/10.1109/TITS.2021.3057956.

Yang, G., Xu, H., Wang, Z., & Tian, Z. (2016). Truck acceleration behavior study and acceleration lane length recommendations for metered on-ramps. *International Journal of Transportation Science and Technology*, *5*, 93–102. http://dx.doi.org/10.1016/j.ijtst.2016.09.006.

Yin, H., Yang, X., & He, C. (2016). Spherical Coordinates Based Methods of Ground Extraction and Objects Segmentation Using 3-D LiDAR Sensor. *IEEE Intelligent Transportation Systems Magazine*, *8*, 61–68. http://dx.doi.org/10.1109/MITS.2015.2494079.

Yuan, C., Liu, X., Hong, X., & Zhang, F. (2021). Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments. *IEEE Robotics and Automation Letters*, *6*, 7517–7524. http://dx.doi.org/10.1109/LRA.2021.3098923.

Zeng, J., Li, Z., & Sreenath, K. (2021). Enhancing Feasibility and Safety of Nonlinear Model Predictive Control with Discrete-Time Control Barrier Functions. In *60th IEEE conference on decision and control* (pp. 6137–6144). [ISSN: 2576-2370] http://dx.doi.org/10.1109/CDC45484.2021.9683174.

Zhou, J., Tian, D., Sheng, Z., Duan, X., Qu, G., Zhao, D., et al. (2022). Robust min–max model predictive vehicle platooning with causal disturbance feedback. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 15878–15897. http://dx.doi.org/10.1109/TITS.2022.3146149.