Model Predictive Path-Following Framework for Generalized N-Trailer Vehicles in the Presence of Dynamic Obstacles Modeled as Soft Constraints

by Deniz, N.N. and Auat Cheein, F.

Deniz, N.N. and Auat Cheein, F. (2025) 'Model Predictive Path-Following Framework for Generalized N-Trailer Vehicles in the Presence of Dynamic Obstacles Modeled as Soft Constraints', *IEEE Transactions on Automation Science and Engineering,* 22, pp. 7018–7032. Available at: https://doi.org/10.1109/TASE.2024.3458809.

Model predictive path-following framework for generalised N-trailer vehicles in the presence of dynamic obstacles modelled as soft constraints

Nestor Deniz[1] and Fernando Auat Cheein, *IEEE Senior Member*[1,2]

[1]Department of Electronic Engineering. Universidad Tecnica Federico Santa Maria. Av España 1680, Valparaiso, Chile

[2]UK National Robotarium, Edinburgh Centre for Robotics, School of Engineering and Physical Sciences, Heriot-Watt University, EH14 4AS, Edinburgh, UK

***Abstract*—Collision avoidance is crucial for autonomous navigation systems. Many studies have addressed obstacle avoidance for single unicycles and car-like vehicles in on-road conditions. In this work, we extend the scope to generalised N-trailer vehicles, comprising a single active segment pulling multiple trailers. Unlike approaches that treat obstacles as hard constraints, we model them as soft constraints using Gaussian functions. This method maintains the convexity of the search space, reducing computational demands. However, the regions occupied by obstacles remain feasible. Thus, the Gaussian function's amplitudes need to be carefully chosen to discourage navigation through these areas. Moreover, closed-loop stability is guaranteed by generating auxiliary references when the nominal path is occluded. The efficacy of this approach is demonstrated through simulated and field experiments with a tractor pulling two trailers. These experiments show the method's capability to navigate around obstacles efficiently while maintaining computational efficiency, validating its practical applicability. Videos of the experiments and the implemented algorithms are available at https://usmcl-my.sharepoint.com/:f:/g/personal/nestor_deniz_usm_cl/EtU54g1NeslNhD8V7dAeu20B0umnQa4FKiMlzThkTAXYvg?e=swEXwg. Despite the success in real-time implementation, more research is needed to address the open questions discussed at the end of this article.**

***Note to Practitioners*—This work focuses on implementing obstacle avoidance for a kind of vehicles widely used in agriculture, mining, luggage transportation, and industry. A LiDAR Velodyne VLP16, configured with its lowest rotation speed for denser point clouds, is used to scan the environment. Proper attachment of the LiDAR to the tractor's body minimises vibration and azimuth movements, ensuring accurate obstacle detection. Obstacles are modelled as Gaussian functions to maintain the convexity and optimise computational efficiency. The Gaussian function's amplitude should be set high enough to effectively avoid collision when density of obstacle is high. The framework uses a control horizon N_c and a prediction horizon N_p beyond the control to anticipate obstacle's position. However, a large prediction horizons N_p is not advised when the the model of the dynamic of the obstacles is not accurate.**

***Index Terms*—N-Trailers vehicles; Obstacle avoidance; Nonlin-**

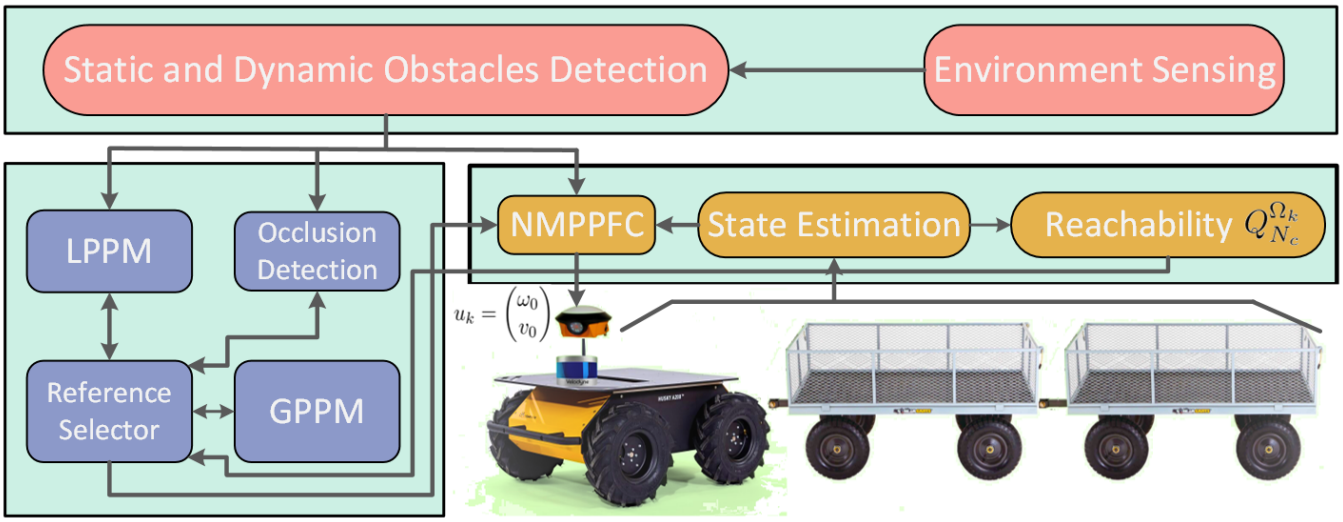

Fig. 1: General scheme representing the proposed framework. The Global Path Planner Module generates a path free of obstacles. The environment is scanned with a LiDAR in search of objects that occlude or may occlude the path in the near future. The Local Path Planner Module generates an auxiliary and reachable reference when the nominal path is occluded.

ear model predictive controller; Efficient motion controller

I. Introduction

N-TRAILER or chained vehicles are a popular choice in transportation due to their versatility, allowing them to transport variable payloads using a single tractor. According to the hitching type, the N-trailer vehicle is classified into Standard N-Trailer (SNT), non-Standard N-Trailer (nSNT), and Generalised N-Trailer (GNT) vehicles [**?**], with GNT being the most flexible configuration. Figure 1 shows a chained vehicle together with the scheme of the proposed framework.

The use of chained vehicles finds multiple applications in diverse environments, such as airports, agricultural, industrial, and mining. These environments are structured environments, and usually, the vehicle has to follow a predefined path or trajectory. Achieving accurate path-following of chained vehicles is challenging, especially when the guided vehicle is the last trailer and the path is highly curved. Compensation for the negative effect of the path curvature during the turning of the trailers by using an observer-based proportional-integral-derivative (PID) controller is proposed in [**?**]. A controller consisting of four control loops is presented in [**?**], where each loop addresses different challenges commonly encountered when controlling N-trailer vehicles, such as tracking errors, model uncertainties, unmeasured states, external disturbances, nonlinear effects and saturation of the actuators.

Accurate control of N-trailer vehicles is a challenging task and it becomes stronger when obstacles are prone to appear during the travel. Therefore, obstacle avoidance is of paramount importance to achieve safe autonomous navigation. The vehicle has to avoid any possible static and dynamic obstacles while it deviates as minimum as possible from the predefined path. Obstacle avoidance for grounded and aerial vehicles, a combination of both, as well as multiple vehicles, have been addressed in the literature. However, the case of chained vehicles with an arbitrary number of passive trailers has received less attention.

To fill this gap, in this work, we introduce a novel Nonlinear Model Predictive Path-Following Controller (NMPPFC) designed for Generalised N-Trailer vehicles operating in structured environments where static and dynamic obstacles are prone to appear. The vehicle's vector state takes into account the coordinates in the plane of every segment within the chain to easily compute the distance to obstacles. The environment is scanned with the help of a LiDAR to find possible obstacles that may occlude the path. This method differentiates itself from others, such as the Artificial Potential Field (APF) method, by incorporating a predictive model that anticipates the future positions of dynamic obstacles and adjusts the vehicle's path accordingly. The NMPPFC leverages optimisation techniques to maintain stability and feasibility while navigating complex environments with both static and dynamic obstacles by properly designing the objective function. Additionally, the use of a Local Path Planner Module aims to avoid being trapped in local minima, a common issue in APF methods in complex environments.

The key contributions of this work can be summarised as follows: i) obstacles are modelled as Gaussian functions that enter the NMPPFC optimisation problem, remaining convex the search space for the optimisation solver, ii) multiple static and dynamic obstacles can be handled within the NMPPFC formulation where future position of dynamic obstacles is predicted, iii) stability and recursive feasibility are analysed for the NMPPFC by a condition that is linked to the vehicle's controllability property, iv) several simulated and field experiments demonstrated the effectiveness of the proposed method and its real-time applicability.

The remainder of this article is structured as follows: Section II reviews related work on the different methods and techniques developed for obstacle avoidance in different scenarios. In Section III, we specify the model of the Generalised N-trailer vehicle and elaborate on the problem of obstacle avoidance for this kind of vehicle. Section IV presents a comparative analysis of our formulation under different scenarios of simulated and field experiments. In Section V, some questions that remain open in this article are discussed. Finally, Section V shows the conclusions of our work and our future research plan regarding obstacle avoidance of GNT vehicles.

II. Related Works

This section reviews previous research on obstacle avoidance of different kinds of vehicles, as well as the methods and strategies used for achieving safe autonomous navigation in different scenarios.

A. Environment sensing

Measuring and estimating the vehicle's state, together with sensing the environment play a key role in detecting and avoiding obstacles effectively. The authors in [**?**] propose an exploration system based on LiDAR for the state estimation of Unmanned Aerial Vehicles (UAV) which is capable of exploring and detecting obstacles and objects. The system couples the vehicle's inertial odometry to the LiDAR, making it capable of adapting to its surroundings. In [**?**], a method for accurately classifying dynamic objects using low-resolution point clouds is presented. The authors use ensemble learning to perform feature-level fusion on multiple networks to exploit their different expression capabilities and by the use of long short-term memory, the dynamic obstacles are classified. Following, in [**?**], the authors address the problem of constructing navigable space from sparse noisy point clouds. The method incrementally seeds and creates local convex regions free of obstacles along the robot's trajectory. Through a map point regulation process, a dense version of the point cloud is reconstructed. In [**?**], an autonomous navigation pipeline using Open Street Map (OSM) information for global planning is presented. To avoid the flaw of local low-accuracy of the OSM, the authors develop a LiDAR-based naive Valley Path (NVP), that exploits the concept of valley areas to infer the local path furthest from obstacles.

B. Handling occlusion

Occlusion presents significant challenges in the field of autonomous navigation, particularly for vehicles operating in off-road conditions. Occlusion occurs when an object in the environment blocks the sensors' line of sight, preventing the detection of other objects or obstacles behind it. This is a well-studied issue in urban and on-road conditions, where occluded objects are typically vehicles, pedestrians, or infrastructure elements [**?**], [**?**]. Various techniques have been developed to address occlusion in these scenarios, including the use of multiple sensor modalities [**?**], predictive modelling [**?**], and advanced algorithms for real-time object detection and tracking [**?**].

However, the problem of occlusion in off-road, agricultural, mining and industrial environments remains less mature. In these settings, occlusion can be caused by natural elements such as vegetation, terrain irregularities, and other static and dynamic obstacles. These environments introduce unique challenges due to the variability and complexity of natural landscapes, which can obscure not only small obstacles but also large machinery or other vehicles. Current research in this area has begun to explore the integration of sensor fusion, machine learning techniques, and predictive algorithms to improve the navigation capabilities of autonomous vehicles in off-road conditions in the presence of occlusion [**?**], [**?**]. Despite these advances, there is still a considerable need for robust solutions that can ensure safe and efficient operation in the diverse and dynamic conditions typical of off-road settings.

C. Strategies for evading obstacles

Among the most common techniques used for navigation and obstacle avoidance, one can find in the literature Artificial Potential Field (APF), fuzzy and machine learning techniques, and, model-based methods. The authors in [**?**] develop a dynamic APF path-planning technique for UAV for following ground-moving targets while avoiding unknown dynamic obstacles. Following, a prediction stage to the APF is introduced in [**?**] to bypass obstacles and to prevent the generation of unsmooth and oscillating motions in Autonomous Ground Vehicles (AGV). Moreover, by adding virtual obstacles which generate repulsive force in critical areas, local minimums can be avoided.

The obstacle avoidance of multiple AGVs is addressed in [**?**]. The authors proposed a unified framework that integrates trajectory planning and motion optimisation based on a spatio-temporal safety corridor, which guarantees collision avoidance and trajectory smoothness. Moreover, path planning and obstacle avoidance for multiple AGVs in changing environments taking into account path length, number of AGVs and time constraints is proposed in [**?**]. The method relies on deformable virtual leader-follower formation to enable AGVs to adapt their configuration based on both planned and real-time data. The use of Voronoi cells is contemplated in [**?**] to achieve collision avoidance of decentralised and communication-free multi-robot formations that accounts for both localisation and sensing uncertainties. The case of distributed leader-follower control problem for networked heterogeneous Unmanned Aerial Vehicle-Unmanned Ground Vehicle (UAV-UGV) is developed in [**?**]. The strategy is developed for unknown environments where is required to keep the vehicle's formation, obstacle avoidance, inter-robot collision avoidance and reliable robot communications.

Other common techniques for obstacle avoidance include machine-learning methods, as in [**?**], where a deep deterministic policy gradient (DDPG) approach to realise path planning and obstacle avoidance is presented. A LiDAR is used to measure the distance to obstacles, while an odometer measures the mileage. This information is then used to train the DDPG method and used with a UGV to reach the goal with fewer steps in comparison with a Deep Q-network algorithm. A Deep Reinforcement Learning (DRL) for the autonomous navigation of UGVs in mapless conditions is presented in [**?**]. The method fuses information from a LiDAR, target position and current UGV's speed to compute the output. The APF and DRL methods are combined in [**?**] to achieve path planning for UAVs tracking grounded targets while avoiding obstacles. The authors formulate a reward function based on line of sight and APF to guide the behaviour of the UAV and achieve the tracking of the target, while a penalty term of action takes the trajectory smoothly.

D. Model predictive in obstacle avoidance

Model-based techniques, such as Model Predictive Controller (MPC) have also been widely used to achieve obstacle avoidance in autonomous navigation in different contexts. Real-time obstacle avoidance and safe navigation of AGVs are reported in [**?**]. It combines selective MPC with APF and particle swarm optimisation, and by defining multiple sets of weight coefficients within the MPC, the method is able to choose between safe or fast paths. A Hybrid MPC is reported in [**?**] to enhance the path-planning of UGVs by combining stochastic dynamic programming (SDP), hybrid MPC and Dijkstra-based pseudo priority queues, where obstacles are avoided by including hard constraints on the MPC formulation.

Obstacle avoidance of small ground vehicles, called Micro Ground Vehicles (MGV) using MPC is addressed in [**?**]. The authors propose trajectory generation for obstacle avoidance by successive convexification considering the shape of the vehicle. Obstacles are evaded by including constraints that are state-triggered. Trajectory planning and tracking for UGV in a dynamic uncertain environment is presented in [**?**]. The artificial fish swarm algorithm is used to connect the start and the destination global trajectory, while the unforeseen obstacles are handled by a trial-based forward search algorithm based on the Markov chain in the local trajectory planner, which is tracked by a multiconstrained MPC (MMPC).

A Nonlinear MPC (NMPC) fault tolerant control scheme for omnidirectional mobile robots is presented in [**?**]. The NMPC is employed to explore the actuation redundancy of the omnidirectional robot, while obstacle avoidance is handled by including constraints on the NMPC formulation. A control

strategy with obstacle avoidance capabilities for a Wing-In-Ground-Effect (WIG) vehicle is presented in [**?**]. The WIG is controlled in different scenarios which include obstacle avoidance, using different control strategies such as NMPC and PID, LQR and Adaptive LQR on a feedback linearisation. Obstacle avoidance and lateral stability of UGVs in high-speed conditions using NMPC are addressed in [**?**]. The visibility graph method is used to plan the global path that avoids collision with static obstacles, while NMPC is used to optimise the path and conduct second path planning that considers both lateral stability and moving obstacles, where their uncertain trajectory are assumed to be described by multivariate Gaussian distribution where a polynomial fitting is utilised to predict the moving trajectories.

An NMPC approach that simultaneously provides an optimising solution for both, path-following and obstacle avoidance tasks in a single optimisation problem is presented in [**?**]. The obstacle avoidance is fulfilled by introducing additional terms in the value functional of the NMPC formulation rather than imposing state constraints. However, only static obstacles are considered. Following MPC strategies, the authors in [**?**] present a steering control MPC-based for obstacle avoidance of UGVs. The method transforms the constraint of the front-wheel-steering angle to lateral acceleration when the path-planning controller reconfigures the path to avoid obstacles. The obstacle avoidance path is generated online by an optimal path reconfiguration based on the direct collocation method. Regarding chained vehicles, in [**?**], an autonomous driving system for a tractor-trailer considering the articulation angle of a semi-trailer truck on a narrow road was presented. Based on MPC, the vehicle's behaviour is forecast and collision detection is detected by using the separate axis theorem, whose results are reflected in the MPC constraints.

Path planning and trajectory tracking of unmanned ground and aerial vehicles, a combination of both, and multiple active ground vehicles with obstacle avoidance using MPC have received attention from the researchers. Moreover, obstacles are usually traduced to constraints within the MPC formulation at expense of losing the convexity of the search space. In addition, controlling vehicles with more than one trailer and no obstacles has also received some attention [**?**], [**?**], [**?**], [**?**]. However, the problem of obstacle avoidance of chained vehicles with only one active segment and an arbitrary number of passive trailers seems to need more attention.

III. Problem formulation

A. Preliminaries and kinematics of N-trailers vehicles

In GNT vehicles, the active segment pulls an arbitrary number N_t of passive trailers connected by passive rotary joints. The system's vector state is $q_t = (\beta_{i,t}, \theta_{0,t}, \theta_{i,t}, x_{0,t}, y_{0,t}, x_{i,t}, y_{i,t})^T$ for $i \in \mathbb{Z}_{[1,N_t]}$, such that $q_t \in \mathcal{Q} \subset \Re^{4N_t+3}$. The joint angle between the segments $i-1$ and i are denoted as $\beta_{i,t}$, $\theta_{0,t}$ is the tractor's attitude, while the attitude of the $i-th$ trailer is denoted as $\theta_{i,t}$, such that $\beta_{i,t} = \theta_{i-1,t} - \theta_{i,t}$, and $\dot{\beta}_{i,t} = \omega_{i-1,t} - \omega_{i,t}$. The coordinates of the tractor in the plane are $(x_{0,t}, y_{0,t})$, whereas $(x_{i,t}, y_{i,t})$ denote the coordinates of the $i-th$ segment. The vector $\mathrm{p}_{i,t} = (x_{i,t},\, y_{i,t},\, \theta_{i,t})^T$, with $i \in \mathbb{Z}_{[0,N_t]}$ is the pose of the $i-th$ segment. In the sequel, it will be useful to define the projection operator that takes the vector state q_t and gives back the pose of the $i-th$ segment, as follows: $\mathrm{proj}_i(q_t) = \mathrm{p}_{i,t}$. With a slight abuse of notation, we will take the projection over the set $\mathcal{Q}$ as $\mathrm{proj}_i(\mathcal{Q}) = (\mathcal{X}_i \times \mathcal{Y}_i \times \Theta_i)$ The control actions are the steering rate $\omega_{0,t}$ and velocity $v_{0,t}$ of the tractor, such that the control vector is $u_t = (\omega_{0,t}, v_{0,t})^T$.

The kinematic model of the chained vehicle is well-known in the literature. It can be expressed succinctly as $\dot{q}_t = G(q_t)u_t$. Readers interested in the model can refer to [**?**] for a detailed explanation. The output r_t is then given by:

$$r_t = h(q_t) = (\beta_{1,t}, ..., \beta_{N_t,t}, \theta_{0,t}, ..., \theta_{N_t,t}, x_{0,t}, y_{0,t}, ..., x_{N_t,t}, y_{N_t,t})^T \tag{1}$$

where $h : \Re^{n_q} \to \Re^{n_r}$, is the output function. Note that not necessarily $n_r = n_q$, i.e., possibly not all states are measured. However, we assume that at least the joint angles and the attitude and coordinates of the tractor are available for measurement. Thus, the system of a GNT vehicle can be described by the following equations:

$$\begin{cases} \dot{q}_t = & G(q_t)\, u_t \\ r_t = & h(q_t) \end{cases} \tag{2}$$

B. Problem statement and motivation

The objective of this work is to design a framework for autonomous generalised N-trailer vehicles to follow a predetermined path while avoiding collisions with static and dynamic obstacles that could appear during the travel. The path-following task entails closely following a reference point that traces a curve, which in this case belongs to the 3-dimensional $x-y-\theta$ space. These references are generated by a Global Path Planner Module (GPPM) and could be generated by using the equation of the path (when available) or in a sampled data fashion.

In this work, obstacles will be modelled as soft constraints that will take part within the objective function of the predictive controller. Specifically, Gaussian functions are employed to represent obstacles due to their continuous and infinitely differentiable nature, posing no issues when solving the optimisation problem. The function's radius, besides the obstacle's size, includes the vehicle's physical dimensions and safety margins ($\mathrm{s_m}$). Special attention is given to choosing the Gaussian amplitude to strongly discourage navigation near obstacles. Since the reference could be occluded and not reachable from the current vehicle's position in the presence of obstacles, a Local Path Planner Module (LPPM) generates auxiliary reachable references for the N-trailer vehicle until the original references of the original path become reachable again.

Variable	Meaning
t	continuous time variable
k	discrete time variable
$\beta_{i,t}$	$i-1$-th joint angle at time t
$\theta_{i,t}$	attitude of the i-th segment
$x_{i,t}$	x coordinate of the i-th segment
$y_{i,t}$	y coordinate of the i-th segment
N_t	number of trailers or passive segments
N_c	length of the control horizon
N_p	length of the prediction horizon
$\omega_{0,t}$	tractor's angular velocity
$v_{0,t}$	tractor's linear velocity
u_t	control vector
q_t	vector of state
r_t	vector of vehicle's output measurements
G	continuous time vehicle's model
$\mathcal{G}$	discrete-time vehicle's model
$\mathrm{p}_{xy\theta}(\gamma_{c,t})$	reference generator function
r_c	reference
$\tilde{\mathrm{r}}_c$	auxiliary reachable reference
$\gamma_{c,t}$	parameter of the reference generator function
σ_t	vel. at which references are generated
$\tilde{\sigma}_t$	vel. at which ref. are gen. during occlusion
$\mathrm{O}_{s,m}$	static obstacle
$\mathrm{O}_{d,m}$	dynamic obstacle
$\Lambda_{m,i}$	function modelling the m-th static obstacle
A_m	amplitude of the m-th static obstacle
ρ_m	radius of the m-th static obstacle
$\Phi_{l,i,t}$	function modelling the l-th dynamic obstacle
A_l	amp. of the m-th dynamic obstacle
ρ_l	radius of the l-th dynamic obstacle
$x_c, y_c, a, b, \varphi_k$	parameters of the elliptical equation
$\mathbb{J}_0(q_k, u)$	nominal cost of the NMPPFC
p_{n-k}	coefficients of the nominal cost
$\mathbb{J}(q_k, u, n)$	cost of the NMPPFC to be minimised
Ω_k	neighbourhood of the reference
$Q_{N_c}^{\Omega_k}$	up to N_c steps controllable set
$\mathcal{O}_s$	total number of static obstacles
$\mathcal{O}_d$	total number of dynamic obstacles
Δ	mean deviation
η	mean computational time
∇	distance to obstacle
Ψ	control effort

TABLE I: Summary of the main variables involved un the proposed framework and their meaning.

C. Global Path Planner Module

Given our primary emphasis on following a predefined path, we will assume that it is a parameterised curve $\mathcal{P}(\gamma_{c,t}) : \Re \to \Re^3$ given by:

$$\mathcal{P}(\gamma_{c,t}) := \quad \left\{ \mathrm{r_c} \in \Re^3 : \mathrm{r_c} = \mathrm{p}_{xy\theta}(\gamma_{c,t}) \right\} \tag{3}$$

where $\mathrm{p}_{xy\theta}(\gamma_{c,t}) : \Re \to \Re^3$, are the reference coordinates and attitude, such that $\mathrm{p}_{xy\theta}(\gamma_{c,t}) = (x(\gamma_{c,t}), y(\gamma_{c,t}), \theta(\gamma_{c,t}))^T$ is generated by the Global Path Planner Module (GPPM). The parameter $\gamma_{c,t}$ is the path parameter, whose dynamics is governed by a virtual input σ_t, such that $\dot{\gamma}_{c,t} = \sigma_t$, with $\sigma_t > 0$, as in [**?**]. Hence, the GPPM assigns a reference $\mathrm{r_c} = \mathbf{p}_{xy\theta}(\gamma_{c,t})$ by integrating $\dot{\gamma}_{c,t} = \sigma_t$. When the line of sight between the guided segment and the target is occluded, the GPPM reduces the value of σ_t to $\tilde{\sigma}_t > 0$. It makes the reference to moving slower along the path, imitating the human driver's behaviour when the path is occluded by pedestrians or other vehicles or objects. In addition, alternative references are generated in these situations by a Local Path Planner Module (LPPM), which will be introduced later.

D. Static obstacles

In order to achieve safe autonomous navigation, the environment need to be scanned in search for obstacles. In this work, we consider an obstacle any object surpassing a predetermined volume that blocks the predefined path of the vehicle. Independently of the obstacle's shape, it will be outer approximated by a circumference of radius r. Then, the $m-th$ static obstacle can be characterised by its coordinates (x_m, y_m) in the plane and its radius r_m, such that $\mathrm{O}_{s,m} = (x_m, y_m, r_m)$. As reviewed in Section II, a common strategy used in the literature to avoid obstacles within an MPC framework is to incorporate the constraint $(x - x_m)^2 + (y - y_m)^2 - r_m^2 > 0$ which causes the loss of the convexity of the search space of the optimisation problem. In this work, detected static obstacles are incorporated into the objective function as an additional stage cost. Specifically, these obstacles enter the objective function as Gaussian functions centred at their coordinates (x_m, y_m). These additional stage cost terms are modelled as follows:

$$\Lambda_{m,i} = \quad A_m e^{\frac{-1}{2\rho_m^2}\left((x_{i,t}-x_m)^2+(y_{i,t}-y_m)^2\right)} \tag{4}$$

where A_m is the amplitude of the $m-th$ Gaussian function modelling the $m-th$ obstacle, ρ_m is the obstacle's radius plus the vehicle's size and a safety margin ($\mathrm{s_m}$), (x_m, y_m) are the coordinates of the $m-th$ obstacle, and, $(x_{i,t}, y_{i,t})$ are the coordinates of the $i-th$ segment of the chained vehicle, with $i \in \mathbb{Z}_{[0,N_t]}$. In this formulation, the point (x_m, y_m) and its vicinity remain feasible zones for the optimisation problem. However, close navigation to this zone is discouraged whenever $\Lambda_{m,i}$ incurs a higher cost when compared to other stage costs ingredients within the prediction window, which will be introduced later. In such cases, an appropriate value for A_m can be determined to ensure that no other stage cost within the prediction window exceeds $\Lambda_{m,i}$.

E. Dynamic obstacles

Dynamics obstacles are all those obstacles whose positions change over time. A key aspect of achieving safe autonomous navigation in the presence of dynamic obstacles is the ability to predict the future position of the moving obstacle. Several strategies have been developed for predicting the trajectory of vehicles and pedestrians in autonomous driving in in-road conditions [**?**]. However, N-trailer vehicles are mainly used in off-road conditions, such as agricultural, mining and industrial environments. In this work, we will use a parametric equation to model all those moving obstacles. Among all possible choices, we will use an elliptical equation to model the dynamic obstacles, since it can describe rectilinear and non rectilinear movements with just a few parameters, avoiding increasing the computational burden excessively. The following Assumption states the validity of the model chosen for the dynamic obstacles.

Assumption 1. All moving obstacles can be described by an elliptical equation, at least for a short period no longer than

$N_p\,T_s$ seconds, where N_p is the prediction horizon length (to be introduced later), and T_s (s) is the sampling time.

The equation describing the movement of any dynamic obstacle is found by minimising the following objective function:

$$\mathcal{J}_d := \frac{1}{2}\sum_{k=1}^{N_o}\left(\frac{(x_k-x_c)^2}{a^2}+\frac{(y_k-y_c)^2}{b^2}-1\right)^2 + (x_k-(x_c+a\cos(\varphi_k)))^2+(y_k-(y_c+b\sin(\varphi_k)))^2 \tag{5}$$

where an amount N_o of the most recent measurements of the position $(x_k,\, y_k)$ of the obstacle are considered, x_c, y_c, a and b are the parameters of the ellipse, and φ_k is the phase of the obstacle within the elliptical curve which allows determining the direction of the movement and future position of the moving obstacles. The parameters and phase are found by minimising Eq. (5) within the following constrained optimisation problem:

$$\begin{aligned} &\min_{a,b,x_c,y_c,\varphi} \quad \mathcal{J}_d \\ &\text{s.t.}\left\{\quad \underline{\Delta}\varphi \le \Delta\varphi \le \overline{\Delta}\varphi\right. \end{aligned} \tag{6}$$

where $\Delta\varphi = \varphi_k - \varphi_{k-1}$, and $\underline{\Delta}\varphi$ and $\overline{\Delta}\varphi$ are the lower and upper bounds, respectively, for the rate of change of the phase, whose value should be fixed as a function of the maximum allowable obstacle's speed. Every time the last measured position of the obstacle is no longer well explained by the current ellipse's equation, formulation given by Eq. (6) is solved again using the last N_o measurements of the position of the obstacle. Then, the dynamic obstacles can be represented as follows:

$$\Phi_{l,i,t} = A_l e^{\frac{-1}{2\rho_l^2}\left((x_{i,t}-x_{l,t})^2+(y_{i,t}-y_{l,t})^2\right)} \tag{7}$$

where $x_{l,t} = x_c + a\cos(\varphi_k + \Delta\varphi\, t)$ and $y_{l,t} = y_c + b\sin(\varphi_k + \Delta\varphi\, t)$ represent the time-varying coordinates of the moving obstacles.

F. Detecting occlusion

Every time the GPPM updates the reference, it is verified if the line of sight connecting the guided segment $\mathrm{p}_{i,t}$ (with $i \in \mathbb{Z}_{[0,N_t]}$ corresponding to that index of the segment being guided) to r_c is free of obstacles. The segment of line connecting $\mathrm{p}_{i,t}$ to r_c can be written as follows:

$$\begin{cases} x_i = x_{i,t} + (x(\gamma_{c,t}) - x_{i,t})\zeta \\ y_i = y_{i,t} + (y(\gamma_{c,t}) - y_{i,t})\zeta \end{cases}$$

where $\zeta \in \Re_{[0,1]}$. The distance function $\mathcal{J}_o$ is defined as follows:

$$\mathcal{J}_o := \sqrt{(x_i-x_\varrho)^2+(y_i-y_\varrho)^2} - r_\varrho - \dim(\mathrm{p}_{i,t}) \tag{8}$$

with $\dim(\mathrm{p}_{i,t}) = \mathrm{width}_i/2 + \mathrm{len}_{\mathrm{proj}}(\mathrm{p}_{i,t})/2 + \mathrm{s_m}$, where $\mathrm{len}_{\mathrm{proj}}(\mathrm{p}_{i,t})$ is the length of the $i-th$ segment projected onto the direction toward the obstacle, and ϱ is the index over the static and dynamic obstacles. Then, the function $\mathcal{J}_o$ is minimised within the following constrained optimisation problem:

$$\begin{aligned} &\min_{\zeta} \quad \mathcal{J}_o \\ &\text{s.t.}\left\{\quad 0 \le \zeta \le 1\right. \end{aligned} \tag{9}$$

Then, for $\mathcal{J}_0 \le 0$, the target is occluded.

G. Nonlinear model predictive path-following controller (NMPPFC)

Let us assume that references generated by Eq. (3) are meant to be followed by the $i-th$ segment of the vehicle, i.e., the goal is to steer $\mathrm{p}_{i,t}$ to r_c. Then, to achieve path-following of the $i-th$ segment of the generalised N-trailer vehicle while collision of the whole chain with static and dynamic obstacles is avoided, we propose the minimisation of the following objective function:

$$\mathcal{J} := \int_{t_k}^{t_k+T_c+T_p}\left(\ell_q(\mathrm{p}_{i,\tau}-\mathrm{r}_c) + \ell_u(u_\tau) + \sum_{i=0}^{N_t}\sum_{m=1}^{\mathcal{O}_s}\sum_{l=1}^{\mathcal{O}_d}\Lambda_{m,i}+\Phi_{l,i,\tau}\right)d\tau + V_f(q_{t_k+T_c+T_p}) \tag{10}$$

where $\mathcal{O}_s$ and $\mathcal{O}_d$ denote the number of static and dynamic obstacles, respectively. The values of T_c and T_p are the control and prediction window lengths, respectively, with $T_p \ge 0$. When solving the problem given by Eq. (10), the optimisation variables u_τ are only computed for $\tau \in \Re_{[t_k,\, t_k+T_c]}$, with $u_\tau = u_{t_k+T_c}\ \forall\, \tau \in \Re_{[t_k+T_c, t_k+T_c+T_p]}$. However, the vehicle's state and obstacle positions are forecast beyond $t_k + T_c$. It allows predicting the position of the obstacles over a long time horizon without incurring a high computational cost. The functions $\ell_q \succeq 0$, $\ell_u \succeq 0$, together with $\Lambda_{m,i}$ and $\Phi_{l,i,t}$, constitute the so-called stage costs, whilst $V_f(q_{t_k+T_c+T_p})$ is the term known as *cost-to-go*. The function ℓ_q penalises deviations of $\mathrm{p}_{i,t}$ respect to the reference r_c, while ℓ_u penalises large values of the velocities $\omega_{0,t}$ and $v_{0,t}$, minimising in this way the control energy. The objective function given by Eq. (10) is minimised within the following constrained optimisation problem:

$$\begin{aligned} &\min_{u_\tau} \quad \mathcal{J} \\ &\text{s.t.}\begin{cases} \dot{q}_\tau = & G(q_\tau)\,u_\tau, \\ \underline{u} \le & u_\tau \le \overline{u}, \\ \underline{\dot{u}} \le & \dot{u}_\tau \le \overline{\dot{u}}, \\ -\frac{\pi}{2}+\delta_\beta \le & \beta_{i,\tau} \le \frac{\pi}{2}-\delta_\beta, \\ q_\tau \in \mathcal{Q}, & q_{t_k+T_c+T_p} \in \mathcal{Q}_f, u_\tau \in \mathcal{U} \end{cases} \end{aligned} \tag{11}$$

where the constraints $\underline{u} \le u_\tau \le \overline{u}$ and $\underline{\dot{u}} \le \dot{u}_\tau \le \overline{\dot{u}}$ impose limits on the velocities and accelerations, while $\mathcal{Q}$ and $\mathcal{U}$ are the sets where states and controls are assumed to belong to, respectively. The terminal constraint $q_{t_k+T_c+T_p} \in \mathcal{Q}_f$, together with the *cost-to-go*, introduced in Eq. (10) are designed to achieve the stability of the closed-loop system. The constraint $-\pi/2+\delta_\beta \le \beta_{i,\tau} \le \pi/2-\delta_\beta$ limit the joint angles in order to avoid the jackknife effect, especially in backward manoeuvres. The parameter $\delta_\beta > 0$ is a tighten factor since $|\beta_{i,t}|$ can exceed $\pi/2$ due to noise and disturbances. By solving the formulation

given by Eq. (11), the sequence of controls $\mathbf{u}_{[t_k,t_k+T_c]}$ is obtained.

H. Discrete-time NMPPFC

In order to effectively deploy and solve the constrained optimisation problem given by Eq. (11), it is necessary to discretise the nonlinear continuous-time model. The controller is implemented in a sampled-data fashion, following the approach described in [**?**]. The control input u_τ is considered to be sampled and applied to a zero-order holder, maintaining a constant value throughout the sampling interval. Specifically, u_τ is set as u_k for all τ belonging to the interval $[t_k, t_{k+1}]$, where t_k and t_{k+1} represent consecutive sampling times and k takes values from 1 to $N_c + N_p$, where N_c and N_p are the control and prediction window lengths, respectively. Among the available discretisation methods, this work employs the multiple shooting technique, which is a numerical method used to solve initial value problems (IVP) over a finite time horizon. At each shooting point, the state variables and their derivatives are approximated using a numerical method, such as Euler's method or the Runge-Kutta method. The discrete-time version of the objective function given by Eq. (10) is as follows:

$$\mathbb{J} := \underbrace{\sum_{n=k+1}^{k+N_c+N_p} \Big(\ell_q(\mathrm{p}_{i,n} - \mathrm{r}_c) + \ell_u(u_n) + \sum_{i=0}^{N_t} \sum_{j=1}^{\mathcal{O}_s} \sum_{l=1}^{\mathcal{O}_d} \Lambda_{j,i} + \Phi_{l,i,n} \Big) p_{n-k}}_{\mathbb{J}_0(q_k,\, u)} \tag{12}$$

where the cost function $\mathbb{J}(q_k, u, n)$ (referred to as $\mathbb{J}$ in Eq. (12) for the sake of space) results from weighing the cost $\mathbb{J}_0(q_k, u)$ with coefficients $p_{\tilde{n}}$, where $\tilde{n} = n - k$, which are introduced for stability purposes, as will be further explained later. The optimisation problem to be solved at every sampling time is the following:

$$\min_{u_{[k,k+N_c-1]}} \quad \mathbb{J}(q_k, u, \tilde{n})$$
$$\text{s.t.} \begin{cases} q_n = & \mathcal{G}(q_{n-1}, u_{n-1}), \\ \underline{u} \leq & u_{n-1} \leq \overline{u}, \\ \underline{\Delta} u \leq & u_{n-1} - u_{n-2} \leq \overline{\Delta} u, \\ -\frac{\pi}{2} + \delta_\beta \leq & \beta_{i,n} \leq \frac{\pi}{2} - \delta_\beta, \\ q_n \in \mathcal{Q}, & u_j \in \mathcal{U} \end{cases} \tag{13}$$

where $n \in \mathbb{Z}_{[k+1,k+N_c]}$, $\mathcal{G}$ is a fourth-order Runge-Kutta (RK4) system's integrator, and $\underline{\Delta} u$ and $\overline{\Delta} u$ are the bounds for the rate of change of the controls. Solving the problem given by Eq. (13) provides a sequence of controls $\boldsymbol{u} = [u_k, u_{k+1}, \ldots, u_{k+N_c}]$. The control law to be applied is defined as $\mathbb{K}(q_k) = u_k$, as usual in the model predictive literature. It means that only the first element of the sequence $\boldsymbol{u}$ is applied to the system while the others are discarded. At the next sampling time, the vehicle's current state is updated and the whole sequence $\boldsymbol{u}$ is computed again, closing the loop with the system.

I. Stability of the NMPPFC

The stability and recursive feasibility of the problem given by Eq. (13) are guaranteed by designing the coefficients $p_{\tilde{n}}$ following the method described in [**?**]. In addition, this method consists of replacing the classical terminal invariant set $\mathcal{Q}_f$ from Eq. (11) with two simpler sets: the inner and outer sets Ω_I and Ω_O, respectively. For the sake of completeness, before proceeding let us define the *Up-To-N_c-Steps Controllable Set* $Q_{N_c}^{\Omega_I}$ (see Definition 1, page 3 of [**?**]):

Definition 1 (*Up-To-N_c-Steps Controllable Set* $Q_{N_c}^{\Omega_I}$). Given the system of Eq. (2), the horizon length N_c and the inner set $\Omega_I \subset \Re^3$, we say that $Q_{N_c}^{\Omega_I} \subseteq \mathcal{Q}$ is the up-to-N_c-steps controllable set to Ω_I provided that if $q_k \in Q_{N_c}^{\Omega_I}$, then there exists a sequence of control inputs $\boldsymbol{u} = (u_k, ..., u_{k+n-1}) \in \mathcal{U}^n$ for some $n \in \mathbb{Z}_{[1,N_c]}$ for which $\mathrm{proj}_i(q_{k+n}) \in \Omega_I$ and $\{q_k, ..., q_{k+n-1}\} \in \mathcal{Q}^n$.

Remark 1. Note that the set $Q_{N_c}^{\Omega_I}$ denotes all possible initial states q_k of the vehicle given by Eq. (2) such that $\mathrm{proj}_i(q_k) = \mathrm{p}_{i,k}$ can be driven inside the set Ω_I in at most N_c steps.

In [**?**], Ω_I is not necessarily the same as Ω_O. In fact, $\Omega_I = \Omega_O$ entails that these sets are an invariant space for the closed-loop system, and this is exactly the condition which is relaxed in [**?**]. However, in this work, the sets $\Omega_I = \Omega_O$, since $G(q_t)u_t = 0$ for $u_t = [0, 0]^T$, according to Eq. (2), or $q_k = \mathcal{G}(q_k, [0, 0]^T)$, for the discrete-time case (i.e., once the set Ω_I has been reached, the pose $\mathrm{p}_{i,k}$ can stay inside by applying no controls. Then, the outer set can be as small as the inner set). Since $\Omega_I = \Omega_O$ in our case, they will be referred to as Ω_k for simplicity. Moreover, it will be defined as a box set centred at the reference $\mathrm{r}_c = \mathrm{p}_{xy\theta}(\gamma_{c,k})$, as follows:

$$\Omega_k := \left\{ \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} : \begin{matrix} \underline{x} \leq & x - x(\gamma_{c,k}) \leq & \overline{x} \\ \underline{y} \leq & y - y(\gamma_{c,k}) \leq & \overline{y} \\ \underline{\theta} \leq & \theta - \theta(\gamma_{c,k}) \leq & \overline{\theta} \end{matrix} \right\} \begin{matrix} \text{(m)} \\ \text{(m)} \\ \text{(rad)} \end{matrix} \tag{14}$$

where $\underline{x}$, $\overline{x}$, $\underline{y}$, $\overline{y}$, $\underline{\theta}$ and $\overline{\theta}$ determine the volume of the set Ω_k. Note moreover that since $\mathrm{p}_{xy\theta}(\gamma_{c,t}) = (x(\gamma_{c,t}), y(\gamma_{c,t}), \theta(\gamma_{c,t}))^T$ is time-varying, the set Ω_k is time-varying too, in contrast to the inner and outer sets defined in [**?**], which are time-invariant. Then, assuming that $q_k \in Q_{N_c}^{\Omega_k}$ and whenever the coefficients $p_{\tilde{n}}$ are designed as follows (see [**?**]):

$$p_{\tilde{n}} \begin{cases} c \; \forall \, \tilde{n} \in \mathbb{Z}_{[1,N_c]} & \text{if } \mathrm{p}_{i,k} \in \Omega, \\ (1 + (\tilde{n} - 1)\, r) & \text{if } \mathrm{p}_{i,k} \notin \Omega \end{cases} \tag{15}$$

where $c \geq 0$ and $r \in [0,\, 1)$ are such that $c\, N_c\, \mathbb{J}_{0_{\max}} \leq r\, \mathbb{J}_{0_{\min}}$ with:

$$\mathbb{J}_{0_{\min}} := \inf_{q_k \,\in\, \mathcal{Q} \setminus \Omega_k,\; u \,\in\, \mathcal{U}} \mathbb{J}_0(q_k, u),$$
$$\mathbb{J}_{0_{\max}} := \sup_{q_k \,\in\, \Omega_k,\; u \,\in\, \mathcal{U}} \mathbb{J}_0(q_k, u)$$

Then, practical stability and recursive feasibility are guaranteed during the path-following in the presence of obstacles when the generalised N-trailer vehicle is controlled by solving at each sampling time the problem given by Eq. (13) $\square$.

Remark 2. It is worth noting the following key aspects: i) The number of trailers is not involved in the former analysis, as

well as the geometrical configuration of the vehicle. However, they are embedded into the set $Q_{N_c}^{\Omega_k}$. As the number of trailers increases and the geometry varies, the manoeuvrability is compromised (see [**?**], [**?**]), entailing *smaller* sets $Q_{N_c}^{\Omega_k}$ for a fixed N_c, Ω_k and a growing number of trailers. In Section Experiments and Results, we compute numerically these sets. ii) The set $Q_{N_c}^{\Omega_k}$ can be enlarged by increasing N_c. However, the real-time applicability could be compromised due to an increase in the computational burden required to solve the problem given by Eq. (13). iii) Since the set Ω_k is time-varying, it is difficult to guarantee that $\mathrm{r}_c \in \mathrm{proj}_i(Q_{N_c}^{\Omega_k})$ $\forall\, k \geq 0$, especially when the current reference r_c is occluded by obstacles or the vehicle is forced to implement an evasive manoeuvre to avoid collision with dynamic obstacles, moving away $\mathrm{proj}_i(q_k) = \mathrm{p}_{i,k}$ from r_c. Then, for the cases where $\mathrm{r}_c \notin \mathrm{proj}_i(Q_{N_c}^{\Omega_k})$, we propose the use of a local planner module that generates an auxiliary reference $\tilde{\mathrm{r}}_c$ such that $\tilde{\mathrm{r}}_c \in \mathrm{proj}_i(Q_{N_c}^{\Omega_k})$, where Ω_k is computed replacing $\mathrm{r}_c = \mathrm{p}_{xy\theta}(\gamma_{c,k})$ with $\tilde{\mathrm{r}}_c$ in Eq. (14), i.e., it keeps its volume but is now centred at $\tilde{\mathrm{r}}_c$.

J. Local path planner module

When the reference r_c is occluded by static or dynamic obstacles, or when the vehicle implements manoeuvres to evade obstacles such that $\mathrm{proj}_i(q_k) = \mathrm{p}_{i,k}$ moves away from r_c such that $\mathrm{r}_c \notin Q_{N_c}^{\Omega_k}$, the Local Path Planner Module (LPPM) is in charge of generating an alternative reference $\tilde{\mathrm{r}}_c$ inside the set $\mathrm{proj}_i(Q_{N_c}^{\Omega_k})$. To generate the auxiliary reference $\tilde{\mathrm{r}}_c$, the following objective function: $\tilde{\mathbb{J}} = \frac{1}{2}|\mathrm{r}_c - \tilde{\mathrm{r}}_c|^2$ is defined, which is minimised within the following constrained optimisation problem:

$$\min_{\tilde{\mathrm{r}}_c} \quad \tilde{\mathbb{J}}$$
$$\text{s.t.} \begin{cases} \tilde{\mathrm{r}}_c \in & \mathrm{proj}_i(Q_{N_c}^{\Omega_k}), \\ x_i = & x_{i,k} + (\tilde{x}_r - x_{i,k})\,\zeta, \\ y_i = & y_{i,k} + (\tilde{y}_r - y_{i,k})\,\zeta, \\ r_{j,k}^2 < & (x_i - x_{j,k})^2 + (y_i - y_{j,k})^2 \;\forall\, \zeta \in \Re_{[0,1]} \end{cases} \tag{16}$$

where, the constraint $\tilde{\mathrm{r}}_c \in \mathrm{proj}_i(Q_{N_c}^{\Omega_k})$ force the reference to be inside the set $\mathrm{proj}_i(Q_{N_c}^{\Omega_k})$, entailing that $\tilde{\mathrm{r}}_c$ can be reached from $\mathrm{proj}_i(q_k)$. The constraints $r_{j,k}^2 < (x - x_{j,k})^2 + (y - y_{j,k})^2$ $\forall\, t \in \Re_{[0,1]}$ guarantee that the segments of the line connecting the $i-th$ element of the chained vehicle to the new reference, i.e., $\overline{\mathrm{p}_{i,k}\,\tilde{\mathrm{r}}_c}$ is not passing through the $j-th$ obstacle. Note that this is a heuristic rule, and further analysis involving the manoeuvrability of the N-trailer vehicle is required, which is a challenging task due to the variable number of trailers and geometry.

Remark 3. Note that the problem given by Eq. (16) introduces hard constraints. However, this problem is only solved when $\mathrm{r}_c \notin \mathrm{proj}_i(Q_{N_c}^{\Omega_k})$. In addition, any sub optimal solution fulfilling the constraints would be useful.

Figure 2 aims to illustrate the interaction between the different components of the proposed framework and the equations involved in each stage. Moreover, Figure 3 shows the detailed

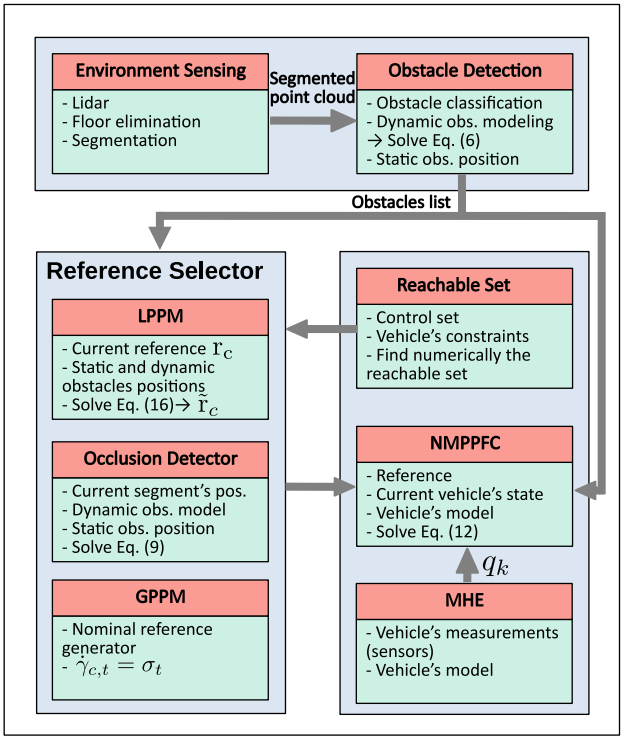

Fig. 2: General diagram showing the interaction between the different functional blocks of the proposed diagram.

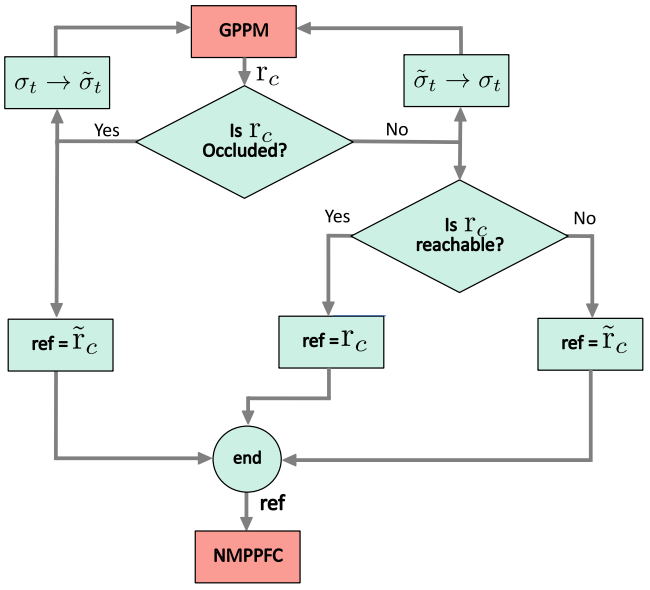

Fig. 3: Reference selector logic. It is in charge of selecting a reachable reference to the NMPPFC.

functioning of the reference selector block. Once the reference is selected, it is passed to the NMMPPFC, which, together with the current vehicle's state (provided by the Moving Horizon Estimator), computes the velocities to be applied to the N-trailer vehicle. Finally, Algorithm 1 summarises the steps involved in solving the NMPPFC framework.

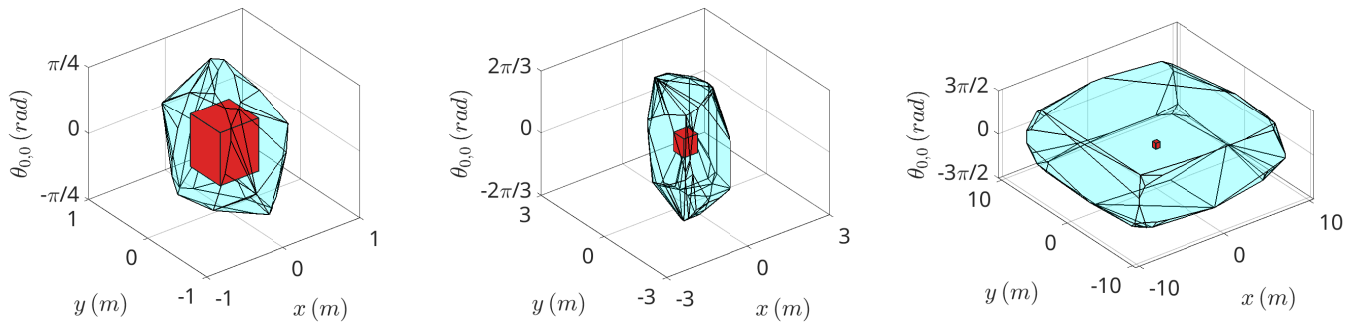

Fig. 4: Different sets $Q_{N_c}^{\Omega_k}$ (cyan) computed for $N_c = 15$, Ω_k (red) defined according to Eq. (17) and centred at the origin, $\underline{\Delta}\omega = -0.6$ (rad/s^2), $\overline{\Delta}\omega = 0.6$ (rad/s^2) $\underline{\Delta}v = -0.3$ (m/s^2) and $\overline{\Delta}v = 0.3$ (m/s^2) (left), $\underline{\Delta}\omega = -1.2$ (rad/s^2), $\overline{\Delta}\omega = 1.2$ (rad/s^2) $\underline{\Delta}v = -0.6$ (m/s^2) and $\overline{\Delta}v = 0.6$ (m/s^2) (centre), and $\underline{\Delta}\omega = -6$ (rad/s^2), $\overline{\Delta}\omega = 6$ (rad/s^2) $\underline{\Delta}v = -3$ (m/s^2) and $\overline{\Delta}v = 3$ (m/s^2) (right).

Algorithm 1: NMPPFC framework Algorithm

Input : T_s, G, σ_t, $\tilde{\sigma}_t$, A_m, A_l, $\mathrm{s_m}$, N_c, N_p, $\mathcal{U}$, Ω_k, $Q_{N_c}^{\Omega_k}$

while $\mathrm{p}_{i,k}$!(reach end of the path) **do**
- **STEP 1:** Update measurements
- **STEP 2: call** MHE
- **STEP 3:** Update obstacles list
- **for** *i=1 to* $\mathcal{O}_d$ **do**
  - **if** !(is ellipse Eq. a good model) **then**
    - **STEP 4:** Update ellipse Eq.
- Reference Selector:
- **STEP 5: call** GPPM
- **STEP 6: call** Occlusion Detector
- **if** ref is occluded $\vee$ is not reachable **then**
  - **STEP 7 call** LPPM
- **STEP 8: call** Update reference
- **STEP 9: call** Update current vehicle's state
- **STEP 10: call** NMPPFC
- **STEP 11:** Apply velocities to the vehicle

IV. Experiments and Results

In this section, we validate the proposed framework through a combination of simulated and field experiments. In simulated experiments, the scenario includes obstacle navigation with a tractor pulling a maximum of 3 passive trailers, and encountering several static and dynamic obstacles during the path-following task. In addition, we compare our method against a method from the state of the art. Specifically, we implemented the strategy published in [**?**], where the authors proposed a model-based approach for unmanned ground vehicles in the presence of disturbances and dynamic obstacles.

Additionally, field experiments are conducted with a tractor pulling 2 passive trailers in a generalised configuration. These experiments challenge the system to navigate different paths when static and dynamic obstacles obstruct the nominal path. A LiDAR scans the surrounding environment for obstacle detection. The NMPPFC is implemented using a sampled-data approach, following the methodology outlined in [**?**]. The discretisation technique employed is the multiple shooting method. Furthermore, the state variables and their derivatives are approximated using a fourth-order Runge-Kutta (RK4) numerical method.

A. Stability sets

Before conducting the experiments, the sets Ω_k and $Q_{N_c}^{\Omega_k}$ from Definition 1 have to be defined for a given N_c. In addition, the Eq. (15) needs to be solved to properly weigh the cost function to guarantee stability according to [**?**]. We design the set Ω_k as a box set such that:

$$\Omega_k := \left\{ \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} : \begin{matrix} -0.25 \leq & x - x(\gamma_{c,k}) \leq & 0.25 \\ -0.25 \leq & y - y(\gamma_{c,k}) \leq & 0.25 \\ \frac{-\pi}{10} \leq & \theta - \theta(\gamma_{c,k}) \leq & \frac{\pi}{10} \end{matrix} \right\} \begin{matrix} \text{(m)} \\ \text{(m)} \\ \text{(rad)} \end{matrix} \tag{17}$$

Then, for each vehicle's configuration and control constraints, the set $Q_{N_c}^{\Omega_k}$ is computed numerically once off-line. A grid of initial conditions q_0 is generated. Then, all those starting points from which $\mathrm{proj}_i(q_0)$ can be steered to Ω_k in at most N_c steps shape the set $Q_{N_c}^{\Omega_k}$ (see [**?**] for more details). The set of controls $\mathcal{U}$ is also designed as a box set, as follows:

$$\mathcal{U} := \left\{ \begin{pmatrix} \omega_{0,k} \\ v_{0,k} \end{pmatrix} : \begin{matrix} -2 \leq & \omega_{0,k} \leq & 2 \\ \underline{\Delta}\omega \leq & \frac{\omega_{0,k} - \omega_{0,k-1}}{T_s} \leq & \overline{\Delta}\omega \\ -1 \leq & v_{0,k} \leq & 1 \\ \underline{\Delta}v \leq & \frac{v_{0,k} - v_{0,k-1}}{T_s} \leq & \overline{\Delta}v \end{matrix} \right\} \begin{matrix} \text{(rad/s)} \\ \text{(rad/s}^2\text{)} \\ \text{(m/s)} \\ \text{(m/s}^2\text{)} \end{matrix} \tag{18}$$

The values of ± 2 (rad/s) and ± 1 (m/s) are taken from the manual of the vehicle used in this work as a tractor, while the accelerations need to be carefully selected to achieve a set $Q_{N_c}^{\Omega_k}$ as large as possible for given N_c and Ω_k and at the same time achieve smooth manoeuvres. Figure 4 shows the set Ω_k (red) and $Q_{N_c}^{\Omega_k}$ (cyan) for $N_c = 15$ and different values of $\underline{\Delta}\omega$, $\overline{\Delta}\omega$, $\underline{\Delta}v$ and $\overline{\Delta}v$, plotted using the multi-parametric toolbox [**?**]. The largest set $Q_{N_c}^{\Omega_k}$ is obtained when the accelerations are set at their maximum values, reported in the vehicle manual.

The stage-costs functions are chosen to be quadratic such that $\ell_q(\mathrm{p}_{i,n} - \mathrm{r}_c) = (\mathrm{p}_{i,n} - \mathrm{r}_c)^T Q (\mathrm{p}_{i,n} - \mathrm{r}_c)$, $\ell_u(u_n) =$

$u_n^T\, R\, u_n$, with $Q = \text{diag}(1,\ 10,\ 10)$ and $R = \text{diag}(0.05,\ 0.1)$ such that $Q, R \succ 0$. In addition, the amplitude of the Gaussian functions for the static and dynamic obstacles are 60 and 100, respectively. To the obstacle size, half of the largest segment's size plus a safety margin $\text{s}_\text{m} = 0.1$ (m), totalling 0.64 (m) is added. Then, once the sets Ω_k and $\mathcal{U}$ have been defined, the values of $\mathbb{J}_{0_\text{min}}$ and $\mathbb{J}_{0_\text{max}}$ needed to find the sequence p_{n-k} can be computed. It gives the following values: $\mathbb{J}_{0_\text{min}} = 0.707$ and $\mathbb{J}_{0_\text{max}} = 7.847$. Choosing $r = 0.1$ in Eq. (15) and $N_c = 25$ gives $c = 3.603 \times 10^{-4}$.

B. Simulations Study

In the following, we perform simulated experiments with different configurations of number of passive trailers and static and dynamic obstacles. In addition, we compare the proposed method (NMPPFC) against the one published in [**?**] (FNMPPC), using the kinematic model of the N-trailer vehicle. The experiments consist of following the Lemniscate of Bernoulli, which is an ∞-shape curve with different values of curvatures. The parametric equation of the path is as follows:

$$\begin{cases} x(\gamma_{c,t}) = & \dfrac{\sqrt{32}\cos(\gamma_{c,t})}{\sin(\gamma_{c,t})^2+1}\ \text{(m)} \\ y(\gamma_{c,t}) = & \dfrac{\sqrt{128}\cos(\gamma_{c,t})\sin(\gamma_{c,t})}{\sin(\gamma_{c,t})^2+1}\ \text{(m)} \end{cases}$$

for $\gamma_{c,t} \in [0,\ 2\pi]$. The experiments aim to evaluate the performance in terms of the mean deviation of the guided segment to the nominal path (Δ), the computational burden (η) and the control effort (Ψ), which are metrics of interest commonly evaluated in the literature of model-based path-following controllers. In addition, we compute the minimum distance between the vehicle's segments to the obstacles (∇) to evaluate the collision avoidance performance. The geometrical configuration of the chained vehicle is $\text{L}_{\text{h}_1} = 0.342$ (m), $\text{L}_{\text{h}_2} = 0$ (m), $\text{L}_1 = 1.08$ (m) and $\text{L}_2 = 0.78$, which is a generalised configuration. The controls are constrained to the box-set $\mathcal{U}$ given by Eq. (18), with $\underline{\Delta}\omega = -6$ (rad/s^2), $\overline{\Delta}\omega = 6$ (rad/s^2), $\underline{\Delta}v = -3$ (m/s^2) and $\overline{\Delta}v = 3$ (m/s^2). Both methods are configured with the same constraints, stage-cost matrices and control window length $N_c = 25$ and the sampling-time is $T_s = 0.05$ (s), as in [**?**]. In addition, the proposed method (NMPPFC) uses a prediction horizon $N_p = 25$.

Twelve different conditions are evaluated, which consist of a tractor pulling 1, 2 and 3 passive trailers. For each number of trailers, 1, 2, 4 and 6 static ($\text{O}_{s,i}$) and dynamic ($\text{O}_{d,i}$) obstacles, appearing both simultaneously, are considered. The static obstacles are configured as follows (x, y, radius) (m): $\text{O}_{s,1} = (10.1,\ 8.1,\ ,0.15)$ (m), $\text{O}_{s,2} = (8,\ 8,\ ,0.15)$ (m), $\text{O}_{s,3} = (6,\ 4,\ ,0.25)$ (m), $\text{O}_{s,4} = (4,\ 4,\ ,0.25)$ (m), $\text{O}_{s,5} = (2,\ 0.5,\ ,0.35)$ (m) and $\text{O}_{s,6} = (0,\ 4,\ ,0.15)$ (m). The dynamic obstacles behave according to the following dynamics:

$$\text{O}_{d,1} = \begin{cases} x_k^1 = & 3 + 4\cos(\pi + 0.0675\, k\, T_s)\ \text{(m)} \\ y_k^1 = & 5 + 3\sin(\pi + 0.0675\, k\, T_s),\ 0.2)\ \text{(m)} \\ r^1 = & 0.2\ \text{(m)} \end{cases}$$

$$\text{O}_{d,2} = \begin{cases} x_k^2 = & \dfrac{\sqrt{32}\cos(t^{1.2}+\pi/4)}{\sin(t^{1.2}+\pi/4)^2+1} + 5.5\ \text{(m)} \\ y_k^2 = & \dfrac{\sqrt{128}\cos(t^{1.2}+\pi/4)\sin(t^{1.2}+\pi/4)}{\sin(t^{1.2}+\pi/4)^2+1} + 4.5\ \text{(m)} \\ r^2 = & 0.2\ \text{(m)} \\ t = & -(0.015\, k\, T_s)\ \text{(s)} \end{cases}$$

$$\text{O}_{d,3} = \begin{cases} x_k^3 = & 5 + 6\cos(\pi + 0.0675\, k\, T_s)\ \text{(m)} \\ y_k^3 = & 4.8\ \text{(m)} \\ r^3 = & 0.35\ \text{(m)} \end{cases}$$

$$\text{O}_{d,4} = \begin{cases} x_k^4 = & 7\ \text{(m)} \\ y_k^4 = & 5 + 3\sin(\pi + 0.0675\, k\, T_s)\ \text{(m)} \\ r^4 = & 0.1\ \text{(m)} \end{cases}$$

$$\text{O}_{d,5} = \begin{cases} x_k^5 = & 9 + 1.5\cos(\pi + 0.08\, k\, T_s)\ \text{(m)} \\ y_k^5 = & 4 + 1.5\sin(\pi + 0.08\, k\, T_s)\ \text{(m)} \\ r^5 = & 0.05\ \text{(m)} \end{cases}$$

$$\text{O}_{d,6} = \begin{cases} x_k^6 = & 2 + 1.5\cos(\pi - 0.08\, k\, T_s)\ \text{(m)} \\ y_k^6 = & 6 + 1.5\sin(\pi - 0.08\, k\, T_s)\ \text{(m)} \\ r^6 = & 0.05\ \text{(m)} \end{cases}$$

Note that the obstacle $\text{O}_{d,2} =$ moves along the ∞-shape path in the opposite direction of the vehicle. Though all obstacles are accelerated (their velocity change over time), the obstacle $\text{O}_{d,2} =$ moves with a speed that increases over time.

Finally, the metrics used to compare the performance are computed as follows:

$$\begin{aligned} \Delta =& \frac{1}{K}\sum_{k=1}^{K} \min\left(|\text{p}_{i,k} - \text{p}_{xy\theta}(\gamma_{c,\kappa})|\right),\ \forall\, \kappa \in \mathbb{Z}_{[1,K]}\ \text{(m)}, \\ \eta =& \frac{1}{K}\sum_{k=1}^{K} T_k\ \text{(s)}, \\ \nabla =& \min\left(\mathbf{d}(\text{p}_{i,k}, \text{O}_{s,j}),\ \mathbf{d}(\text{p}_{i,k}, \text{O}_{d,l})\right)\ \text{(m)} \\ & \forall\, k \in \mathbb{Z}_{[1,K]}, i \in \mathbb{Z}_{[0,N_t]}, j \in \mathbb{Z}_{[1,\mathcal{O}_s]}, l \in \mathbb{Z}_{[1,\mathcal{O}_d]}, \\ \Psi =& \frac{1}{K}\sqrt{\sum_{k=1}^{K} \omega_{0,k}^2 + v_{0,k}^2}\ \left(\sqrt{(\text{rad/s})^2 + (\text{m/s})^2}\right) \end{aligned}$$

where the function $\mathbf{d}(\text{p}_{i,k}, \text{O})$, with $\text{O} = (x, y, \text{radius})$, x and y are the coordinates of the obstacle at time k, is defined as follows:

$$\mathbf{d}(\text{p}_{i,k}, \text{O}) := \sqrt{(x_{i,k} - x)^2 + (y_{i,k} - y)^2} - \text{radius} - \dim(\text{p}_{i,k})$$

with $\dim(\text{p}_{i,k}) = \text{width}_i/2 + \text{len}_\text{proj}(\text{p}_{i,k})/2 + \text{s}_\text{m}$, where $\text{len}_\text{proj}(\text{p}_{i,k})$ is the length of the $i-th$ segment projected onto the direction toward the obstacle and s_m is the safety margin, which is chosen to 0.1 (m). Note that $\mathbf{d}(\text{p}_{i,k}, \text{O}) < 0$ entails a collision.

Figure 6 summarises the performance of both methods. The horizontal axe denotes the trial's configuration: 1-3: 1 static and dynamic obstacles with a tractor pulling 1, 2 and 3 trailers, 4-6: 2 static and dynamic obstacles with a tractor pulling 1, 2 and 3 trailers, 7-9: 4 static and dynamic obstacles with a tractor pulling 1, 2 and 3 trailers, and 10-12: 6 static and dynamic obstacles with a tractor pulling 1, 2 and 3 trailers.

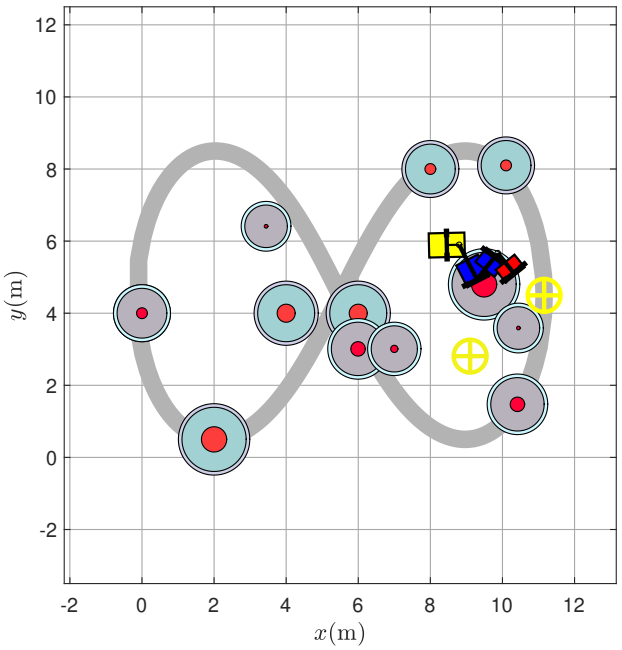

Fig. 5: Collision against a dynamic obstacle (grey obstacles) during trial 12.

Figure 6-left summarises the mean deviation of the guided segment (the tractor) with respect to the nominal path. The proposed method (NMPPFC) presents a higher deviation in general. In addition, the deviation seems to increase with the number of trailers and the number of obstacles. The second picture from left to right in Figure 6 depicts the minimum distance between the segments (tractor and trailers) and the obstacles. The competitor method fails to avoid collision with the obstacles on almost all the trials, while the proposed method only fails in the most challenging conditions where the total number of obstacles is 12.

The third (from left to right) bar plot in Figure 6 shows the mean control effort per sampling step. The competitor method requires less control effort. In those trials where the proposed method succeed in evading obstacles, the control effort exhibits a clear pattern as a function of the number of trailers and obstacles. Figure 6-right summarise the mean normalised time required for both methods in computing the velocities to be applied to the vehicle. The proposed method requires much lower time in computing the velocities, even when it is using the prediction horizon $N_p = 25$. Moreover, the mean time required for the competitor method is extremely high for some of the trials.

Figure 5 shows the most challenging situation where the vehicle navigates among 12 obstacles and a collision occurs. Green and Gray circumferences represent the static and dynamic obstacles, respectively. The proposed method is not capable of avoiding this collision with its current configuration. However, choosing $N_c = 35$, $N_p = 65$, increasing the amplitude of the Gaussian functions to 100 for both, static and dynamic obstacles, and increasing the safe margin from 0.1 (m) to 0.3 (m), the vehicle succeeds in avoiding collision with the 6 static and 6 dynamic obstacles at the cost of an increased mean deviation with respect to the nominal path. Figures 7-left and 7-right summarise the new values of Δ (m) and ∇ (m) for this new configuration.

C. Field Experiments

1) Hardware and software implementation: The proposed framework, given by Eq. (13), was implemented with CasADi [**?**] on Matlab+ROS (Robot Operating System). The algorithms run on a PC Dell Latitude 5430 Rugged with 16 GB of RAM and an 11-th Gen Intel Core i7 with Ubuntu 18.04. The vehicle available for experiments is composed of a Clearpath Husky A200 and 2 passive trailers, where only one has carriage capacity. The vehicle and its sensors are shown in Figure 8.

The proposed framework runs on the computer numbered as 2 in Figure 8. All available sensors are connected through the USB port with the help of a hub, and they are accessed and read using ROS on Matlab. The framework computes the velocities $\omega_{0,k}$ and $v_{0,k}$, which are considered to be sampled and applied to a zero-order holder, maintaining a constant value throughout the sampling interval. The velocities are sent to the Husky A200 tractor over the USB port using ROS commands. The internal computer of this vehicle transforms the velocities into torques to be applied to the wheels. The geometrical configuration of the chained vehicle is $\mathrm{L}_{\mathrm{h}_1} = 0.342$ (m), $\mathrm{L}_{\mathrm{h}_2} = 0$ (m), $\mathrm{L}_1 = 1.08$ (m) and $\mathrm{L}_2 = 0.78$.

2) Estate estimation: Furthermore, some of the vehicle's states are not measured, such as the position of the trailers, which is essential to effectively implement collision avoidance. Therefore, a Moving Horizon Estimator (MHE) is used to estimate the trailer's position and filter the measured states' noise. The measured states, together with the vehicle's model, are used within the MHE to estimate the whole system's state, which is then fed to the NMPPFC (see [**?**], [**?**] for more details on MHE).

3) Obstacle detection: The environment is scanned with the help of a Lidar Velodyne VLP16. It is connected to the Ethernet port of the computer. The Lidar is configured with its minimum rotational speed to obtain a denser point cloud. The data is read using standard Matlab commands. Point cloud processing involves floor elimination and segmentation by density and volume. Once the detected objects have been isolated, their positions are computed as the midpoint of their minimum and maximum values on x and y, and their size is taken as the radius of their outer circumference.

Experiment Number 1: The initial experiments involve following a straight line segment with a length of 25 (m). Two static obstacles are located on the path, such that $\mathrm{O}_{s,1} = (5.7, 6.15, 0.5)$ (m) and $\mathrm{O}_{s,2} = (18.5, 6.5, 0.6)$ (m). The objective function is designed again to be quadratic. Deviations of $\mathrm{p}_{i,n}$ respect to r_c are penalised with the following matrix: $\mathrm{Q} = 5\,\mathbf{I}_{2(N+1)\times 2(N+1)}$, the control energy tends to be minimised through the matrix $\mathrm{R} = \mathrm{diag}(0.35,\, 0.05)$. Figure 9 shows the estimated vehicle's position over 5 trials of the field experiment.

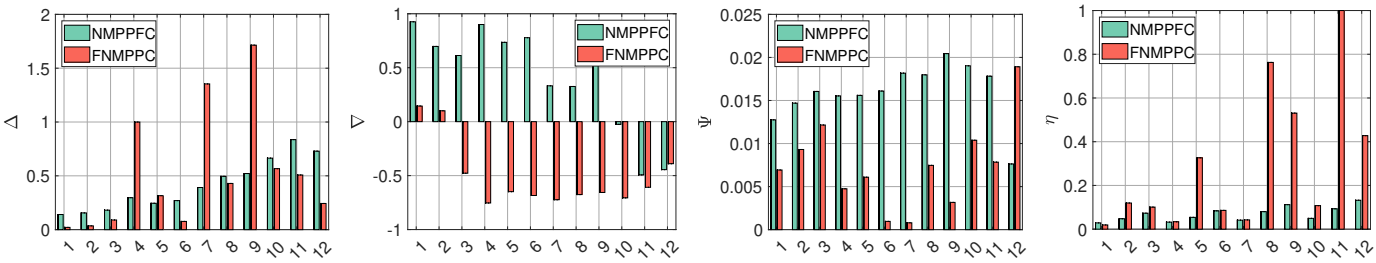

Fig. 6: From left to right: mean deviation (Δ) of the guided segment with respect to the nominal path for the proposed (green) and the competitor (red); minimum distance between the vehicle's segments and both, static and dynamic obstacles (∇) (negative values indicate collision); mean control effort per sampling instant (Ψ), and, mean normalised time (η) required for both methods to solve the control problems.

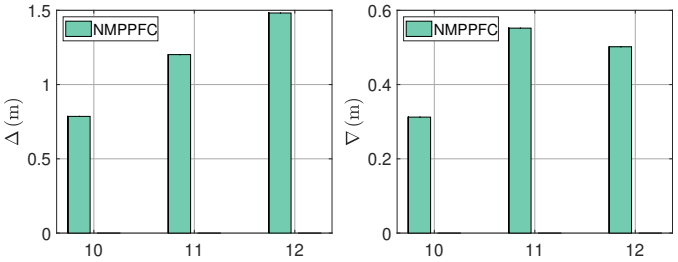

Fig. 7: Mean deviation (Δ) with respect to the nominal path (left) and minimum distance to the obstacles (∇) with the new configuration.

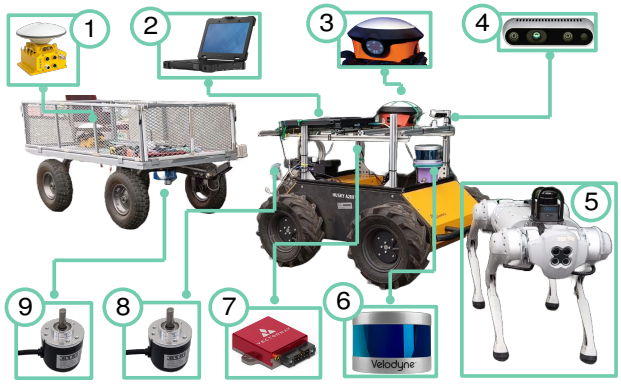

Fig. 8: Generalised N-trailer vehicle used during the filed experiments with $N = 2$. The G2T vehicle is equipped with the following: 1) Swift Duro GPS (not used during the experiments); 2) PC Dell Latitude 5430 with Ubunbtu 18.04, Matlab 2023 and the Robot Operating System (ROS) 18.04; 3) Navcom SF-3040 RTK GPS; 4) Intel Realsense Depth Camera 435 (not used during the experiments): 5) Unitree Go1 legged robot used as dynamic obstacle; 6) LiDAR Velodyne VLP16; 7) Vectornav VN200 GNSS-Aided Inertial Navigation System; 8) Encoder for measuring $\beta_{1,t}$; 9) Encoder for measuring $\beta_{2,t}$.

Experiment Number 2: The second experiment involves following the path given by the following parametric equation:

$$\begin{cases} x(\gamma_{c,t}) = & a\sin(\gamma_{c,t}) + 6\,(\text{m}) \\ y(\gamma_{c,t}) = & 7.5\sin(\gamma_{c,t})^2\cos(\gamma_{c,t}) + 4.5\,(\text{m}) \end{cases} \tag{19}$$

for $\gamma_{c,t} \in [0,\ \pi]$ and a is a parameter that determines how far the path spans through the x axis. Four different scenarios are considered in this second field experiment: i) a nominal path with $a = 5$ and one static obstacle $\text{O}_{s,1} = (10.9,\ 4.25)$ (m), ii) $a = 7.5$ and one static obstacle $\text{O}_{s,1} = (11.35,\ 4.25)$ (m), iii) $a = 7.5$ and $\text{O}_{s,1}(11.35,\ 4.25)$ (m) and $\text{O}_{s,2}(7.5,\ 2.25)$ (m), and, iv) $a = 7.5$ and two static obstacles $\text{O}_{s,1} = (11.35,\ 4.25)$ (m) and $\text{O}_{s,2} = (9.32,\ 2.44)$ (m).

Figure 10-A illustrates the vehicle's performance over 2 trials of the field experiments i). The path, given by Eq. (19), exhibits a larger curvature for the case of $a = 5$. Throughout the traversal of the path, the proposed framework successfully guided the vehicle backwards to evade the obstacle and minimise deviation from the nominal path. In Figure 11-A, both the computed and measured linear velocity profiles for one trial are presented. Following the obstacle evasion with a backward manoeuvre, the vehicle concludes the path with a forward manoeuvre. Importantly, the joint angles never exceeded $\pm\pi/2$, effectively avoiding the Jackknife effect. The constraints $-\pi/2 + \delta_\beta \leq \beta_{[1,2],t} \leq \pi/2 - \delta_\beta$ in formulation given by Eq. (13) were set with $\delta_\beta = 20 \times \pi/180$ (rad).

Figure 10-B shows the field experiments during scenario ii). During this scenario, the vehicle evades the obstacle with forward manoeuvres. The tractor's speed is positive throughout the journey, expect for a brief period when it moves backwards to implement a correcting manoeuvre, as shown in Figure 11-B for 1 trial. In scenario iii), which involves an additional obstacle, the vehicle successfully navigated, avoiding both obstacles and reaching the path's end. Figure 10-C illustrates the performance over 2 trials of field experiments in scenario iii), while Figure 11-C displays the linear velocity profile for one trial. Notably, the vehicle executes a brief backward manoeuvre during an off-tracking correction, as depicted. In scenario iv), one of the static obstacles was relocated in proximity to the nominal path. The vehicle successfully avoided the obstacles. Again, the manoeuvre to evade the second obstacle led to a higher deviation from the nominal path, as depicted in Figure 10-D.

Experiment Number 3: The third field experiment involves traversing a straight line segment with a length of 25 (m). One static and one dynamic obstacle are incorporated: $\text{O}_{s,1} = (18.5,\ 6.5)$ (m) while the dynamic obstacle does not follows a predefined movement. To ensure a safe interaction with

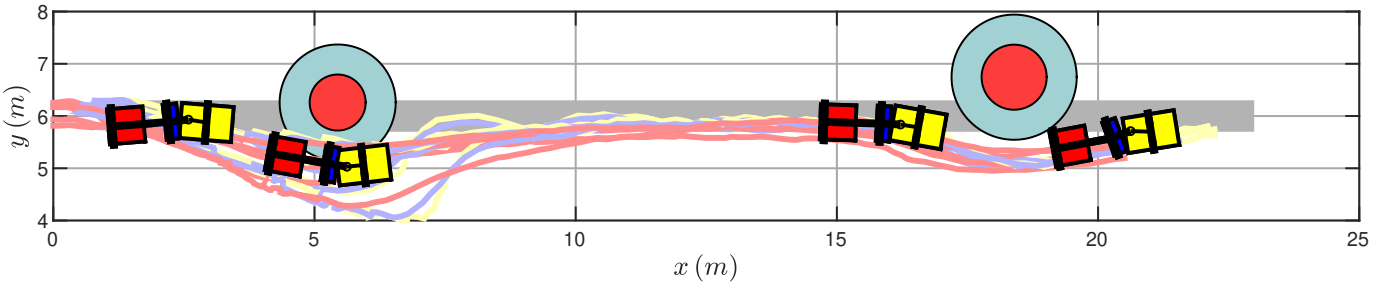

Fig. 9: Field Experiment 1: The path is a segment of a line approximately 25 (m) in length. Two static obstacles, depicted in red, are situated in the vicinity. The light blue circles indicate the size of the obstacles plus the safety margin (s_m).

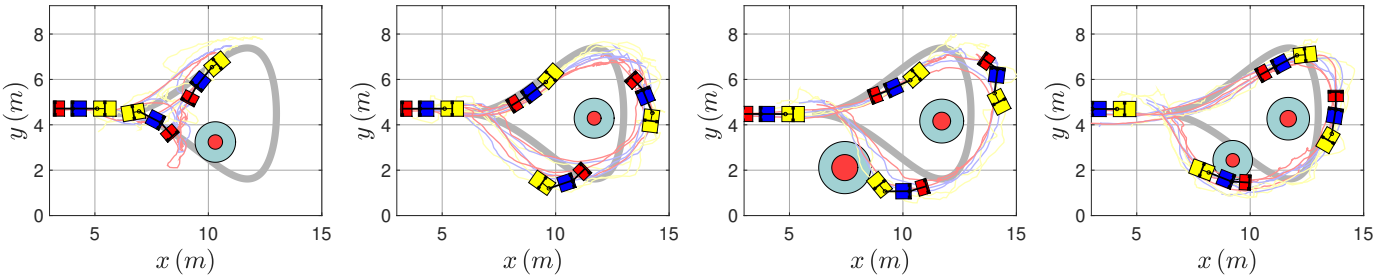

Fig. 10: Field Experiment 2: In Fig. A, the trajectory is shaped with $a = 5$ using Eq. (19), and the obstacle is positioned at $(10.9,\ 4.25)$ (m). The curvature is higher compared to Figs. B-D, where the vehicle successfully avoided the obstacle by moving backwards. Fig. B illustrates the trajectory shaped with $a = 7.5$, with the obstacle relocated to $(11.35,\ 4.25)$ (m) Figs. C and D display trajectories shaped with $a = 7.5$, and obstacles at $(11.35,\ 4.25)$ (m) and $(7.5,\ 2.25)$ (m), and, $(11.35,\ 4.25)$ (m) and $(9.32,\ 2.44)$ (m), respectively. In all figures, the estimated positions of the tractor, first trailer, and second trailer are represented by yellow, blue, and red lines, respectively.

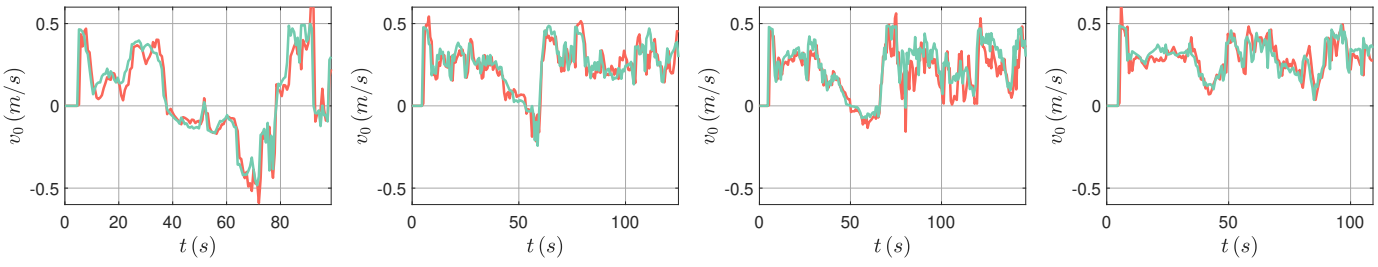

Fig. 11: Linear velocity profiles for one trial of Field Experiment 2 are depicted in Figs. A-D. Each figure corresponds to a different trial shown in Figure 10, with Fig. A corresponding to one trial from Figure 10-A, Fig. B to Figure 10-B, and so forth. In all these figures, the green line represents the velocity computed by the formulation in Eq. (11), while the red line represents the measured linear velocity.

the dynamic obstacle, a legged robot Unitree Go1 (refer to Figure 8) was manually controlled with a joystick to move in proximity to the vehicle as it follows the nominal path. Figure 12 displays the results of 5 trials in this field experiment, featuring 1 static obstacle and 1 dynamic obstacle. The figure illustrates the temporal evolution of the vehicle and the dynamic obstacle. The scatter plot in Figure 13 shows the evolution of the position of the tractor (yellow), first trailer (blue), last trailer (red) and the moving obstacle (green), where the vehicle successfully evades the dynamic obstacle.

V. Discussion

Although we successfully implemented autonomous navigation of generalised N-trailer vehicles in the presence of static and dynamic obstacles, further improvement and research are required.

The Global Path Planner Module (GPPM) generates a free-of-obstacles nominal trajectory, while the Local Path Planner Module (LPPM) generates an auxiliary reference when the original one is outside the set $\mathrm{proj}_i(Q_{N_c}^{\Omega_k})$. Further research is needed to establish the conditions under which the LPPM, once activated, will return to standby mode and transfer control back to the GPPM. While the use of an LPPM guarantees the

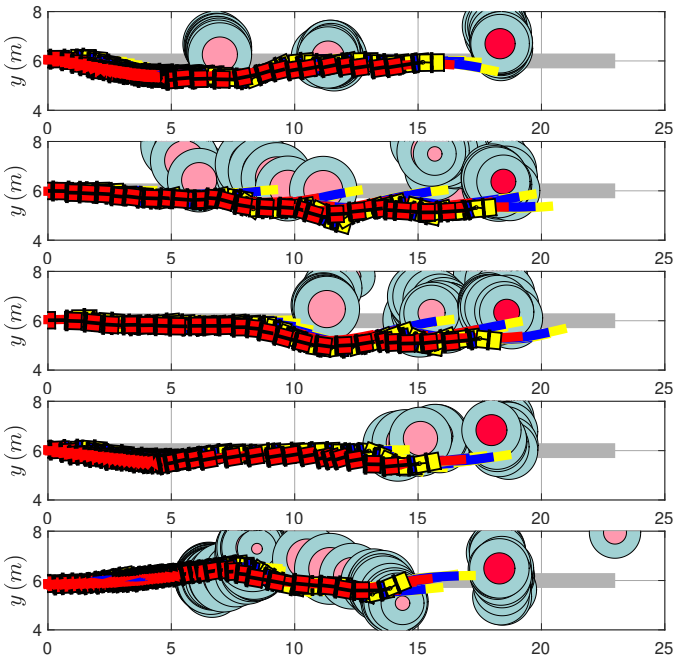

Fig. 12: Field Experiment 3 involves both static and dynamic obstacles. The dynamic obstacle is a legged robot controlled with a joystick. The series of five trials is presented from top to bottom, where the legged robot is directed to traverse various regions. In each Figure, the vehicle and obstacles are plotted for each sampling instance. Due to the vehicle and dynamic obstacle traversing the same coordinates at different time instants, an apparent intersection is depicted.

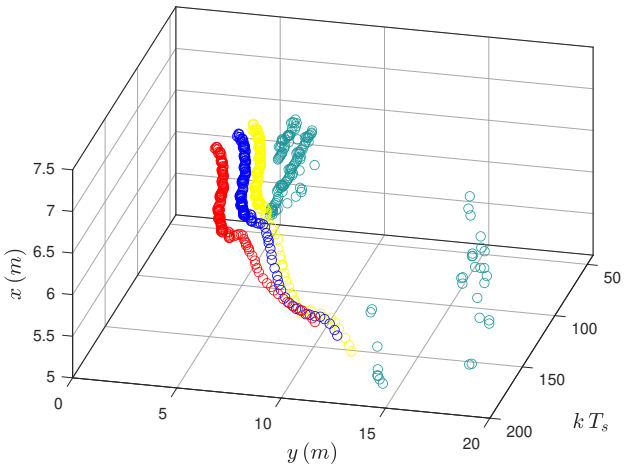

Fig. 13: Scatter plot of the position of the tractor (yellow), first trailer (blue) and last trailer (red) of the vehicle and the dynamic obstacle (green) for the fifth trial of experiment number 3.

stability of the closed-loop system, it is crucial that it remains active only for a short period, ensuring the auxiliary reference converges in finite time to the original one.

The constraints involved in the optimisation problem given by Eq. (16), are derived from heuristic rules. The mechanism for generating auxiliary references should be closely related to the manoeuvrability of the chained vehicle, necessitating further research, especially for large values of N_t.

Regarding the predictive controller, accuracy and performance can be improved by using the dynamic model of the vehicle instead of the kinematic model employed in this work. Additionally, incorporating skid-slip and external disturbances into the model is essential to address challenges encountered in rough terrains, such as those in agricultural and mining environments.

In terms of obstacle handling, our current implementation can manage a limited number of simultaneous obstacles. For efficient real-time implementation, the structure of the optimisation problem to be solved at each sampling time must remain time-invariant. Therefore, handling an arbitrary and potentially large number of simultaneous static and dynamic obstacles, along with ensuring efficient real-time implementation, requires further investigation.

The use of Gaussian functions was demonstrated to be efficient in avoiding collisions with obstacles. However, the use of more peaky functions, such as loss functions, may allow for navigating closer to obstacles without increasing the risk of collision. In addition, how to include non-convex constraints without loose the benefits of solving an optimisation problem on a convex space deserves further investigation.

VI. Conclusions and Future Works

In this study, we presented a novel Nonlinear Model Predictive Path-Following Controller (NMPPFC) tailored for generalised N-trailer vehicles navigating amidst dynamic obstacles. We characterised detected obstacles as Gaussian functions, with their radii determined by adding the vehicle dimensions and a safety margin to the obstacle size. The careful choice of the Gaussian amplitude is essential, as the regions occupied by these functions remain feasible for the NMPPFC.

Our approach has been validated through a series of simulated and field experiments, demonstrating its efficacy in obstacle avoidance in challenging scenarios with up to 12 obstacles. Additionally, our approach alleviates the computational burden compared to a state-of-the-art method that handles obstacles as hard constraints. The performance in collision avoidance is also superior to our method in comparison with the competitor.

Future endeavours will involve addressing the issues discussed in the previous section, such as the explicit inclusion of the vehicle's manoeuvrability and enhancing the accuracy of our model by substituting the kinematic vehicle model with a dynamic counterpart. We will also explore incorporating skid-slip and external disturbances into the model to improve performance in rough terrains. Furthermore, we aim to refine the Local Path Planner Module (LPPM) activation/deactivation conditions and investigate more efficient obstacle-handling techniques to support a larger number of simultaneous obstacles. Finally, obstacle detection can be improved by incorporating a point cloud processing technique from the state-of-the-art.

Acknowledgments

This work was partially supported by the Advanced Center for Electrical and Electronic Engineering, AC3E, ANID basal

project FB0008. Since the authors are not native English speakers, the grammar was checked and enhanced using ChatGPT 4.0 (06/28/24, https://chat.openai.com/).

Nestor Nahuel Deniz received the Bachelor's degree in electronic engineering from the National Technological University in Parana, Entre Rios, Argentina. He obtained his PhD from the National University of the Litoral in Argentina. Currently, he serves as a postdoctoral researcher at the Advanced Centre for Electrical and Electronic Engineering (AC3E) in Valparaiso, Chile. His current research interests centre on robotics, model-based estimation and control techniques, autonomous navigation of vehicles, data-driven control, neural networks, and Koopman operator's theory.

Fernando Auat Cheein (Senior Member, IEEE) received the Bachelor degree in electronic engineering from Universidad Nacional de Tucuman, Argentina, in 2002, and the M.Sc. and Ph.D. degrees in Engineering from Universidad Nacional de San Juan, Argentina, in 2005 and 2009, respectively. He is currently an Associate Professor at the School of Engineering and Physical Sciences, Heriot-Watt University and part of the National Robotarium, Edinburgh, UK. His research is focused on autonomous and intelligent vehicles, robotics and perception in agriculture, motion planning and control, and efficient navigation strategies. Prof. Auat Cheein is currently an Associate Editor of several journals.