automation

MDPI

Article

Optimizing Unmanned Air–Ground Vehicle Maneuvers Using Nonlinear Model Predictive Control and Moving Horizon Estimation

Alessandra Elisa Sindi Morando [1,2,*], Alessandro Bozzi [1], Simone Graffione [1], Roberto Sacile [1] and Enrico Zero [1]

[1] Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genova, 16100 Genova, Italy; alessandro.bozzi@edu.unige.it (A.B.); simone.graffione@edu.unige.it (S.G.); roberto.sacile@unige.it (R.S.); enrico.zero@dibris.unige.it (E.Z.)

[2] Heudiasyc (Heuristics and Diagnosis of Complex Systems) CNRS Laboratory of the Université de Technologie de Compiègne, 60200 Compiègne, France

* Correspondence: alessandra.elisa.sindi.morando@edu.unige.it

Abstract: In this paper, Nonlinear Model Predictive Control (NMPC) and Nonlinear Moving Horizon Estimator (NMHE) are combined to control, in a distributed way, a heterogeneous fleet composed of a steering car and a quadcopter. In particular, the ground vehicle in the role of the leader communicates its one-step future position to the drone, which keeps the formation along the desired trajectory. Inequality constraints are introduced in a switching control fashion to the leader's NMPC formulation to avoid obstacles. In the literature, few works using NMPC and NMHE deal with these two vehicles together. Moreover, the presented scheme can tackle noisy, partial, and missing measurements of the agents' state. Results show that the ground car can avoid detected obstacles, keeping the tracking errors of both robots in the order of a few centimeters, thanks to trustworthy NMHE estimates and NMPC predictions.

Keywords: unmanned ground vehicle; unmanned aerial vehicle; nonlinear model predictive control; nonlinear moving horizon estimation

Citation: Morando, A.E.S.; Bozzi, A.; Graffione, S.; Sacile, R.; Zero, E. Optimizing Unmanned Air–Ground Vehicle Maneuvers Using Nonlinear Model Predictive Control and Moving Horizon Estimation. *Automation* **2024**, *5*, 324–342. https://doi.org/10.3390/automation5030020

Academic Editor: Simon Tomažič

Received: 11 June 2024
Revised: 15 July 2024
Accepted: 25 July 2024
Published: 30 July 2024

1. Introduction

Multiple robot systems have become widespread, as they can perform hard and possibly dangerous tasks for humans while improving efficiency and productivity. Among the plethora of applications, there are industry, agriculture, construction, and emergency rescue [1].

Among the different kinds of collaboration, hybrid teams can tackle more complex tasks requiring different capabilities, notably air–ground cooperation. On the one hand, Unmanned Aerial Vehicles (UAVs) have long-range sensors on board, making them suitable for monitoring from the sky. On the other side, Unmanned Ground Vehicles (UGVs) have short-range sensors and high load and computational capacities. The main issues are motion control, formation control, and collision avoidance. Vinicius et al. [2] proposed a light path following a virtual structure-based controller for package delivery. As the UAV flies over it, the UGV has to follow a desired trajectory while avoiding obstacles.

A PI optimal control theory is presented in [3], and results show that the proposed scheme can overcome input and model disturbances and linearization biases. Again considering air–ground cooperation, ref. [4] presents a leader–follower architecture, where the quadcopter has to reach the desired waypoints and send its position to the unicycle, which uses these data as the desired position. In addition, the ground agent should avoid obstacles around it.

Classical control methods assume complete knowledge of the system state without data loss. When agents are not tracked, the control action cannot be computed, and the

robots must halt. Nonlinear Model Predictive Control (NMPC) provides a solution to this issue. In the absence of measurements, NMPC can apply the optimal control trajectory computed in the previous step until updated data become available. The key idea is to use an internal nonlinear model to predict the system's future response. Due to its simplicity, MPC has been widely used in various applications, with numerous studies experimentally validating its superiority over classical linear controllers. In [5], the authors compared the performance of PID, LQR and MPC in controlling a drone. Both simulations and real experiments with a Parrot mini-drone were carried out. It turns out that MPC outperforms other methods, ensuring robustness, stability, and small errors in terms of maximum pitch deviation. Al-Saoudi et al. [6] carried out a similar study comparing model-based controllers such as PID, fuzzy logic, and MPC with an ANFIS-based controller. From the MATLAB/SIMULINK R2022a tests, it is shown that the MPC is superior, as it can handle input constraints and has lower overshoot and response time compared to other techniques. Taking into account a separation clutch system in the slipping state, ref. [7] compares PI-PD and LQR methods with a scheme composed of multiple MPCs but again this last turns out to be the best in terms of performance. Always in the field of autonomous driving, the paper [8] studies and evaluates in terms of tracking error different methods (Pure Pursuite, Stanley, LQR and MPC) for the trajectory tracking mission of a steering car. In both tests, MPC provides better accuracy both for the eight shape and the circle path. While NMPC has been consolidated over the years for controlling UGVs, leading to embedded implementations [9,10], it has been applied to UAVs only recently. Indeed, with the advent of more powerful microprocessors and efficient nonlinear solvers, a real-time application of NMPC for small drones is now possible [11].

However, one of the primary limitations of NMPC is the assumption that the complete state is measurable. To overcome this, NMPC is often paired with nonlinear estimators, such as the Nonlinear Moving Horizon Estimator (NMHE) [12]. NMHE addresses the dual problem of NMPC by providing state estimates that incorporate constraints and nonlinearities without assuming a specific error distribution. This pairing allows NMPC to function effectively even when full state information is not directly measurable. These two complementary state-of-the-art tools, when combined, can accurately control and estimate complex underactuated systems. In [13], the authors propose an NMHE-NMPC scheme to address the path-following problem of generalized N-vehicles under noisy measurements and disturbances. To evaluate its performance, simulations and real-world experiments have been performed to show its accuracy and applicability to real-time scenarios. A combination of both techniques is presented in [14] to control a selective catalytic reduction process for reducing engine emissions. In the context of UAVs, ref. [15] proposes a new NMPC method for UAV navigation that can deal with dynamic obstacles by predicting their movement and using a fast solver to find safe trajectories in real time, while [16] presents a nonlinear fault-tolerant control system for a quadcopter. This approach combines NMPC and NMHE to maintain trajectory control even in the presence of rotor failures. In the worst-case scenario where all motors are damaged, the NMHE estimates allow the NMPC to effectively stabilize and accurately follow the reference trajectory.

This work addresses the UAV-UGV formation and path-tracking problem. In particular, a distributed leader–follower NMPC-NMHE scheme is defined. The main novelties and contributions of this paper are the following:

(a) Although promising results have been obtained in various applications, NMPC and NMHE have not been used together for air–ground cooperation.
(b) To the best of the authors' knowledge, these two receiving horizon techniques have not been used to control a heterogeneous fleet consisting of a steering car and a quadcopter. In fact, most current work considers unicycles as ground vehicles.
(c) It is worth noting that the proposed solution deals with partially noisy state measurements and is, in principle, robust to information loss.

The paper unfolds in three key sections. Section 2 outlines the methods and definitions devised to tackle the focal problem under examination. Moving forward, Section 3 consolidates the numerical simulation results. Ultimately, Section 4 encapsulates the conclusions, emphasizing future avenues for development.

2. Materials and Methods

The following sections will introduce the mathematical models employed in the problem formulation. Even if the UAV and the UGV present different dynamics, an NMPC and an NMHE have been developed for both agents. Moreover, it is assumed that the steering car communicates to the drone the one-step prediction of its state.

2.1. UAV and UGV Models

For describing the drone dynamics, the model defined in [17] is used.

$$\begin{cases} \dot{x_1} &= x_2 \\ \dot{x_2} &= -u_1 \sin(\theta_1)/M \\ \dot{y_1} &= y_2 \\ \dot{y_2} &= u_1 \cos(\theta_1) \sin(\phi_1)/M \\ \dot{z_1} &= z_2 \\ \dot{z_2} &= u_1 \cos(\theta_1) \cos(\phi_1)/M - g \\ \dot{\theta_1} &= \theta_2 \\ \dot{\theta_2} &= u_2 \\ \dot{\phi_1} &= \phi_2 \\ \dot{\phi_2} &= u_3 \\ \dot{\psi_1} &= \psi_2 \\ \dot{\psi_2} &= u_4 \end{cases} \tag{1}$$

The state variables are defined according to Figure 1. Hereinafter, let the state and control variables of the UAV at time t be denoted as:

$$\boldsymbol{x}_{UAV} = \begin{bmatrix} x_1 & x_2 & y_1 & y_2 & z_1 & z_2 & \theta_1 & \theta_2 & \phi_1 & \phi_2 & \psi_1 & \psi_2 \end{bmatrix}^T \tag{2}$$

$$\boldsymbol{u}_{UAV} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T \tag{3}$$

and rewrite Equation (1) in a vectorial form

$$\dot{\boldsymbol{x}}_{UAV} = f_{UAV}(\boldsymbol{x}_{UAV}, \boldsymbol{u}_{UAV}) \tag{4}$$

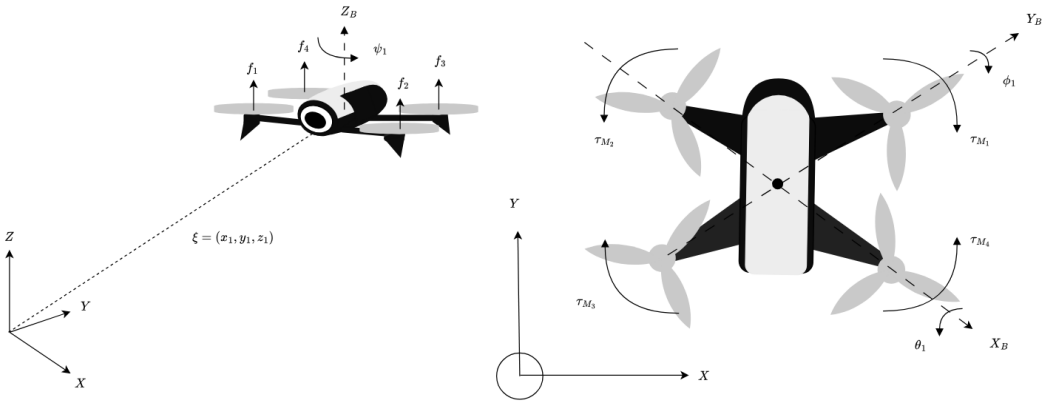

Figure 1. The quadcopter state consists of its position, translational velocity (both in the global reference frame), orientation, and rotational velocities (expressed in terms of Euler angles and in the body frame). The dynamics are obtained using the Euler–Lagrange approach.

To study the steering car, the Ackermann model is used [18]. All the state quantities of the ground vehicle of the ground robot are depicted in Figure 2. The control variables are the steering angle $w_1(t)$ and the forward acceleration $w_2(t)$. The evolution model is

$$\begin{cases} \dot{x} &= v\cos(\theta) \\ \dot{y} &= v\sin(\theta) \\ \dot{\theta} &= (v/L)\tan(w_1) \\ \dot{v} &= w_2 \end{cases} \tag{5}$$

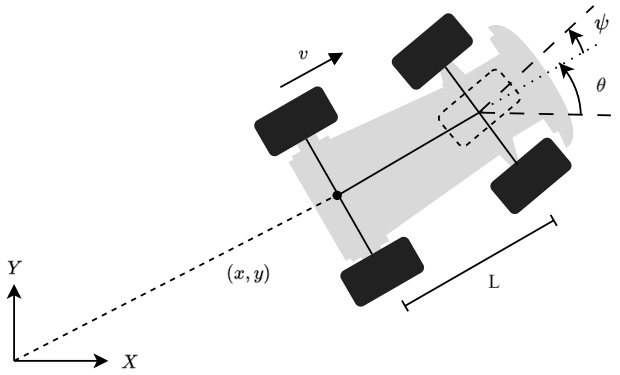

Figure 2. UGV Ackermann's dynamic model with corresponding state and control variables.

The remainder of this paper will make use of the following notation:

$$\boldsymbol{x}_{UGV} = \begin{bmatrix} x & y & \theta & v \end{bmatrix}^T \tag{6}$$

$$\boldsymbol{u}_{UGV} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T \tag{7}$$

Thus, Equation (5) becomes

$$\dot{\boldsymbol{x}}_{UGV} = f_{UGV}(\boldsymbol{x}_{UGV}, \boldsymbol{u}_{UGV}) \tag{8}$$

2.2. UAV NMPC

The drone's NMPC aims to find the optimal control input to be applied by solving an optimization problem. Given the current estimated quadcopter's state $\hat{\boldsymbol{x}}_{UAV}(k)$, the objective is to minimize the deviations from the reference $r_{UAV}(k)$ over a prediction horizon H_p weighted by $\mathcal{Q}_{UAV}$. To predict the future behavior of the UAV, the discrete-time evolution f_{UAV} is used, obtained by discretizing Equation (4) with sampling time T_s and applying a trapezoidal rule:

$$\min \quad \sum_{i=1}^{H_p} \|\boldsymbol{x}_{UAV}(k+i|k) - \boldsymbol{r}_{UAV}(k)\|^2_{\mathcal{Q}_{UAV}} \tag{9a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{UAV}(k|k) = \hat{\boldsymbol{x}}_{UAV}(k) \tag{9b}$$

$$\boldsymbol{x}_{UAV}(k+i+1|k) = f_{UAV}(\boldsymbol{x}_{UAV}(k+i|k), \boldsymbol{u}_{UAV}(k+i|k)) \tag{9c}$$

$$x^{min}_{UAV} \leq \boldsymbol{x}_{UAV}(k+i|k) \leq x^{MAX}_{UAV} \tag{9d}$$

The decision variables are the state and the input trajectory, respectively $x_{UAV}(k+i|k)$ and $u_{UAV}(k+i|k)$, over the time window. To reduce the computational effort, it is assumed that after H_u steps, the control remains constant. Regarding the reference, the drone should track the future position of the leader over the prediction horizon. Here, the trajectory is

approximated with the future pose at the next step $k+1$ obtained by the car's NMPC (see Section 2.5):

$$\begin{aligned} \boldsymbol{r}_{UAV}(k) = \big[& x(k+1|k) \quad 0 \quad y(k+1|k) \quad 0 \quad z_{1,Ref}(k) \quad 0 \\ & 0 \quad 0 \quad 0 \quad 0 \quad \theta(k+1|k) \quad 0 \big]^T \end{aligned} \tag{10}$$

Constraint Equation (9d) represents the structural limitations inherent to the drone's design.

Conceptually, it is important to verify that the controller is stable and provides a control law that does not diverge, regardless of the system's objective. Although the proof of stability is not included in this work, readers interested in a generalization may refer to Theorem 3 of [19], which is largely similar to the proposed theorem in this work, with the exception of minor constraints that are reasonably assumed not to affect the system's stability. This is supported by the results in Section 3, which empirically confirm the validity of the control architecture for the proposed scenarios.

2.3. MHE for the UAV

The defined NMPC assumes that all state variables are known. However, some quantities cannot be measured directly, necessitating the introduction of an estimator, such as the MHE, which can deal with the exact nonlinear model and state and control boundaries. Taking into account a set of past noisy measurements,

$$\boldsymbol{y}(k-i) \quad i = 0, \cdots, H_e \tag{11}$$

and control inputs

$$\boldsymbol{u}(k-i) \quad i = 0, \cdots, H_e \tag{12}$$

over an estimation horizon H_e, the estimated state trajectory over the time window

$$\hat{\boldsymbol{x}}(k-i) \quad i = 0, \cdots, H_e \tag{13}$$

is obtained by solving a constrained nonlinear optimization problem that penalizes the variations of the predicted measurement from the actual past values. Once the sequence of estimates is obtained, only the final value is retained, and the estimation horizon is shifted forward to repeat the computations at the next instant $k+1$. It is important to note that although the focus here is on nonmeasurable quantities, this technique can also be employed to estimate unknown parameters.

For a generic unmanned vehicle, the equations are

$$\min \quad \sum_{i=1}^{H_e} \|\boldsymbol{y}(k-i) - h(\hat{\boldsymbol{x}}(k-i), \boldsymbol{u}(k-i))\|_{\mathcal{V}}^2 \tag{14a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{x}}(k-i+1) = f(\hat{\boldsymbol{x}}(k-i), \boldsymbol{u}(k-i)) \tag{14b}$$

$$\boldsymbol{x}_{UV}^{min} \leq \hat{\boldsymbol{x}}(k-i) \leq \boldsymbol{x}_{UV}^{MAX} \tag{14c}$$

where f and h are, respectively, the evolution and the measurement models. When considering a Gaussian measurement noise

$$\nu(t) \sim \mathcal{N}(0_{n_y \times 1}, \Sigma_\nu) \quad \Sigma_\nu = \text{diag}(\sigma_1^2 \quad \cdots \quad \sigma_{n_y}^2) \tag{15}$$

the weighting matrix $\mathcal{V}$ is chosen as

$$\mathcal{V} = \left(\sqrt{\Sigma_\nu}\right)^{-1} = \text{diag}(1/\sigma_1 \quad \cdots \quad 1/\sigma_{n_y}) \tag{16}$$

In the case of the quadcopter, $\hat{\boldsymbol{x}}(k-i)$ and $\boldsymbol{u}(k-i)$ are, respectively, the estimate of the state $\boldsymbol{x}_{UAV}$ and the control $\boldsymbol{u}_{UAV}$ at instant $k-i$, $i = 0, \cdots, H_e$. It is reasonable to assume that a motion capture system can measure the pose of the robot with high accuracy. Although the Euler angles are defined in the body frame, a rotation matrix from the global reference frame can be defined. However, although this aspect is not addressed in the current context, measurement noise should be considered when developing an embedded system:

$$\nu_{UAV}(t) \sim \mathcal{N}(0_{6\times 1}, \Sigma_{\nu_{UAV}}) \tag{17a}$$

$$\Sigma_{\nu_{UAV}} = \text{diag}(\begin{bmatrix}\sigma_{x_1}^2 & \sigma_{y_1}^2 & \sigma_{z_1}^2 & \sigma_{\theta_1}^2 & \sigma_{\phi_1}^2 & \sigma_{\psi_1}^2\end{bmatrix}) \tag{17b}$$

$$\mathcal{V}_{UAV} = \text{diag}(\begin{bmatrix}1/\sigma_{x_1} & 1/\sigma_{y_1} & 1/\sigma_{z_1} & 1/\sigma_{\theta_1} & 1/\sigma_{\phi_1} & 1/\sigma_{\psi_1}\end{bmatrix}) \tag{18}$$

$$h_{UAV}(\hat{\boldsymbol{x}}(k-i), \boldsymbol{u}(k-i)) = C_{UAV}\,\hat{\boldsymbol{x}}(k-i) \tag{19a}$$

$$C_{UAV} \in \mathbb{R}^{6\times 12} \quad \text{s.t.} \tag{19b}$$

$$C_{UAV}[i,j] = \begin{cases} 1 & \text{if } \begin{matrix} i = 1, \cdots, 6 \\ j = 2i-1 \end{matrix} \\ 0 & \text{otherwise} \end{cases} \tag{19c}$$

The discrete-time evolution model used in the UAV MHE formulation is obtained through the Runge–Kutta (RK4) method:

$$k_1 = f_{UAV}(\hat{\boldsymbol{x}}_{UAV}(k-i), \boldsymbol{u}_{UAV}(k-i)) \tag{20a}$$

$$k_2 = f_{UAV}(\hat{\boldsymbol{x}}_{UAV}(k-i) + (T_e/2)\,k_1, \boldsymbol{u}_{UAV}(k-i)) \tag{20b}$$

$$k_3 = f_{UAV}(\hat{\boldsymbol{x}}_{UAV}(k-i) + (T_e/2)\,k_2, \boldsymbol{u}_{UAV}(k-i)) \tag{20c}$$

$$k_4 = f_{UAV}(\hat{\boldsymbol{x}}_{UAV}(k-i) + (T_e/2)\,k_3, \boldsymbol{u}_{UAV}(k-i)) \tag{20d}$$

$$\begin{aligned}\hat{\boldsymbol{x}}_{UAV}(k-i+1) = \hat{\boldsymbol{x}}_{UAV}(k-i) + \\ + (T_e/6)\,(k_1 + 2k_2 + 2k_3 + k_4)\end{aligned} \tag{20e}$$

with T_e the estimation time step. Finally, the constraint Equation (14c) becomes

$$\boldsymbol{x}_{UAV}^{min} \leq \hat{\boldsymbol{x}}(k-i) \leq \boldsymbol{x}_{UAV}^{MAX} \tag{21}$$

2.4. MHE for the UGV

Considering the UGV, an MHE should be introduced to estimate the forward velocity, which cannot be measured directly by the motion capture system. In particular,

$$\nu_{UGV}(t) \sim \mathcal{N}(0_{3\times 1}, \Sigma_{\nu_{UGV}}) \tag{22a}$$

$$\Sigma_{\nu_{UGV}} = \text{diag}(\begin{bmatrix}\sigma_x^2 & \sigma_y^2 & \sigma_\theta^2\end{bmatrix}) \tag{22b}$$

$$\mathcal{V}_{UGV} = \text{diag}(\begin{bmatrix}1/\sigma_x & 1/\sigma_y & 1/\sigma_\theta\end{bmatrix}) \tag{23}$$

$$h_{UGV}(\hat{\boldsymbol{x}}(k-i), \boldsymbol{u}(k-i)) = C_{UGV}\hat{\boldsymbol{x}}(k-i) \tag{24a}$$

$$C_{UGV} = I_{3\times 4} \tag{24b}$$

Similarly, the RK4 method has been employed for the discretization of state differential equations in the ground agent prediction model Equation (8).

2.5. UGV NMPC with Obstacle Avoidance

To control the steering car and avoid obstacles in the meantime, a NMPC is used. Assume that the UGV has remote sensing technology on board, such as LIDAR, with a 360° full scanning range up to d_{LIDAR} meters. Once an obstacle is detected in the sensor's field, the robot should keep a safe distance d_{Safe} from it as depicted in Figure 3.

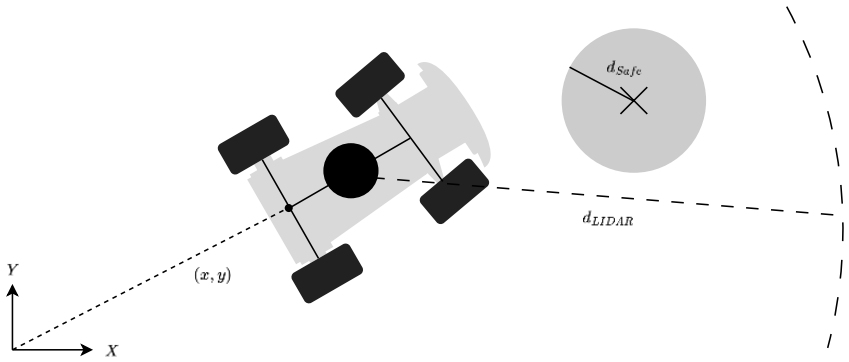

Figure 3. Obstacles are detected up to a distance d_{LIDAR}, and the UGV should keep a safe distance d_{Safe}.

The mathematical formulation of the steering car's NMPC in the absence of obstacles is

$$\begin{aligned}
\min \quad & \sum_{i=1}^{H_p} \|\boldsymbol{x}_{UGV}(k+i|k) - \boldsymbol{r}_{UGV}(k+i)\|^2_{\mathcal{Q}_{UGV}} + \sum_{i=0}^{H_u-1} \|\boldsymbol{u}_{UGV}(k+i|k)\|^2_{\mathcal{R}} + \\
& + \sum_{i=0}^{H_u-1} \|\Delta\boldsymbol{u}_{UGV}(k+i|k)\|^2_{\mathcal{R}_\Delta} && \text{(25a)} \\
\text{s.t.} \quad & \boldsymbol{x}_{UGV}(k|k) = \hat{\boldsymbol{x}}_{UGV}(k) && \text{(25b)} \\
& \boldsymbol{x}_{UGV}(k+i+1|k) = f_{UGV}(\boldsymbol{x}_{UGV}(k+i|k), \boldsymbol{u}_{UGV}(k+i|k)) && \text{(25c)} \\
& 0 \leq \boldsymbol{e}_{xy}(k+H_p|k)^T \boldsymbol{e}_{xy}(k+H_p|k) \leq (d_{Safe} + \varepsilon)^2 && \text{(25d)} \\
& x^{min}_{UGV} \leq \boldsymbol{x}_{UGV}(k+i|k) \leq x^{MAX}_{UGV} && \text{(25e)} \\
& u^{min}_{UGV} \leq \boldsymbol{u}_{UGV}(k+i|k) \leq u^{MAX}_{UGV} && \text{(25f)}
\end{aligned}$$

where, for readability, the input variation is denoted as

$$\Delta\boldsymbol{u}_{UGV}(k+i+1|k) = \boldsymbol{u}_{UGV}(k+i+1|k) - \boldsymbol{u}_{UGV}(k+i|k) \quad \text{(26a)}$$

$$\Delta\boldsymbol{u}_{UGV}(k|k) = \boldsymbol{u}_{UGV}(k|k) - \boldsymbol{u}_{UGV}(k-1) \quad \text{(26b)}$$

with $\boldsymbol{u}_{UGV}(k-1)$ the control input at instant k. Let us see in detail the formulation which is similar to the one in Section 2.2. In the objective function, the control input and its variations are also penalized respectively with weights $\mathcal{R}$ and $\mathcal{R}_\Delta$. Secondly, the decision variables are the future evolution of the state

$$\boldsymbol{x}_{UGV}(k+i|k) \quad i = 0, \ldots, H_p \quad (27)$$

and the control trajectory

$$\boldsymbol{u}_{UGV}(k+i|k) \quad i = 0, \ldots, H_u - 1 \quad (28)$$

For the ground vehicle, a reference trajectory (which varies along the prediction horizon) is defined for all state variables

$$\boldsymbol{r}_{UGV}(k+i) = \begin{bmatrix} x_{Ref}(k+i) & y_{Ref}(k+i) & \theta_{Ref}(k+i) & v_{Ref}(k+i) \end{bmatrix}^T \tag{29}$$

Concerning the constraints, $\hat{\boldsymbol{x}}_{UGV}(k)$ is the MHE estimate of the ground vehicle state. In Equation (25c), f_{UGV} represents the evolution model used in the UGV estimator. Terminal region constraints Equation (25d) have been added to force the vehicle to stay in a neighborhood of the reference at the end of the prediction horizon. In Equation (25d), $\boldsymbol{e}_{xy}(k+H_p|k)$ is the (x,y)-tracking error at instant $k+H_p$:

$$\boldsymbol{e}_{xy}(k+H_p|k) = \begin{bmatrix} x(k+H_p|k) - r_x(k+H_p) \\ y(k+H_p|k) - r_y(k+H_p) \end{bmatrix} \tag{30}$$

Note that the (x,y)-terminal region has been chosen as a circular area of size $d_{Safe}+\varepsilon$, as, in theory, the reference could coincide with the obstacle as in the case study considered in Section 3. Bounds are defined for the state and the control.

Up to now, Equation (25) has considered the scenario in which no obstruction is found by the onboard sensor. The approach to address obstacle avoidance closely mirrors the one of Sani et al. [20]. If a barrier is encountered, the controller is switched, and a further family of constraints is added to Equation (25):

$$d_{Obs}(k+i|k)^T d_{Obs}(k+i|k) \geq (d_{Safe}+\varepsilon)^2 \tag{31}$$

with

$$d_{Obs}(k+i|k) = \begin{bmatrix} x(k+i|k) - x_{Obs}(k) \\ y(k+i|k) - y_{Obs}(k) \end{bmatrix} \tag{32}$$

and (x_{Obs}, y_{Obs}) is the position of the detected obstacle. The margin $\varepsilon > 0$ is added such that the safety distance is always respected.

3. Results

To validate the proposed solution, various simulations were conducted in MATLAB/Simulink, altering the reference trajectories, the number of obstacles, and their positions. In particular, to implement MHE and the switching NMPC of the UGV, the CasADi framework is used as in [10,20]. CasADi [21] is an open-source tool that provides different facilities useful in academic and industrial applications, notably the Internal Point Optimizer to solve nonlinear programming problems efficiently.

The robots parameters are set to $L = 0.14\,\text{m}$ and $M = 0.50\,\text{kg}$ in accordance with the values of the JetRacer race car and the Parrot Bebop 2.0 drone. It is assumed that the agent can detect barriers up to $d_{LIDAR} = 5\,\text{m}$, in line with the datasheet of the RPLIDAR A1 included with the JetRacer ROS Kit. The safety distance for the UGV is set to $d_{Safe} = 0.25\,\text{m}$ and $\varepsilon = 0.10\,\text{m}$. At the beginning of the simulations, the states of both robots are

$$\boldsymbol{x}_{UGV}(0) = \begin{bmatrix} 2\,\text{m} & 0\,\text{m} & \pi/2\,\text{rad} & 0\,\text{m}\,\text{s}^{-1} \end{bmatrix}^T \tag{33}$$

$$\begin{aligned} \boldsymbol{x}_{UAV}(0) = \big[& 2\,\text{m} \quad 0\,\text{m}\,\text{s}^{-1} \quad 0\,\text{m} \quad 0\,\text{m}\,\text{s}^{-1} \quad 1\,\text{m} \quad 0\,\text{m}\,\text{s}^{-1} \\ & 0\,\text{rad} \quad 0\,\text{rad}\,\text{s}^{-1} \quad 0\,\text{rad} \quad 0\,\text{rad}\,\text{s}^{-1} \quad \pi/2\,\text{rad} \quad 0\,\text{rad}\,\text{s}^{-1} \big]^T \end{aligned} \tag{34}$$

Concerning the structural boundaries of the quadcopter, the maximum tilt angle, vertical speed, and rotational speed around Z_B are

$$|\theta_1(t)| \leq 30\pi/180\,\text{rad} \tag{35a}$$

$$|\phi_1(t)| \leq 30\pi/180\,\text{rad} \tag{35b}$$

$$|z_1(t)| \leq 1\,\text{m}\,\text{s}^{-1} \tag{35c}$$

$$|\psi_2(t)| \leq 100\pi/180\,\mathrm{rad\,s^{-1}} \tag{35d}$$

On the other hand, for the steering car's state, only the forward velocity is bounded

$$|v(t)| \leq 0.30\,\mathrm{m\,s^{-1}} \tag{36}$$

while for its control inputs

$$-0.30\,\mathrm{rad} \leq w_1(t) \leq 0.80\,\mathrm{rad} \tag{37a}$$

$$|w_2(t)| \leq 1\,\mathrm{m\,s^{-2}} \tag{37b}$$

Regarding the NMPC parameters, $T_s = 0.1\,\mathrm{s}$ is set for both $H_p = 10$ and $H_u = 5$, while for the weights

$$\mathcal{Q}_{UGV} = 1000\,\mathrm{diag}(\begin{bmatrix}20 & 20 & 1 & 1\end{bmatrix}) \tag{38a}$$

$$\mathcal{Q}_{UAV} = 100\,\mathrm{diag}(\begin{bmatrix}1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}) \tag{38b}$$

$$\mathcal{R} = 10\,I_{2\times 2} \quad \mathcal{R}_\Delta = 100\,I_{2\times 2} \tag{38c}$$

Concerning the estimators, $H_e = 5$, $T_e = 0.1\,\mathrm{s}$ is chosen, and

$$\begin{gathered}\sigma_{x_1} = \sigma_{y_1} = \sigma_{z_1} = \sigma_x = \sigma_y = 0.02 \times 10^{-3}\,\mathrm{m} \\ \sigma_{\theta_1} = \sigma_{\phi_1} = \sigma_{\psi_1} = \sigma_\theta = 0.1\,\mathrm{rad}\end{gathered} \tag{39}$$

as defined in the Optitrack's specification.

For all the cases studies, the quantitative analysis is split considering the robots separately, as, indeed, the controller is distributed. Let us focus first on the UGV. To investigate the NMPC performance, the Root Mean Square Error (RMSE) is considered for the tracking:

$$RMSE^{\boldsymbol{r}_{UGV}} = \sqrt{\mathrm{E}\left[(\boldsymbol{x}_{UGV}(k) - \boldsymbol{r}_{UGV}(k))^2\right]} \tag{40}$$

and for the one-step prediction

$$RMSE^{\boldsymbol{y}_{+1}} = \sqrt{\mathrm{E}\left[(\boldsymbol{y}(k+1|k) - \boldsymbol{x}_{UGV}(k+1))^2\right]} \tag{41}$$

Regarding the MHE, the estimation's RMSE is considered as

$$RMSE^{\hat{\boldsymbol{x}}_{UGV}} = \sqrt{\mathrm{E}\left[(\hat{\boldsymbol{x}}_{UGV}(k) - \boldsymbol{x}_{UGV}(k))^2\right]} \tag{42}$$

Regarding the quadcopter tracking, it is important to distinguish the deviation from the reference used by the NMPC objective, where zeros corresponding to nontracked quantities are not considered:

$$\boldsymbol{r}_{UAV}(k) = \begin{bmatrix}x(k+1|k) & y(k+1|k) & z_{1,Ref}(k) & \theta(k+1|k)\end{bmatrix}^T \tag{43}$$

and the actual one

$$\boldsymbol{r}_{T,UAV}(k) = \begin{bmatrix}x_{Ref}(k) & y_{Ref}(k) & z_{1,Ref}(k) & \theta_{Ref}\end{bmatrix}^T \tag{44}$$

Depending on whether Equation (43) or Equation (44) is chosen, two tracking errors can be defined:

$$RMSE^{\boldsymbol{r}_{UAV}} = \sqrt{\mathrm{E}\left[(\boldsymbol{x}_{T,UAV}(k) - \boldsymbol{r}_{UAV}(k))^2\right]} \tag{45}$$

$$RMSE^{\boldsymbol{r}_{T,UAV}} = \sqrt{\mathrm{E}\left[(\boldsymbol{x}_{T,UAV}(k) - \boldsymbol{r}_{T,UAV}(k))^2\right]} \tag{46}$$

where $\boldsymbol{x}_{T,UAV}(k)$ is the subset of the tracked drone's state's components, i.e.,

$$\boldsymbol{x}_{T,UAV}(k) = \begin{bmatrix} x_1(k) & y_1(k) & z_1(k) & \psi_1(k) \end{bmatrix}^T \tag{47}$$

Finally, to evaluate the MHE performance, the metric below is considered

$$RMSE^{\hat{\boldsymbol{x}}_{UAV}} = \sqrt{\mathrm{E}\left[\left(\hat{\boldsymbol{x}}_{UAV}(k) - \boldsymbol{x}_{UAV}(k)\right)^2\right]} \tag{48}$$

3.1. Circle with One Obstacle

The first scenario lasts 2 min and considers a circular trajectory centered in the origin with radius of 2 m, described by the following equations:

$$\begin{cases} x_{Ref}(t) = 2\cos\left(\frac{2\pi}{60}\,t\right) \\ y_{Ref}(t) = 2\sin\left(\frac{2\pi}{60}\,t\right) \\ \theta_{Ref}(t) = \frac{2\pi}{60}\,t + \frac{\pi}{2} \\ v_{Ref}(t) = 2\,\frac{2\pi}{60} \end{cases} \tag{49}$$

A static obstacle is placed along the reference trajectory in $(x_{Obs}, y_{Obs}) = (-2\,\mathrm{m}, 0\,\mathrm{m})$. While the UGV should track the reference, the drone should follow the leader, keeping an altitude defined as

$$z_{1,Ref}(t) = \begin{cases} 2\,\mathrm{m} & \text{if} \quad t < 1\,\mathrm{min} \\ 3\,\mathrm{m} & \text{if} \quad t \geq 1\,\mathrm{min} \end{cases} \tag{50}$$

The aim is to demonstrate that even if the drone must increase its altitude to avoid an obstacle, the controller effectively manages this change in trajectory. It should be noted that the UAV NMPC formulation currently does not include constraints such as Equation (31). Figure 4 shows the trajectories of the two agents.

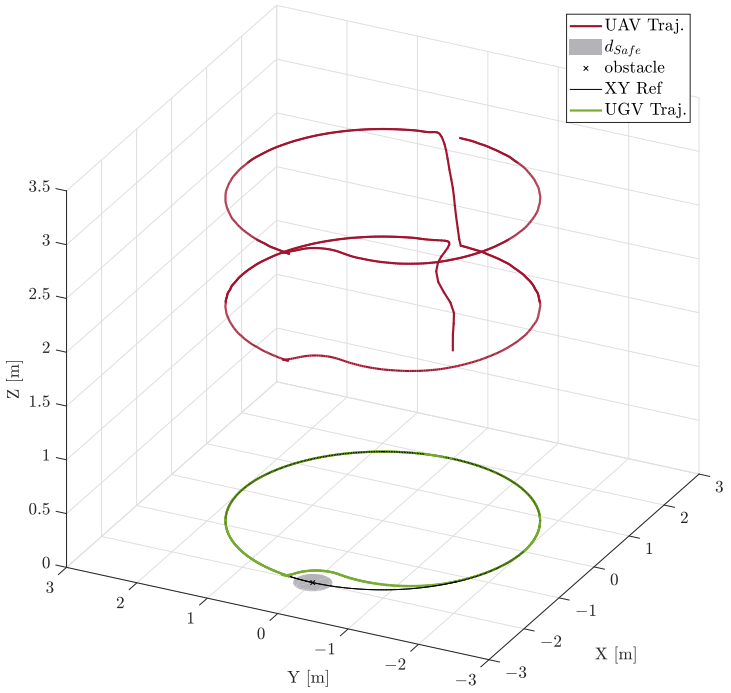

Figure 4. Trajectories of the two robots versus the reference. In grey, the unsafe obstacle's neighborhood.

The UGV moves in a circle as long as it maintains a safe distance from the obstacle. When the car approaches the obstacle, it turns away to avoid it and then realigns with the reference trajectory as illustrated in Figure 5. Meanwhile, the UAV keeps the formation with the leader and increases its height when required.

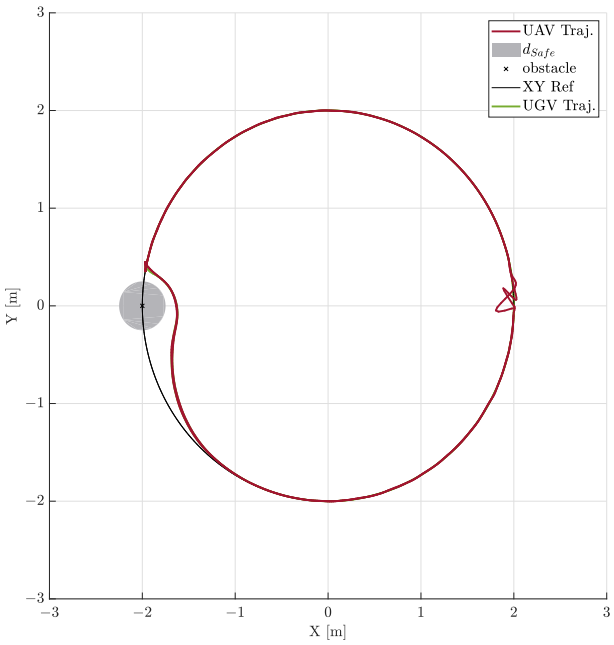

Figure 5. Top view of the trajectory of the UVs. Although the reference passes close to the obstacle, the steering car deviates to keep the safe distance.

Table 1 summarizes all the indexes. The small errors in estimating and predicting the vehicle's state allow to track the reference with deviations of a few centimeters and radiant.

Table 1. The UGV MHE-NMPC scheme performance—Scenario n.1.

$\diamondsuit$	$RMSE_x^{\diamondsuit}$ [m]	$RMSE_y^{\diamondsuit}$ [m]	$RMSE_\theta^{\diamondsuit}$ [rad]	$RMSE_v^{\diamondsuit}$ [m s^{-1}]
Equation (40)	5.30×10^{-2}	1.60×10^{-1}	1.53×10^{-1}	5.77×10^{-2}
Equation (41)	1.55×10^{-2}	1.576×10^{-2}	2.39×10^{-2}	1.06×10^{-2}
Equation(42)	3.24×10^{-5}	4.46×10^{-5}	6.49×10^{-4}	1.73×10^{-4}

As summed up in Table 2, resorting to the one-step prediction ensures the tracking of the desired attitude. Naturally, Equation (45) is smaller in magnitude than Equation (46), as in the NMPC formulation there appears $\boldsymbol{r}_{UAV}(k)$ and not $\boldsymbol{r}_{T,UAV}(k)$. Still, the variations from the actual reference are quite satisfactory.

Table 2. The UAV NMPC performance—Scenario n.1.

$\diamondsuit$	$RMSE_{x_1}^{\diamondsuit}$ [m]	$RMSE_{y_1}^{\diamondsuit}$ [m]	$RMSE_{z_1}^{\diamondsuit}$ [m]	$RMSE_{\psi_1}^{\diamondsuit}$ [rad]
Equation (45)	8.30×10^{-2}	8.74×10^{-2}	8.94×10^{-2}	6.54×10^{-2}
Equation (46)	8.07×10^{-2}	2.00×10^{-1}	8.94×10^{-2}	1.52×10^{-1}

Table 3 reports for each quadcopter's state's components the negligible estimation error.

Table 3. The UAV MHE performance for all components (★)—Scenario n.1.

★	$RMSE_{\star}^{Est,UAV}$	Unit
x_1	1.05×10^{-3}	[m]
x_2	2.41×10^{-2}	$[\mathrm{m\,s^{-1}}]$
y_1	1.01×10^{-3}	[m]
y_2	2.27×10^{-2}	$[\mathrm{m\,s^{-1}}]$
z_1	2.59×10^{-3}	[m]
z_2	1.40×10^{-3}	$[\mathrm{m\,s^{-1}}]$
θ_1	1.63×10^{-2}	[rad]
θ_2	5.25×10^{-2}	$[\mathrm{rad\,s^{-1}}]$
ϕ_1	1.59×10^{-2}	[m]
ϕ_2	5.045×10^{-2}	$[\mathrm{m\,s^{-1}}]$
ψ_1	1.28×10^{-3}	[m]
ψ_2	4.18×10^{-3}	$[\mathrm{m\,s^{-1}}]$

3.2. Circle with Two Obstacles

The second case study maintains the circular trajectory previously described but considers two obstacles, placed in $(x_{Obs1}, y_{Obs1}) = (0\,\mathrm{m}, 2\,\mathrm{m})$ and $(x_{Obs2}, y_{Obs2}) = (0\,\mathrm{m}, 2\,\mathrm{m})$. The aircraft should adhere to a specified take-off reference trajectory as delineated below:

$$z_{1,Ref}(t) = \begin{cases} (1 + (t/30))\,\mathrm{m} & \text{if} \quad t < 30\,\mathrm{s} \\ 2\,\mathrm{m} & \text{if} \quad t \geq 30\,\mathrm{s} \end{cases} \tag{51}$$

The simulation time is also reduced to one minute. Figures 6 and 7 demonstrate that both the leader and the follower qualitatively follow the reference without collisions. Tables 4–6 show the performance indices for both agents. In order to avoid the two obstacles, the ground robot has to deviate its trajectory, which leads to an increase in its $RMSE_x^{\diamondsuit}$ of Equation (40). The drone accurately follows the one-step prediction of the car. However, the increase in the leader's tracking error caused by the obstacles is reflected in the follower's performance.

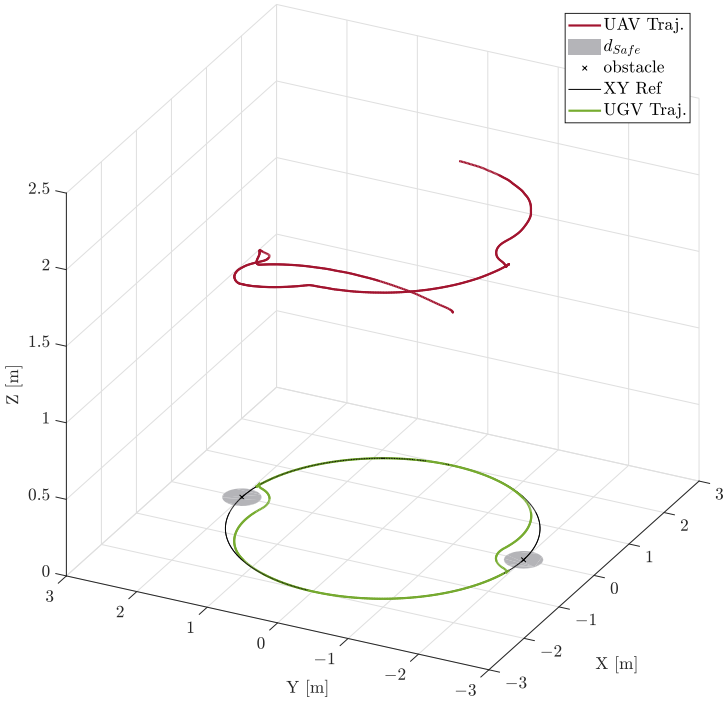

Figure 6. Trajectories of the two robots versus the reference. The drone increases, in the first half of the simulation, its altitude up to 2 m.

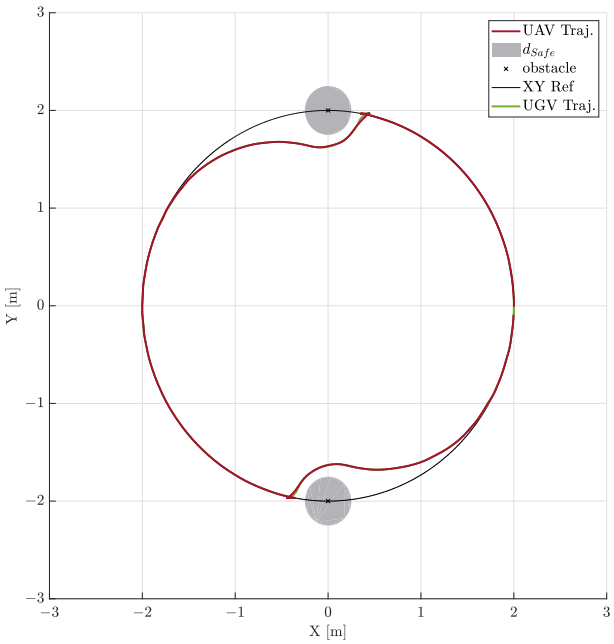

Figure 7. Top view of the trajectory of the UVs in the second scenario with two obstacles along the reference trajectory.

Table 4. The UGV MHE-NMPC scheme performance—Scenario n.2.

◇	$RMSE_x^{\diamondsuit}$ [m]	$RMSE_y^{\diamondsuit}$ [m]	$RMSE_\theta^{\diamondsuit}$ [rad]	$RMSE_v^{\diamondsuit}$ [m s^{-1}]
Equation (40)	2.32×10^{-1}	7.44×10^{-2}	2.17×10^{-1}	8.21×10^{-2}
Equation (41)	1.66×10^{-2}	1.61×10^{-2}	3.20×10^{-2}	1.50×10^{-2}
Equation (42)	5.96×10^{-5}	3.62×10^{-5}	6.55×10^{-4}	2.30×10^{-4}

Table 5. The UAV NMPC performance—Scenario n.2.

◇	$RMSE_{x_1}^{\diamondsuit}$ [m]	$RMSE_{y_1}^{\diamondsuit}$ [m]	$RMSE_{z_1}^{\diamondsuit}$ [m]	$RMSE_{\psi_1}^{\diamondsuit}$ [rad]
Equation (45)	9.05×10^{-2}	8.28×10^{-2}	7.99×10^{-3}	8.58×10^{-2}
Equation (46)	2.80×10^{-1}	9.08×10^{-2}	7.99×10^{-3}	2.14×10^{-1}

Table 6. The UAV MHE performance for all components (★)—Scenario n.2.

★	$RMSE_\star^{Est,UAV}$	Unit
x_1	4.49×10^{-5}	[m]
x_2	1.15×10^{-3}	[m s^{-1}]
y_1	6.87×10^{-5}	[m]
y_2	1.22×10^{-3}	[m s^{-1}]
z_1	1.48×10^{-5}	[m]
z_2	4.93×10^{-5}	[m s^{-1}]
θ_1	1.03×10^{-3}	[rad]
θ_2	3.88×10^{-3}	[rad s^{-1}]
ϕ_1	1.03×10^{-3}	[m]
ϕ_2	3.88×10^{-2}	[m s^{-1}]
ψ_1	1.29×10^{-3}	[m]
ψ_2	4.19×10^{-3}	[m s^{-1}]

3.3. Lemniscate of Bernoulli with One Obstacle

For the third simulation, Bernoulli's lemniscate is chosen as the desired trajectory, described by the following equations:

$$\begin{cases} x_{Ref}(t) & = 2\,c(t)\,\cos\left(t\,\frac{2\pi}{60}\right) \\ y_{Ref}(t) & = c(t)\,\sin\left(2t\,\frac{2\pi}{60}\right) \\ \theta_{Ref}(t) & = \arctan2(v_{y,Ref}(t), v_{x,Ref}(t)) \\ v_{Ref}(t) & = \sqrt{v_{x,Ref}(t)^2 + v_{y,Ref}(t)^2} \end{cases} \tag{52}$$

with

$$c(t) = 2\,/\left(3 - \cos\left(2t\,\frac{2\pi}{60}\right)\right) \tag{53}$$

$$v_{x,Ref}(t) = \left.\frac{dx_{Ref}(\tau)}{d\tau}\right|_{\tau=t} \quad v_{y,Ref}(t) = \left.\frac{dy_{Ref}(\tau)}{d\tau}\right|_{\tau=t} \tag{54}$$

and the only obstacle is in the origin. The reference altitude of the UAV is kept constant at $z_{1,Ref}(t) = 2\,\mathrm{m}$.

The trajectories of the two agents are depicted in Figures 8 and 9. The UGV initially follows the lemniscate path until it approaches too close to the origin. At that point, it navigates around the obstacle, resulting in a decrease in tracking accuracy as shown in Table 7. Due to the control constraints, the agent gradually returns to the reference path without sudden movements. When the steering car returns to its starting point, it must pass the origin again, thereby repeating the same behavior. Regarding the quadcopter, except for the initial moments of the simulation due to the initial conditions, it maintains the desired height while following the leader from above. For a quantitative analysis, see Tables 8 and 9.

Table 7. The UGV MHE-NMPC scheme performance—Scenario n.3.

$\diamondsuit$	$RMSE_x^{\diamondsuit}$ [m]	$RMSE_y^{\diamondsuit}$ [m]	$RMSE_\theta^{\diamondsuit}$ [rad]	$RMSE_v^{\diamondsuit}$ [m s^{-1}]
Equation (40)	3.26×10^{-1}	1.93×10^{-1}	2.84×10^{-1}	9.55×10^{-2}
Equation (41)	1.79×10^{-2}	1.62×10^{-2}	3.39×10^{-2}	1.53×10^{-2}
Equation (42)	1.17×10^{-4}	6.36×10^{-5}	7.02×10^{-4}	5.13×10^{-4}

Table 8. The UAV NMPC performance—Scenario n.3.

$\diamondsuit$	$RMSE_{x_1}^{\diamondsuit}$ [m]	$RMSE_{y_1}^{\diamondsuit}$ [m]	$RMSE_{z_1}^{\diamondsuit}$ [m]	$RMSE_{\psi_1}^{\diamondsuit}$ [rad]
Equation (45)	9.16×10^{-2}	7.01×10^{-2}	8.46×10^{-2}	9.53×10^{-2}
Equation (46)	3.63×10^{-1}	2.16×10^{-1}	8.46×10^{-2}	2.90×10^{-1}

Table 9. The UAV MHE performances for all components ($\bigstar$)—Scenario n.3.

$\bigstar$	$RMSE_\bigstar^{Est,UAV}$	Unit
x_1	8.14×10^{-4}	[m]
x_2	1.84×10^{-2}	[m s^{-1}]
y_1	1.25×10^{-3}	[m]
y_2	2.82×10^{-2}	[m s^{-1}]
z_1	2.57×10^{-3}	[m]
z_2	1.13×10^{-3}	[m s^{-1}]
θ_1	1.23×10^{-2}	[rad]
θ_2	3.92×10^{-2}	[rad s^{-1}]
ϕ_1	1.97×10^{-2}	[m]
ϕ_2	6.21×10^{-2}	[m s^{-1}]
ψ_1	1.29×10^{-3}	[m]
ψ_2	4.21×10^{-3}	[m s^{-1}]

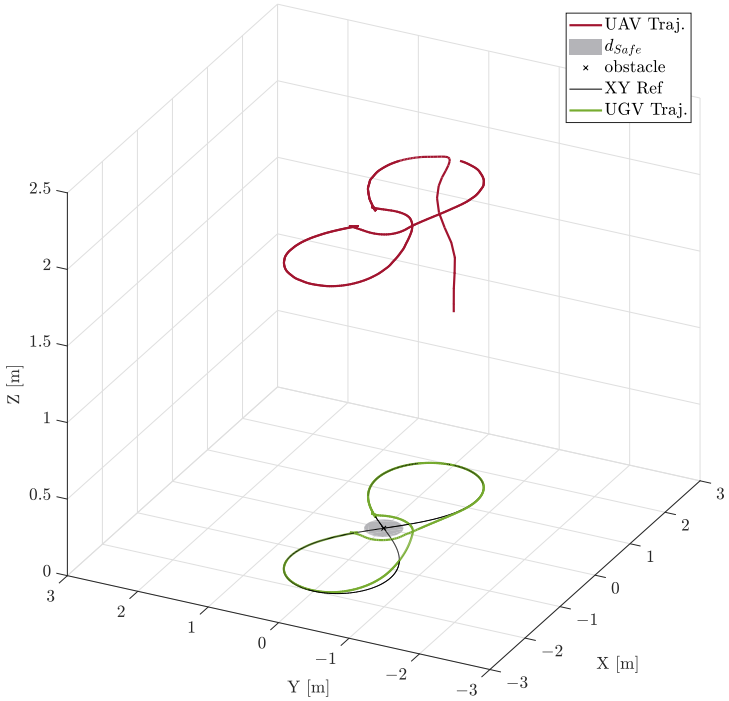

Figure 8. Trajectories of the two robots versus the reference. The drone's reference altitude is 2 m.

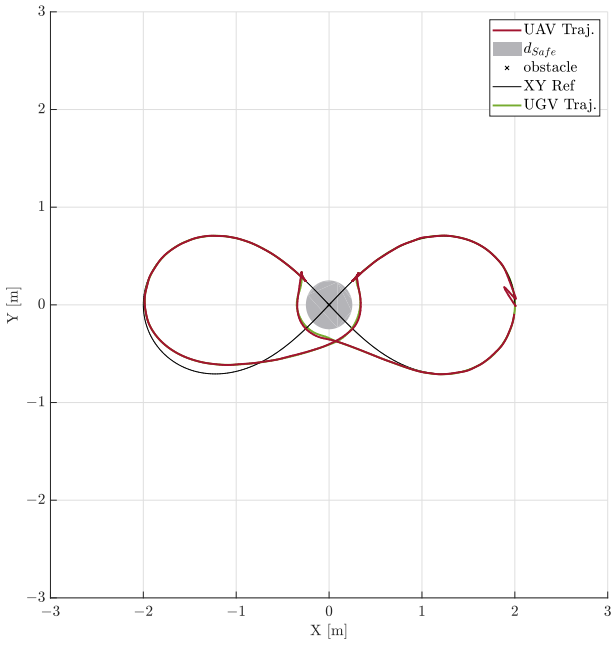

Figure 9. Top view of the trajectory of the UVs in the third scenario with the reference trajectory of the lemniscate of Bernoulli.

3.4. Lemniscate of Bernoulli with Two Obstacles

The last case study is similar to the previous one, but a second obstacle is added. Both are in the reference positions that the robot should ideally pass at $t = 24\,\text{s}$ and $t = 54\,\text{s}$. To complicate the test, the constant height is replaced by a decreasing trajectory:

$$z_{1,Ref}(t) = \begin{cases} 2\,\text{m} & \text{if} \quad t < 30\,\text{s} \\ 2 - (1.8\,(t-30)/30)\,\text{m} & \text{if} \quad t \geq 30\,\text{s} \end{cases} \tag{55}$$

The resulting performance indices are presented in Tables 10–12. As can be seen from Figures 10 and 11, considerations similar to those for the single barrier case can be drawn. Note that the $RSME_x$ and $RMSE_y$ of the tracking error Equation (40) are smaller compared to the values in the first row of Table 10, which remain the worst values among all the simulations. Nevertheless, in none of the case studies did the errors diverge and the ground vehicle entered the unsafe region.

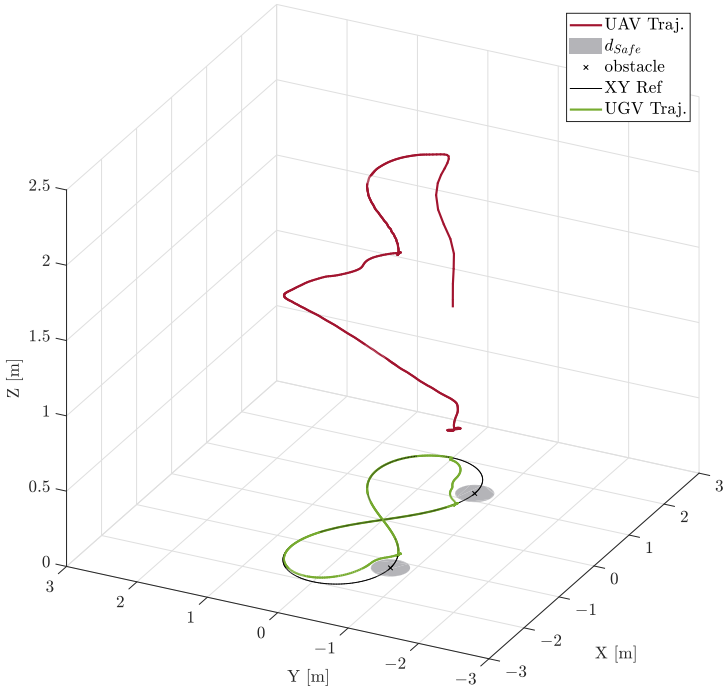

Figure 10. Trajectories of the ground and the aerial robots. The UGV has to follow the lemniscate of Bernoulli trajectory while avoiding two obstacles on its way. The quadcopter in the second half of the simulation reduces its height.

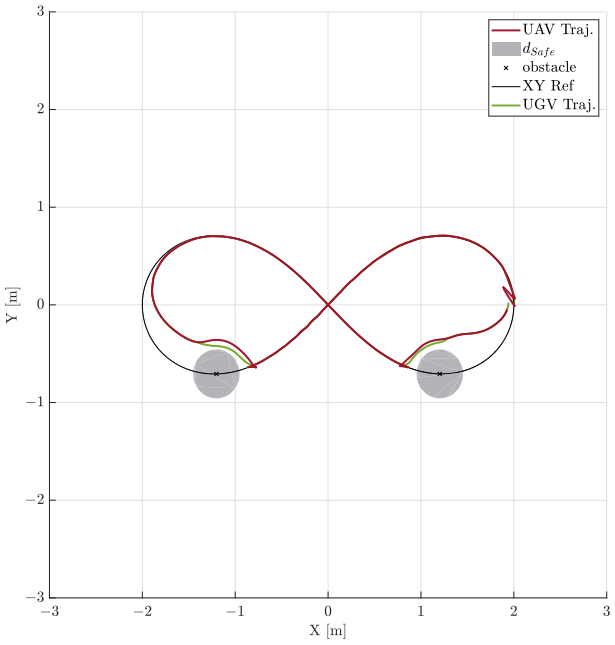

Figure 11. Top view of the two agents' paths compared with the reference in the fourth scenario.

Table 10. The UGV MHE-NMPC scheme performance—Scenario n.4.

$\diamondsuit$	$RMSE_x^{\diamondsuit}$ [m]	$RMSE_y^{\diamondsuit}$ [m]	$RMSE_\theta^{\diamondsuit}$ [rad]	$RMSE_v^{\diamondsuit}$ [m s^{-1}]
Equation (40)	2.17×10^{-1}	4.61×10^{-2}	2.03×10^{-1}	7.65×10^{-2}
Equation (41)	1.50×10^{-2}	1.92×10^{-2}	3.46×10^{-2}	1.51×10^{-2}
Equation (42)	2.91×10^{-5}	2.85×10^{-5}	6.94×10^{-4}	9.77×10^{-5}

Table 11. The UAV NMPC performance—Scenario n.4.

$\diamondsuit$	$RMSE_{x_1}^{\diamondsuit}$ [m]	$RMSE_{y_1}^{\diamondsuit}$ [m]	$RMSE_{z_1}^{\diamondsuit}$ [m]	$RMSE_{\psi_1}^{\diamondsuit}$ [rad]
Equation (45)	8.22×10^{-2}	6.10×10^{-2}	1.12×10^{-1}	9.20×10^{-2}
Equation (46)	2.54×10^{-1}	6.30×10^{-2}	1.13×10^{-1}	2.18×10^{-1}

Table 12. The UAV MHE performance for all components ($\bigstar$)— Scenario n.4

$\bigstar$	$RMSE_{\bigstar}^{Est,UAV}$	Unit
x_1	8.14×10^{-4}	[m]
x_2	1.84×10^{-2}	[m s^{-1}]
y_1	1.25×10^{-3}	[m]
y_2	2.82×10^{-2}	[m s^{-1}]
z_1	2.57×10^{-3}	[m]
z_2	1.13×10^{-3}	[m s^{-1}]
θ_1	1.22×10^{-2}	[rad]
θ_2	3.92×10^{-2}	[rad s^{-1}]
ϕ_1	1.97×10^{-2}	[m]
ϕ_2	6.21×10^{-2}	[m s^{-1}]
ψ_1	1.29×10^{-3}	[m]
ψ_2	4.21×10^{-3}	[m s^{-1}]

4. Discussion

In this paper, nonlinear rolling horizon techniques are applied to a heterogeneous fleet. In particular, a distributed solution is proposed to address the formation path-tracking problem while avoiding obstacles. Results across various scenarios and maneuvers confirm the validity of the proposed control architecture, ensuring small errors with respect to reference trajectories. The possibilities for future research are manifold. Firstly, the results suggest that the proposed scheme ensures small tracking errors, but still the stability and convergence of the NMPC should be proved in a more rigorous way. Secondly, the works [5–8] indicate the superiority of NMPC over classical controllers. Conducting a similar comparative analysis, particularly in scenarios involving data loss, would be valuable to quantify the improvements in performance brought by the usage of a NMPC. Lastly, to go beyond numerical simulation, the implementation and embedding of the control architecture in real robots, such as the Parrot Bebop 2.0 and JetRacer equipped with LIDAR mapping, would strengthen the validation and applicability of the proposed methods in practical scenarios.

Author Contributions: Conceptualization, A.E.S.M., A.B., S.G., R.S. and E.Z.; methodology, A.E.S.M., A.B., S.G., R.S. and E.Z.; software, A.E.S.M., A.B., S.G., R.S. and E.Z.; validation, A.E.S.M., A.B., S.G., R.S. and E.Z.; formal analysis, A.E.S.M., A.B., S.G., R.S. and E.Z.; investigation, A.E.S.M., A.B., S.G., R.S. and E.Z.; resources, A.E.S.M., A.B., S.G., R.S. and E.Z.; data curation, A.E.S.M., A.B., S.G., R.S. and E.Z.; writing—original draft preparation, A.E.S.M., A.B., S.G., R.S. and E.Z.; writing—review and editing, A.E.S.M., A.B., S.G., R.S. and E.Z.; visualization, A.E.S.M., A.B., S.G., R.S. and E.Z.; supervision, A.E.S.M., A.B., S.G., R.S. and E.Z.; project administration, A.E.S.M., A.B., S.G., R.S. and E.Z.; funding acquisition, A.E.S.M., A.B., S.G., R.S. and E.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available at https://github.com/AlessandraElisaSindiMorando/Nonlinear_Rolling_Horizon_Control_Automation2024 (accessed on 20 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Verma, J.K.; Ranga, V. Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope. *J. Intell. Robot. Syst.* **2021**, *102*, 10. [CrossRef] [PubMed]
2. Bacheti, V.P.; Brandão, A.S.; Sarcinelli-Filho, M. A Path-Following Controller for a UAV-UGV Formation Performing the Final Step of Last-Mile-Delivery. *IEEE Access* **2021**, *9*, 142218–142231. [CrossRef]
3. Zhang, J.; Yue, X.; Zhang, H.; Xiao, T. Optimal Unmanned Ground Vehicle—Unmanned Aerial Vehicle Formation-Maintenance Control for Air-Ground Cooperation. *Appl. Sci.* **2022**, *12*, 3598. [CrossRef]
4. Tassanbi, A.; Iskakov, A.; Do, T.D.; Ali, M.H. Interactive Real-time Leader Follower Control System for UAV and UGV. In Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Male, Malediven, 16–18 November 2022; pp. 1–8. [CrossRef]
5. Okasha, M.; Kralev, J.K.; Islam, M. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. *Aerospace* **2022**, *9*, 298. [CrossRef]
6. Al-Saoudi, A.F.; Al-Aubidy, K.M.; Al-Mahasneh, A.J. Comparison of PID, Fuzzy Logic, ANFIS and Model Predictive Controllers for Cruise Control System. In Proceedings of the 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD), Erbil, Iraq, 22–25 April 2024; pp. 263–265.
7. Kaçmaz, B.; Söylemez, M.T. Implementation and Comparison of PID, PI-PD, LQR and MPC on Separation Clutch System in Slip. In Proceedings of the 2021 22nd IEEE International Conference on Industrial Technology (ICIT), Valencia, Spain, 10–12 March 2021; Volume 1, pp. 61–67.
8. Liu, J.; Yang, Z.; Huang, Z.; Li, W.; Dang, S.; Li, H. Simulation Performance Evaluation of Pure Pursuit, Stanley, LQR, MPC Controller for Autonomous Vehicles. In Proceedings of the 2021 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 1444–1449.
9. Cenerini, J.; Mehrez, M.W.; Han, J.w.; Jeon, S.; Melek, W. Model Predictive Path Following Control without terminal constraints for holonomic mobile robots. *Control Eng. Pract.* **2023**, *132*, 105406. [CrossRef]
10. Bozzi, A.; Graffione, S.; Kockelkoren, M.M.; Sacile, R.; Zero, E. Path Tracking for Wheeled Mobile Robot Using Non Linear Model Predictive Control in Indoor Environment. In Proceedings of the 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), Rome, Italy, 3–6 July 2023; pp. 1379–1384. [CrossRef]
11. Recalde, L.F.; Guevara, B.S.; Andaluz, V.; Varela, J.; Gimenez, J.; Gandolfo, D. Nonlinear MPC for Multiple Quadrotors in Dynamic Environments. In Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, 21–24 June 2022; pp. 1201–1209. [CrossRef]
12. Biegler, L. A perspective on nonlinear model predictive control. *Korean J. Chem. Eng.* **2021**, *38*, 1317–1332. [CrossRef]
13. Deniz, N.; Jorquera, F.; Torres-Torriti, M.; Cheein, F.A. Model predictive path-following controller for Generalised N-Trailer vehicles with noisy sensors and disturbances. *Control Eng. Pract.* **2024**, *142*, 105747. [CrossRef]
14. Aaltonen, O. Nonlinear Model Predictive Control and Estimation applied to Selective Catalytic Reduction. Master's Thesis, Åbo Akademi University, Turku, Finland, 2022.
15. Lindqvist, B.; Mansouri, S.S.; Agha-mohammadi, A.A.; Nikolakopoulos, G. Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6001–6008. [CrossRef]
16. Eltrabyly, A.; Ichalal, D.; Mammar, S. Quadcopter Trajectory Tracking in the Presence of 4 Faulty Actuators: A Nonlinear MHE and MPC Approach. *IEEE Control Syst. Lett.* **2022**, *6*, 2024–2029. [CrossRef]
17. Castillo Garcia, P.; Munoz, L.; Gil, P. Modeling Approaches—The results in this chapter were developed in collaboration with H. Abaunza and J. Carino from the UMI LAFMIA 3175, Mexico . In *Indoor Navigation Strategies for Aerial Autonomous Systems*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 31–50. [CrossRef]
18. Lin, X.; Chen, X. Realization of Ackermann robot obstacle avoidance navigation based on Multi-sensor fusion SLAM. In Proceedings of the 2023 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT), Guangzhou, China, 18–20 August 2023; pp. 354–359. [CrossRef]
19. Köhler, J.; Müller, M.A.; Allgöwer, F. A nonlinear tracking model predictive control scheme for dynamic target signals. *Automatica* **2020**, *118*, 109030. [CrossRef]

20. Sani, M.; Robu, B.; Hably, A. Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control. In Proceedings of the IECON 2021—47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–6. [CrossRef]
21. Andersson, J.A.E.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi—A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [CrossRef]