This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3554368



# Smooth Human-Robot Shared Control for Autonomous Orchard Monitoring with UGVs

Cheikh Melainine El Bou[1], Michele Focchi[2,3], Michael Chang[1], Marco Camurri[4,1], and Karl D. von Ellenrieder[1]

***Abstract*—Precision agriculture offers the opportunity to automate routine or difficult tasks in orchards and vineyards, such as spraying or inspection, with Unmanned Ground Vehicles (UGV). In this context, human operators should be kept in the closed-loop control of the robot for safety and reliability. This work is motivated by the challenges of effectively deploying human-robot shared control in the field. First, an asymptotically stable controller keeps the robot on the desired trajectory between rows of trees, whose distance is on the order of the robot's width. Second, the robot must efficiently avoid static and moving obstacles on its path. Third, the control inputs must not exceed the actuator limits, which can degrade trajectory tracking performance, cause instability, or damage critical hardware. Finally, in real-life scenarios, user intervention is sometimes required to manage unpredictable situations. To overcome these challenges, we propose and deploy a shared controller that continuously and smoothly varies the ratio of human and automatic control inputs depending on the human's intent, geometrically rescales trajectory inputs to maintain bounded control, and incorporates obstacle avoidance capabilities – all while preserving asymptotic stability of the closed-loop system. Additionally, we introduce a time re-scaling strategy that modifies trajectory evolution, ensuring target positions remain within a defined vicinity of the robot. The system performance was assessed in simulation and in 26 field trials inside an apple orchard using different obstacle configurations, weather, and terrain conditions, with a success rate of 100% and an average tracking error of 0.1 m.**

***Note to Practitioners*—The proposed shared control approach is developed for use with a differentially steered Uncrewed Ground Vehicle (UGV) with first order kinematic constraints and can be adapted to different indoor and outdoor scenarios. The environment in which the UGV is deployed should be mapped in advance to create a reference trajectory for the UGV to follow. If the location of obstacles is not known in advance, an obstacle detection and tracking system, as well as an online mapping system, must be developed. A simulated model of the UGV and the environment are useful for determining initial values of the shared control gains that can be further tuned when the physical platform is first deployed. In GPS-denied scenarios, a Similtaneous Localization and Mapping (SLAM) system must be implemented; a lidar-inertial based SLAM system is recommended. A force-reflexive joystick is ideal for sensitive human input. In addition, the communication between the UGV and the base station (where the human operator supervises the UGV and provides commands to it) should have minimal time delays, not exceeding typical human reaction time (ca. 0.25 s).**

[1]The authors are with Facoltà di Ingegneria, Libera Università di Bolzano, 39100 BZ Italy, cheikhmelainine@ieee.org, mchang@unibz.it, karl.vonellenrieder@ieee.org.

[2]The author is with the Dipartimento di Ingegneria e Scienza dell'Informazione (DISI), Università di Trento, michele.focchi@ieee.org

[3]The author is with the Dynamic Legged Systems, Istituto Italiano di Tecnologia (IIT), Genova

[4]The author is with the Dipartimento di Ingegneria Industriale (DII), Università di Trento, marco.camurri@unitn.it

This work was sponsored in part by the Autonomous Province of Bolzano, Italy (Recoaro Project #4122), and in part by the European Commission Horizon Europe (Sestosenso project, Grant 101070310)

Fig. 1. The experimental area, consisting of an apple orchard with 2.7 m wide and 25 m long rows. The robot has been tested under various weather and terrain conditions, such as dry sunny days and tall grass, as well as muddy terrain conditions in rainy and cloudy weather (as shown in picture).

***Index Terms*—Human-robot shared control, collision avoidance, trajectory tracking, field robotics, agricultural automation.**

## I. Introduction

Human-robot shared control is becoming increasingly popular in many fields, such as the automotive industry [1], [2], [3], healthcare [4], [5], agriculture [6], [7], surveying [8], and environmental monitoring [9], [10]. Shared control lies in between teleoperation, where the control input is generated solely by a human, and supervisory control, where the control input is fully automatic, but a human can change the control objective. These types of semiautomatic control permit humans and robots to mutually benefit: humans can operate with a decreased demand on attention, while robotic systems are safer, thanks to human intervention. Additional benefits are cost reduction (fewer sensors needed, as the system is augmented by human perception) and improved robustness (anomalies can be efficiently detected and handled by the human).

In this work, we focus on the design of a shared-control strategy for a differentially steered UGV (Fig. 1), designed with an eye towards the development of precision agriculture. UGVs are often deployed at agricultural sites [11] to autonomously perform tasks, such as: spraying [12], [13], inspection [14], and harvesting [15], [16], [17]. Several mobile robot platforms have been designed for the above tasks, significantly improving production [18].

Many of these platforms are still in the pilot research stage because of the challenges faced in agricultural environments. In particular, orchards are densely populated with irregularly distributed obstacles and narrow pathways, in which tolerance to errors in trajectory planning and trajectory following are much lower than they would be in open agricultural fields.

To overcome these challenges, vision-based algorithms were developed for autonomous mobile robot navigation [19] and path planning, such as bi-directional RRT [20]. However, disturbances such as bushes, leaves, and lighting conditions significantly affect the performance of vision-based algorithms, especially for collision avoidance. In [21], the authors proposed a modular and scalable sensing system with a distributed architecture for assisting GPS-based autonomous navigation, where three Time-of-Flight (ToF) laser range finders and seven multichannel infrared (IR) sensor arrays mounted on the front, left, and right sides of the robot provided it with collision avoidance capabilities.

Despite these advances towards full autonomy, for the aforementioned tasks the introduction of a human operator into the closed-loop system is still desirable to improve safety and reliability [22], [23]. The navigation algorithm for the mobile robot can be implemented with the minimum sensing system, where human awareness and decision-making can be replaced by relying on the sensing system in emergency cases.

In this paper, we consider a Human-robot shared control system consisting of one human operator and a UGV, in which we ensure that the UGV can autonomously maneuver in an agriculture environment with static and moving obstacles. We designed the automatic controller to allow any differential drive robot to maneuver in an environment with static and moving obstacles, in which the localization of the UGV can be provided using a vision-based or GPS-based system. Thus, our proposed shared control is more cost-effective since it can be deployed on a different mobile platform and with different navigation systems. However, several challenges must be addressed to effectively deploy human-shared control in the field, including keeping the robot on the desired trajectory between rows of trees, avoiding static and moving obstacles, ensuring the actuator limits are never exceeded, and smoothly fusing the human control input and the automatic control input, in proportions that correctly reflect the human's intent.

### A. Contributions

The main contributions of our work are:

1) We present a human-robot shared control trajectory tracking architecture that supports: (a) a continuously varying ratio of human input to automatic control input depending on the human's intent; (b) a continuous geometrical re-scaling of the trajectory tracking control input to ensure it is bounded; and (c) an automatic obstacle avoidance controller that ensures safe maneuvering in a complex environment. Our approach is unique in achieving these criteria *with* with the mathematical guarantee of the ultimate boundedness of the closed loop human-robot shared control system.
2) We also propose a *time re-scaling* strategy to modify the evolution of the reference trajectory, ensuring that the actual target position remains within a user-defined vicinity of the robot at all times. This approach situates the method as an intermediary between traditional trajectory tracking, with its precise temporal evolution, and path following, which primarily focuses on spatial positioning, thus balancing temporal and spatial adaptability.
3) We extensively test the above on a differential drive mobile robot operating in a real apple orchard at different times of the year, including in unfavorable conditions, such as muddy terrain and tall grass. We also compare our proposed control architecture against a state-of-the-art nonlinear Model Predictive Control (MPC) in simulation, demonstrating that our system has comparable performance, at a fraction of the computational power.

The remainder of this paper is structured as follows. Section II covers the relevant literature in human-robot shared control. The control problem addressed by the paper is stated in Section III. In Section IV, we present the design of the shared control approach that allows the differential drive robot to maneuver in an unstructured and partially unknown environment. The experimental setup is illustrated in Section V. Section VI shows experimental validation of the approach both in simulation and real experiments, together with a comparison with a state-of-the-art approach based on MPC. Finally, Section VII presents our main conclusions and Future work.

## II. Related Works

The focus of human-robot shared control research can be divided into two main categories: a) how to model human behavior in order to anticipate human intention; and b) how to define switching policies to mix the human command input with the automatic controller input. For completeness, we briefly cover the first category, but concentrate on the second one in this paper.

Additionally, the interface used by the human operator to generated the human control input affects the design of the shared control system. For example, we can have direct contact between the human and the robot, as in automotive applications [24] where a human uses a steering wheel to provide commands to the vehicle, or we can have remote human input, as in telerobotic systems where a human uses a master robot to command a slave robot located far away [25].

### A. Modelling human behavior

Many applications of shared control focus on the modeling of human behaviors, especially for teleoperation [26], [27], [28], where the human intended trajectory is predicted and fed back to the robot as a reference trajectory. A second application is assistive robotics. For example in [29], [30] a shared control approach allows a person with a disability to control an actuated wheelchair. In these works, the automatic controller predicts the path the person intends to take and follows it. An example in the automotive field is [31], where an MPC is used as the automatic controller to guide a four-wheel vehicle, while a gated recurrent neural network is used to model the driver's behavior to better predict their steering intention.

### B. Switching Policy

Switching between the human commands and the automatic controller while ensuring asymptotic stability is challenging from a control point of view. A few studies investigate this

problem, such as [32], where an Advanced Driver Assistance System (ADAS) shares the control of the steering wheel with the driver and a cybernetic driver model (based on the driver visualization information) is combined with the vehicle model for lane keeping. A linear driver-vehicle model is used in the H2-Preview Control Approach, where the control inputs are designed to ensure the stability of the closed-loop system considering the uncertainties in the human model. Because the model parameters are obtained from human trials, its generalization depends on how the data are collected.

In contrast, [33] proposes a driver-vehicle shared control for lane-keeping and obstacle avoidance, in which the steady-state yaw rate and the rear slip angle at peak tire force are used to create a stable control that does not violate the vehicle handling limits. A graph search algorithm forms collision-free tubes through the environment in which the vehicle trajectory must be contained. A model predictive controller uses the cost between the driver commands and the smooth trajectory, the constraints on yaw rate and rear slip angle, and collision-free tube constraints. This work was developed to assist a driver by adjusting the driver steering command to respect the given constraint. Thus, the driver's level of authority is always limited, which means the interaction is not a mixed-initiative interaction, in which the driver can have control when necessary and suspend his/her input, when desired [34].

The work in [35] introduces a novel exponential mixed-initiative interaction policy characterized by a smooth transition between human input and an automatic control input. The shared controller was designed for a second-order system with single input and single output, where the asymptotic stability of the human-robot closed-loop system was proven, but not demonstrated on a physical system.

In a follow up work, the same authors [36] develop a shared control system for a human user and a four-wheeled, front-steered vehicle with second-degree non-holonomic acceleration constraints. A second order sliding mode automatic control input is mixed with the human control input using the exponential policy. An exponential policy is used also in [37] to mix the human commands with a sliding mode controller for an underwater vehicle. While addressing the mixed-initiative shared control and stability, neither work addresses the safe avoidance of obstacles in the environment.

In [38] we develop a shared trajectory tracking with collision avoidance controller for a differentially steered robot, in which the trajectory tracking controller is a Lyapunov-based controller combined with a collision-avoidance controller to form the automatic control input. This approach is validated in an indoor scenario with a differential drive service robot. However, this work does not consider actuator magnitude limits, which is critical in field operations where terrain properties change over time.

Here, we introduce the geometric scaling [39] of the Lyapunov-based trajectory tracking controller developed in [38], to continuously rescale the control inputs so that the actuator magnitude constraints of the robot are respected at all times. We also introduce a new control barrier function design that includes turning in a preferred direction to avoid deadlocks that can occur with symmetrical control barrier functions when an obstacle is located directly in front of the robot.

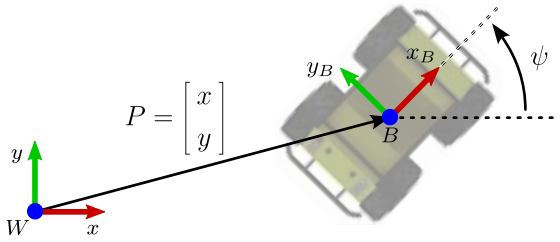



Fig. 2. Reference frames and conventions definitions. The World frame $W$ is fixed to Earth, with the $z$-axis pointing upwards. The origin of $W$ is arbitrary and defined as the starting point of the experiments. The position of the robot's base frame $B$ is identified by $P$, whose two coordinates $x, y$ are defined in $W$, while the robot's yaw is denoted by $\psi$.

### C. Human-robot interface

Our proposed shared control is developed for use with a remotely controlled UGV. Joystick devices can be used to remotely command a robot, as in telerobotics, where the joystick is the master and the robot is the slave [40]. In [41] the robot is a mobile manipulator, and the joystick is a haptic device with the same number of degrees of freedom as the manipulator. Here, we consider a mobile robot commanded with two control inputs provided using the two degrees of freedom of a force-reflexive joystick. The joystick tracks the automatic control input signal and resists the human user when he/she tries to drive the UGV away from its preplanned trajectory, thus providing intuitive haptic feedback to the human user about how far away, and in which direction, the robot is from its desired trajectory.

Our proposed approach is validated with outdoor field experiments in an apple orchard. Furthermore, to provide a benchmark for its effectiveness, we compare our approach in simulation against a shared human-robot controller where the trajectory tracking and obstacle avoidance are implemented using a nonlinear MPC strategy.

## III. Problem Statement

Consider a differential drive robot, such as shown in Fig. 2. Suppose the robot's wheels roll without slipping (i.e. the sway speed is identically zero). In a three-dimensional configuration space, the kinematic equations of motion are given by

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

with the mapping $\boldsymbol{g} : \mathcal{S} \to \mathbb{R}^{3\times 2}$,

$$\boldsymbol{g} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where

$$\boldsymbol{x} := \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \in \mathbb{R}^2 \times \mathcal{S}, \quad \text{and} \quad \boldsymbol{u} := \begin{bmatrix} u \\ r \end{bmatrix} \in \mathbb{R}^2. \tag{3}$$

The variables appearing in (3) include $x$-$y$ Cartesian position of the robot, the heading angle $\psi$ (about the axis $z$ pointing

up), the surge speed (i.e. longitudinal speed in body-fixed coordinates) $u$ and the yaw rate $r$. The set of Euler angles defined on the interval $[-\pi \quad \pi]$ is indicated with $\mathcal{S}$. Since we assume no lateral slippage, the motion of the robot is nonholonomic first-order kinematic constraints, as given by (1).

The speeds of the two motors can be independently controlled to provide a desired surge speed $u$ and yaw rate $r$, which we take to be the two control inputs. Let

$$\boldsymbol{x}_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ \psi_d(t) \end{bmatrix} \in \mathbb{R}^2 \times \mathcal{S} \tag{4}$$

be the trajectory to be tracked. The trajectory $\boldsymbol{x}_d(t)$ and its first and second derivatives, $\dot{\boldsymbol{x}}_d(t)$ and $\ddot{\boldsymbol{x}}_d(t)$, are assumed to be smooth and bounded. The trajectory could be designed using a simple point-to-point planning method (e.g. [42]), or more advanced motion planning techniques that take energetic requirements or risk into account [43], [44], [45], [46], [47], [48]. The output of the system is the pose of the robot in the inertial frame, i.e. $\boldsymbol{y}(t) = \boldsymbol{x}(t)$.

We consider $n$ obstacles represented by the position vector $\boldsymbol{x}_o(t) = [\boldsymbol{x}_{o1}(t), ..., \boldsymbol{x}_{on}(t)]^\mathsf{T} \in \mathbb{R}^{2n}$ in the inertial reference frame, where

$$\boldsymbol{x}_{oi}(t) = \begin{bmatrix} x_{oi} \\ y_{oi} \end{bmatrix} \in \mathbb{R}^2, \quad \text{for} \quad i \in \{1, ..., n\}. \tag{5}$$

The problem we address in this work can be summarized as designing a shared controller with the following properties:

1) The differential drive UGV can operate autonomously in unstructured and partially unknown environments while following the desired trajectory (4) with static and moving obstacles represented by (5).
2) At any time, the human operator should be able to manually take and release control of the UGV whenever needed, without loss of vehicle autonomous control.
3) The trajectory tracking control input is globally uniformly ultimately bounded.
4) The subsequent closed-loop human-robot shared controller is globally uniformly ultimately bounded.

## IV. Control Design

In this section we introduce our proposed shared trajectory tracking with a collision avoidance controller for a human-robot system composed of one human operator and a differential drive UGV. The trajectory tracking controller design is based on the Lyapunov stabilization approach. The design of the associated collision avoidance controller is based on the use of a control barrier function, and the rule for blending the human control input with the automatic control input is designed using a Lyapunov-function-like storage function (to obtain a measure of human intent) as the argument of convex combination of exponential functions.

Since it is impossible to guarantee that a human will always have sufficient situational awareness and response time to avoid obstacles, an automatic collision avoidance method is used to help ensure the safe operation of the UGV. A typical situation where this feature is needed is when the original (i.e., global) plan (e.g., made using a satellite map) needs to be updated/modified based on *local* visual information (e.g. when the presence of a previously unperceived obstacle is detected).

We have included a block diagram of the shared trajectory tracking with a collision avoidance controller Fig. 3.

### A. Kinematic Trajectory Tracking Controller

Consider the trajectory tracking controller inputs

$$\boldsymbol{u}_c = \begin{bmatrix} u_c \\ r_c \end{bmatrix} = \begin{bmatrix} u_d \cos(\tilde{\psi}_b) + k_x \tilde{x}_b \\ r_d + k_\psi \sin \tilde{\psi}_b + \tilde{y}_b u_d \end{bmatrix}, \tag{6}$$

where $u_d$ is the desired surge speed, $r_d$ is the desired yaw rate, and $k_x > 0$, $k_\psi > 0$ are control design parameters. The error between the desired and actual state expressed in the body-fixed frame is $\tilde{\boldsymbol{x}}_b = \left[\tilde{x}_b, \tilde{y}_b, \tilde{\psi}_b\right]^\mathsf{T}$, where

$$\begin{aligned} \tilde{x}_b &= \tilde{x} \cos\psi + \tilde{y} \sin\psi, \\ \tilde{y}_b &= -\tilde{x} \sin\psi + \tilde{y} \cos\psi, \\ \tilde{\psi}_b &= \tilde{\psi}, \end{aligned} \tag{7}$$

$\tilde{x} = x_d - x$, $\tilde{y} = y_d - y$, and $\tilde{\psi} = \psi_d - \psi$.

The kinematic error model in the body-frame for the system (1) is

$$\begin{aligned} \dot{\tilde{x}}_b &= u_d \cos\tilde{\psi}_b - u_c + r_c \tilde{y}_b, \\ \dot{\tilde{y}}_b &= u_d \sin\tilde{\psi}_b - r_c \tilde{x}_b, \\ \dot{\tilde{\psi}}_b &= r_d - r_c. \end{aligned} \tag{8}$$

In the absence of exogenous disturbances and unmodeled dynamics, the trajectory tracking controller in (6) is asymptotically stable according to [Section 5.4][49]. However, these effects will be present in real-world implementations. Here, we build upon the work of [49] by examining the stability of the closed loop tracking system when exogenous disturbances and unmodeled dynamics affect the kinematic model (1) through the same channels as the control inputs (matched disturbances). The resulting kinematic model with disturbances is

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})(\boldsymbol{u} + \boldsymbol{d}), \tag{9}$$

where $\boldsymbol{d} := [d_x \quad d_\psi]^\mathsf{T}$ is the disturbance vector.

**Assumption 1.** *The disturbances $\boldsymbol{d}(t) : \mathbb{R} \to \mathbb{R}^2$ are unknown, but continuous and bounded, such that*

$$\|d\|^2 \leq C_d, \tag{10}$$

*where $C_d > 0$ is constant.*

Using the kinematic error model in the body-frame for the system (9) in (8) gives

$$\begin{aligned} \dot{\tilde{x}}_b &= u_d \cos\tilde{\psi}_b - u_c + r_c \tilde{y}_b - d_x, \\ \dot{\tilde{y}}_b &= u_d \sin\tilde{\psi}_b - r_c \tilde{x}_b, \\ \dot{\tilde{\psi}}_b &= r_d - r_c - d_\psi, \end{aligned} \tag{11}$$

**Theorem 1.** *The closed loop system* (9)*, with the trajectory tracking control input* (6) *is globally uniformly ultimately bounded.*

*Proof.* Consider the Lyapunov function defined in [Section 5.4][49]

$$V = \frac{1}{2}\tilde{x}_b^2 + \frac{1}{2}\tilde{y}_b^2 + 1 - \cos\tilde{\psi}_b. \tag{12}$$

Taking the derivative of this gives

$$\begin{aligned}\dot{V} &= \tilde{x}_b\dot{\tilde{x}}_b + \tilde{y}_b\dot{\tilde{y}}_b + \dot{\tilde{\psi}}_b\sin\tilde{\psi}_b,\\ &= \begin{bmatrix}\tilde{x}_b & \tilde{y}_b & \sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}\dot{\tilde{x}}_b\\ \dot{\tilde{y}}_b\\ \dot{\tilde{\psi}}_b\end{bmatrix}.\end{aligned}$$

Using the kinematic error model (11) we get

$$\begin{aligned}\dot{V} &= \begin{bmatrix}\tilde{x}_b & \tilde{y}_b & \sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}u_d\cos\tilde{\psi}_b - u_c + r_c\tilde{y}_b\\ u_d\sin\tilde{\psi}_b - r_c\tilde{x}_b\\ r_d - r_c\end{bmatrix}\\ &\quad - \begin{bmatrix}\tilde{x}_b & \sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}d_x\\ d_\psi\end{bmatrix}.\end{aligned}$$

Inserting (6) into the latter equation gives

$$\dot{V} = -k_x\tilde{x}_b^2 - k_\psi\sin^2\tilde{\psi}_b - \begin{bmatrix}\tilde{x}_b & \sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}d_x\\ d_\psi\end{bmatrix}. \tag{13}$$

Applying Young's Inequality to this latter expression and taking $\|d\|^2 = d_x^2 + d_\psi^2$, under Assumption 1, we have

$$\begin{aligned}\dot{V} &\leq -k_x\tilde{x}_b^2 - k_\psi\sin^2\tilde{\psi}_b + \frac{1}{2}\tilde{x}_b^2 + \frac{1}{2}\sin^2\tilde{\psi}_b + \frac{1}{2}\|d\|^2,\\ &\leq -\left(k_x - \frac{1}{2}\right)\tilde{x}_b^2 - \left(k_\psi - \frac{1}{2}\right)\sin^2\tilde{\psi}_b + \frac{1}{2}\|d\|^2,\\ &\leq -\left(k_x - \frac{1}{2}\right)\tilde{x}_b^2 - \left(k_\psi - \frac{1}{2}\right)\sin^2\tilde{\psi}_b + \frac{1}{2}C_d.\end{aligned} \tag{14}$$

For a given initial value of $V$, it can be seen from (14), that a suitable selection of $k_x > 1/2$ and $k_\psi > 1/2$ will ensure that $\dot{V} \leq 0$ for a given value of $C_d$. Under Assumption 1, it can be seen from (13), that $\ddot{V}$ is bounded (so that $\dot{V}$ is uniformly continuous). Since $\dot{V} \leq 0$ at such an initial time, according to Barbalat's Lemma [50], the function $\dot{V} \to 0$ as $t \to \infty$ so that the solution trajectories of the closed loop system are ultimately bounded.

Therefore, we can conclude that both $\tilde{x}_b$ and $\tilde{\psi}_b$ are bounded. From (7), we see that this implies $\tilde{x}$, $\tilde{y}$, and $\tilde{\psi}$ are also bounded. As all the solution trajectories of the tracking errors of all the terms in the closed loop system are ultimately bounded, the closed loop system is uniformly ultimately bounded.

□

To implement (6) on a physical system, we need to ensure that the actuator magnitude limits are respected, in which the controller (6) magnitude increases when the error between the robot and the desired trajectory increases.
Let

$$V_\lambda = \frac{1}{2\lambda^2}\tilde{x}_b^2 + \frac{1}{2\lambda^2}\tilde{y}_b^2 + \frac{1}{\lambda}(1 - \cos\tilde{\psi}_b), \tag{15}$$

be a scaled Lyapunov function, where $\lambda : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, if we consider the following equation:

$$V_\lambda = c\,, \tag{16}$$

where $c$ is a positive constant, the positive solution of (16) is

$$\lambda = \frac{2(1 - \cos\tilde{\psi}_b) + \sqrt{\Delta}}{4c}, \tag{17}$$

where $\Delta = 4(1 - \cos\tilde{\psi}_b)^2 + 8c(\tilde{x}_b^2 + \tilde{y}_b^2)$.

Given the scaled Lyapunov function (15), the trajectory tracking control input (6), can be rewritten as

$$\boldsymbol{u}_\lambda = \begin{bmatrix}u_\lambda\\ r_\lambda\end{bmatrix} = \begin{bmatrix}u_d\cos(\tilde{\psi}_b) + \frac{1}{\lambda}k_x\tilde{x}_b\\ r_d + \frac{1}{\lambda^2}k_\psi\sin\tilde{\psi}_b + \frac{1}{\lambda}\tilde{y}_b u_d\end{bmatrix}. \tag{18}$$

The objective is to ensure the control inputs are bounded while smoothly minimizing the error between the robot and the desired trajectory.

**Theorem 2.** *The closed loop system* (9)*, with the scaled trajectory tracking control input* (18) *is globally uniformly ultimately bounded if the function $\lambda$ is chosen as*

$$\lambda = \begin{cases}1, & V \leq c\\ \dfrac{2(1 - \cos\tilde{\psi}_b) + \sqrt{\Delta}}{4c}, & V > c,\end{cases} \tag{19}$$

*where $c$ is a positive constant, and*

$$V = \frac{1}{2}\tilde{x}_b^2 + \frac{1}{2}\tilde{y}_b^2 + 1 - \cos\tilde{\psi}_b. \tag{20}$$

*Proof.* The first case when $\lambda = 1$, means the Lyapunov function $V = (\tilde{x}_b^2 + \tilde{y}_b^2)/2 + 1 - \cos\tilde{\psi}_b \leq c$, that is the error states $\tilde{x}_b$ and $\tilde{y}_b$ are bounded, then the control inputs (18) is bounded.

In the second case $\lambda$ is as defined in (19) and $V_\lambda = c$. We must prove that the scaled trajectory control input (18) asymptotically stabilizes (9).
The derivative of $V_\lambda = c$ is

$$\begin{aligned}\dot{V}_\lambda =& \frac{1}{\lambda^2}(\tilde{x}_b\dot{\tilde{x}}_b + \tilde{y}_b\dot{\tilde{y}}_b + \lambda\dot{\tilde{\psi}}_b\sin\tilde{\psi}_b)\\ &- \frac{\dot{\lambda}}{\lambda^3}(\tilde{x}_b^2 + \tilde{y}_b^2 + \lambda(1 - \cos\tilde{\psi}_b)) = 0.\end{aligned} \tag{21}$$

Inserting the kinematic error model for the system with disturbances (11) in the first term we have

$$\begin{aligned}\dot{V}_\lambda =& \frac{1}{\lambda^2}\left(\begin{bmatrix}\tilde{x}_b & \tilde{y}_b & \lambda\sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}u_d\cos\tilde{\psi}_b - u_\lambda + r_\lambda\tilde{y}_b\\ u_d\sin\tilde{\psi}_b - r_\lambda\tilde{x}_b\\ r_d - r_\lambda\end{bmatrix}\right)\\ &- \frac{1}{\lambda^2}\left(\begin{bmatrix}\tilde{x}_b & \lambda\sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}d_x\\ d_\psi\end{bmatrix}\right)\\ &- \frac{\dot{\lambda}}{\lambda^3}(\tilde{x}_b^2 + \tilde{y}_b^2 + \lambda(1 - \cos\tilde{\psi}_b)) = 0,\end{aligned}$$

Using the control input (18), we have

$$\begin{aligned}&\frac{\dot{\lambda}}{\lambda^3}(\tilde{x}_b^2 + \tilde{y}_b^2 + \lambda(1 - \cos\tilde{\psi}_b)) =\\ &- \frac{1}{\lambda^3}\left(k_x\tilde{x}_b^2 + k_\psi\sin^2\tilde{\psi}_b\right) - \frac{1}{\lambda^2}\left(\begin{bmatrix}\tilde{x}_b & \lambda\sin\tilde{\psi}_b\end{bmatrix}\begin{bmatrix}d_x\\ d_\psi\end{bmatrix}\right)\end{aligned} \tag{22}$$

Applying Young's inequality to the vectors

$$\lambda \begin{bmatrix} \frac{1}{\lambda^3}\tilde{x}_b & \frac{1}{\lambda^2}\sin\tilde{\psi}_b \end{bmatrix} \begin{bmatrix} d_x \\ d_\psi \end{bmatrix},$$

we have

$$\lambda \begin{bmatrix} \frac{1}{\lambda^3}\tilde{x}_b & \frac{1}{\lambda^2}\sin\tilde{\psi}_b \end{bmatrix} \begin{bmatrix} d_x \\ d_\psi \end{bmatrix} \leq \frac{1}{2\lambda^6}\tilde{x}_b^2 + \frac{1}{2\lambda^4}\sin^2\tilde{\psi}_b + \lambda(\frac{1}{2}d_x^2 + \frac{1}{2}d_\psi^2) \tag{23}$$

inserting (23) into (22) we have

$$\begin{aligned} &\frac{\dot{\lambda}}{\lambda^3}(\tilde{x}_b^2 + \tilde{y}_b^2 + \lambda(1-\cos\tilde{\psi}_b)) \leq \\ &-\frac{1}{\lambda^3}\left(k_x - \frac{1}{2\lambda^3}\right)\tilde{x}_b^2 - \frac{1}{\lambda^3}\left(k_\psi - \frac{1}{2\lambda}\right)\sin^2\tilde{\psi}_b + \lambda(\frac{1}{2}d_x^2 + \frac{1}{2}d_\psi^2). \end{aligned} \tag{24}$$

Given Assumption 1, the disturbance magnitude is bounded by $C_d$. Thus, by carefully selecting the value $c$ in (17), we have

$$(d_x^2 + d_\psi^2) \leq C_d \leq \frac{1}{2\lambda^4}\tilde{x}_b^2.$$

Then, we have

$$\dot{\lambda} = -\frac{\left(k_x - \frac{1}{2} - \frac{1}{2\lambda^3}\right)\tilde{x}_b^2 + \left(k_\psi - \frac{1}{2\lambda}\right)\sin^2\tilde{\psi}_b}{\tilde{x}_b^2 + \tilde{y}_b^2 + \lambda(1-\cos\tilde{\psi}_b)} < 0, \tag{25}$$

which implies that for $k_x > 1/2 + 1/(2\lambda^3)$ and $k_\psi > 1/(2\lambda)$ the function $\lambda$ is decreasing, and from its definition (19), we conclude that the pose errors in the body frame are decreasing. By selecting the constant c, we can ensure that the control input (18), is globally uniformly ultimately bounded. □

### *B. Safety-Critical Control*

The collision avoidance controller from [38] ensures that the safe set is forward invariant; however, the cosine in the definition of the control barrier function causes the control input to drop to zero when $|\tilde{\psi}_b| = \pi/2$ and the robot is avoiding an obstacle. To prevent this problem, we instead use the $\tanh(\cdot)$ function in the control input, which requires the relaxed term in the control barrier function C to be greater than the parameter $k_{\text{CBF}}$.

During the experiments presented in [38] it was noted that the condition $C > k_{\text{CBF}}$ makes the tuning of the controller more difficult. Thus, we design a collision avoidance controller that does not require $C > k_{\text{CBF}}$.

**Proposition 1.** *The following function*

$$B(\boldsymbol{x}, t) = \sum_{i=1}^{n}\left(\frac{1}{(x-x_{oi})^2 + (y-y_{oi})^2 - r_{safe,i}^2}\right), \tag{26}$$

*is a relaxed time-varying control barrier function, where $\boldsymbol{x}$ is the robot pose.*

*Proof.* The function (26) is a relaxed time-varying control barrier function if it satisfies the three conditions in Definition 1 of [38]. The function (26) already satisfies the first and second conditions in Definition 1 of [38].

To prove that (26) also satisfies the third condition, we need to show that $\dot{B}(\boldsymbol{x}, \boldsymbol{u}_c, \boldsymbol{u}_h, t) < KB + C$.

The derivative of the function $B(\boldsymbol{x}, t)$ is

$$\begin{aligned} \dot{B}(\boldsymbol{x}, \boldsymbol{u}_c, \boldsymbol{u}_h, t) &= \sum_{i=1}^{n}\left(\frac{\partial B}{\partial x}\dot{x} + \frac{\partial B}{\partial y}\dot{y} + \frac{\partial B}{\partial \psi}\dot{\psi} + \frac{\partial B}{\partial \boldsymbol{x}_{oi}}\dot{\boldsymbol{x}}_{oi}\right), \\ &= L_g B(\boldsymbol{u}_c + \boldsymbol{u}_o) + \sum_{i=1}^{n}\frac{\partial B}{\partial \boldsymbol{x}}_{oi}\dot{\boldsymbol{x}}_{oi}, \end{aligned} \tag{27}$$

with

$$L_g B = \begin{bmatrix} L_g B_0 \\ L_g B_1 \end{bmatrix} = \sum_{i=1}^{n}\left(\begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} & \frac{\partial B}{\partial \psi} \end{bmatrix}\boldsymbol{g}(\boldsymbol{x})\right) \in \mathbb{R}^2, \tag{28}$$

and

$$\sum_{i=1}^{n}\frac{\partial B}{\partial \boldsymbol{x}}_{oi}\dot{\boldsymbol{x}}_{oi} = \sum_{i=1}^{n}\frac{-2((x-x_{oi})\dot{x}_{oi} + (y-y_{oi})\dot{y}_{oi})}{((x-x_{oi})^2 + (y-y_{oi})^2 - r_i^2)^2}, \tag{29}$$

where $\dot{x}_{oi}$ and $\dot{y}_{oi}$ are the speed of the $i^{\text{th}}$ obstacle in the $x$ direction and $y$ direction respectively.

From (2), and since $\partial B/\partial\psi = 0$, $L_g B_1 = 0$. Let $I \in \mathbb{R}$ and $J \in \mathbb{R}$ be defined as

$$I := L_g B \boldsymbol{u}_c + \sum_{i=1}^{n}\frac{\partial B}{\partial \boldsymbol{x}}_{oi}\dot{\boldsymbol{x}}_{oi}, \tag{30}$$

and

$$J := KB + C. \tag{31}$$

If $I \leq J$ and if $\boldsymbol{u}_o = 0$, then

$$\dot{B}(\boldsymbol{x}, \boldsymbol{u}_c, \boldsymbol{u}_h, t) = I < KB + C$$

If $I > J$ with

$$\boldsymbol{u}_o = -\frac{I}{\|L_g B\|^2}L_g B^{\mathsf{T}} - \begin{bmatrix} 0 \\ k_{\text{CBF}} r_{\max}\tanh\left(\frac{H}{r_{\max}}\right) \end{bmatrix},$$

where $r_{\max}$ is the maximum value of the yaw rate, and

$$H = \frac{(I-J)}{\|L_g B\|^2}L_g B_o,$$

then

$$\dot{B}(\boldsymbol{x}, \boldsymbol{\tau}, \boldsymbol{u}, t) = 0 < KB + C,$$

and the third condition is satisfied. □

Based on the proof of Proposition 1 and on the Proposition 2 in [38], the collision avoidance control

$$\boldsymbol{u}_o = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathsf{T}}, & I \leq J \\ -\frac{I}{\|L_g B\|^2}L_g B^{\mathsf{T}} - \begin{bmatrix} 0 \\ k_{\text{CBF}}\boldsymbol{r}_{\max}\tanh\left(\frac{H}{\boldsymbol{r}_{\max}}\right) \end{bmatrix}, & I > J \end{cases} \tag{32}$$

ensures the graph set defined in [38] is invariant. Thus, if the robot starts inside the safe set, it remains inside the safe set.

The collision avoidance controller (32) is added to the trajectory tracking controller (18) to form the automatic controller, which is given by

$$\boldsymbol{u}_a = \boldsymbol{u}_\lambda + \boldsymbol{u}_o\ . \tag{33}$$

### C. Shared Control

The human operator uses a joystick to provide commands to the UGV while getting a feedback force. The automatic control input $\boldsymbol{u}_a$ is mapped to the normalized surge speed and yaw rate $u_m, r_m \in \big[-1, 1\big]$, respectively. The normalization constants $k_{jx}, k_{j\psi}$ represent the maximum surge speed and yaw rate from the automatic controller, respectively.

$$u_m = u_a / k_{jx} \tag{34}$$

$$r_m = r_a / k_{j\psi}. \tag{35}$$

The actual joystick positions pitch and roll $u_{\text{joy}}, r_{\text{joy}} \in \big[-1, 1\big]$ are also normalized, with 0 corresponding to the rest position. The joystick is programmed to follow the normalized surge speed $u_m$ and yaw rate $r_m$, respectively. The errors between the mapped automatic control inputs and the actual joystick positions are also defined as $\tilde{x}_h = u_m - u_{\text{joy}}$ and $\tilde{\psi}_h = r_m - r_{\text{joy}}$, respectively.

The human control input is:

$$\boldsymbol{u}_h = \begin{bmatrix} u_h \\ r_h \end{bmatrix} = \begin{bmatrix} k_{hx}\tilde{x}_h \\ k_{h\psi}\tilde{\psi}_h \end{bmatrix} \tag{36}$$

where $k_{hx},\ k_{h\psi} > 0$ are positive constants.

The human intent is derived using the storage functions

$$V_{hx} = \frac{1}{2} k_s \tilde{x}_h^2 \geq 0, \tag{37}$$

$$V_{h\psi} = \frac{1}{2} k_s \tilde{\psi}_h^2 \geq 0, \tag{38}$$

where $k_s > 0$ is a spring constant. When the joystick handle is displaced from the tracked position in the front-to-back direction $\tilde{x}_h \neq 0$, $V_{hx} > 0$, and when it is displaced in the side-to-side direction $\tilde{\psi}_h \neq 0$, $V_{h\psi} > 0$. Thus, the values of the two storage functions are convenient measures of the human's intent to control the system.

The human control input (36) is mixed with the automatic control input (33) using a linear blending law such that

$$\boldsymbol{u}(t) = \boldsymbol{K}_h \boldsymbol{u}_h(t) + \boldsymbol{K}_c \boldsymbol{u}_a(t), \tag{39}$$

where

$$\boldsymbol{K}_h = \begin{bmatrix} (1 - \mathrm{e}^{-V_{hx}}) & 0 \\ 0 & (1 - \mathrm{e}^{-V_{h\psi}}) \end{bmatrix} \tag{40}$$

$$\boldsymbol{K}_c = \begin{bmatrix} \mathrm{e}^{-V_{hx}} & 0 \\ 0 & \mathrm{e}^{-V_{h\psi}} \end{bmatrix}. \tag{41}$$

The variations of $\tilde{x}_h$, $\tilde{\psi}_h$, are due only to human intervention and cause a change in the potentials $V_{hx}$ and $V_{h\psi}$, which are proxies for human intent. The blending parameters $\boldsymbol{K}_h$, $\boldsymbol{K}_c$ in (40), (41) are designed so that when there is a human intervention, $V_{hx}$ and $V_{h\psi}$ will increase making the human input $\boldsymbol{u}_h(t)$ dominate the automatic control input $\boldsymbol{u}_a(t)$. Therefore, the human input always has priority.

Note that the exponential function in (40) and (41) ensures a smooth transition between the human commands and the automatic control, while the speed of transition depends on the value of $k_s$.

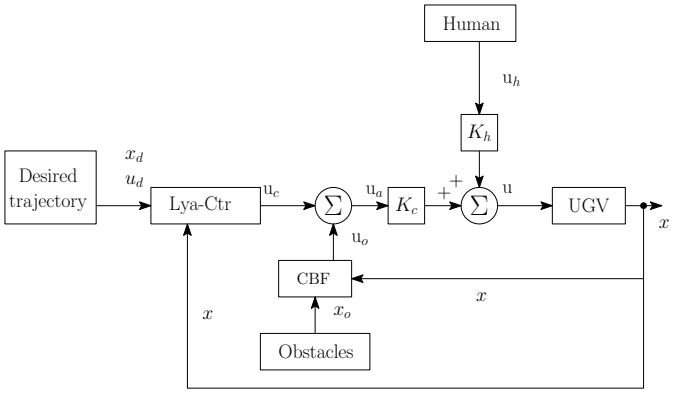



Fig. 3. Block diagram of the shared control system with collision avoidance, the blue part of the figure represents the block diagram of the shared control system without collision avoidance, where Lya-Ctr is the trajectory tracking controller $\boldsymbol{u}_c$, CBF the collision avoidance controller $\boldsymbol{u}_o$, $\boldsymbol{x}$ is the robot pose in the NED-frame, the robot desired surge speed $u_d$, the desired pose $\boldsymbol{x}_d$, and the obstacle position $\boldsymbol{x}_o$.

### D. Reference Time scaling

User intervention can deviate the actual position of the robot from its desired position. If this deviation increases too much, the robot might follow an undesirable path. For example, the human could force the robot to stop for a certain amount of time to examine an object of interest. If, during that time, the desired trajectory moves to the next row of trees, the robot may try to pass through the line of trees when the human relinquishes control to the automatic controller.

To address this, we implemented a time scaling strategy that gradually reduces (or increases) the rate of change of the desired trajectory if the error goes beyond a predefined threshold. To this end, we use a time parametrization $h(t)$ for the desired trajectory that differs from the controller time $t$. We assume the desired trajectory is generated with an interpolant (e.g. a polynomial) parametrized with $h(t)$ instead of $t$. While the time $t$ evolves at a constant rate $F_c$ (where $F_c$ is the frequency of the control loop) $h(t)$ can vary its evolution depending on the tracking error. This will change the evolution of the desired trajectory.

First, we consider the case where the desired position lies in the half-space ahead of the robot. In this case, the time scaling parameter $\epsilon$ will adjust the time update in order to "slow down" the trajectory. This is equivalent to having $\tilde{x}_b > 0$ and $u_d > 0$. Without loss of generality, similar reasoning can be done to speed up the desired position if $\tilde{x}_b < 0$ (i.e., the robot is ahead of the desired position). The parametrization $h(t)$ is defined (in continuous time) as

$$h(t) = \int \epsilon(\|{}_b\boldsymbol{e}_{xy}(t)\|) \frac{1}{F_c}, \tag{42}$$

where $\|{}_b\boldsymbol{e}_{xy}\| = \sqrt{\tilde{x}_b^2 + \tilde{y}_b^2}$ is the norm of the tracking error at time $t$ along the $X$, $Y$ coordinates (i.e. Cartesian error) expressed in the base frame, $\epsilon \in [0, 1]$ scales the nominal time update which is set equal to the control interval $1/F_c$. Now, let us consider $\|{}_b\boldsymbol{e}_{xy}\|_{\max}$ the norm of the maximum error beyond which we want the reference to stop. Then, we can define $\epsilon$

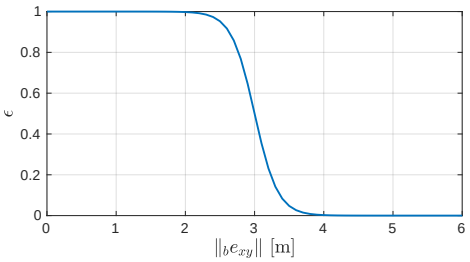



Fig. 4. Plot of the time scaling function, for activation $a = 2m$ and maximum error $\|_b \boldsymbol{e}_{xy}\|_{\max} = 4m$.

as a function of the norm of the Cartesian error $\|_b \boldsymbol{e}_{xy}\|$,

$$\epsilon(t) = \frac{1}{2} + \frac{1}{2}\tanh\left[s\left(\rho - \|_b \boldsymbol{e}_{xy}(t)\|\right)\right], \tag{43}$$

where $\rho = a + (\|_b \boldsymbol{e}_{xy}\|_{\max} - a)/2$ sets the position of the middle of the slope, $a$ is the activation value and $s$ is a sensitivity parameter that we set to $s = 3/(\rho - a)$ such that the slope starts to decrease at exactly $a$. The plot of $\epsilon$ is shown in Fig. 4.

### *E. Model Predictive Controller*

To compare our control strategy with a state-of-the-art approach, we opted for a model predictive controller based on numerical optimization as baseline [51]. In particular, we implemented a Nonlinear Model Predictive Control (NMPC) formulation because it can handle both the nonlinear system dynamics and the constraints. We use a *single shooting* approach to transcribe the optimal control problem where we discretize the vector of inputs $\boldsymbol{U}$ along a *prediction horizon* $T$, which is divided into $N$ discrete time control intervals of length $T_{\mathrm{s}} = T/N$.

The states $\boldsymbol{x}(k) \in [0, N]$ are treated as *dependent* variables. Therefore, the decision variables are the vector of control inputs $\boldsymbol{U} = \left[\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}\right]$ to be given to (1), with $\boldsymbol{u}_k = [u_k \;\; r_k]^{\mathsf{T}}$, while the states are $\boldsymbol{x}_k = [x_k \;\; y_k \;\; \psi_k]^{\mathsf{T}} \in \mathbb{R}^3$ and are computed from the inputs along the horizon. The resulting optimization problem is

$$\min_{\boldsymbol{U}} \quad \sum_{k=0}^{N-1} \ell\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \ell_{\mathrm{T}}\left(\boldsymbol{x}_N\right) \tag{44a}$$

$$\text{s.t.} \quad \boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0, \tag{44b}$$

$$\boldsymbol{x}_{k+1} = f\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \qquad k \in \mathbb{I}_0^{N-1}, \tag{44c}$$

$$h\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \leq 0, \qquad k \in \mathbb{I}_0^{N-1}, \tag{44d}$$

where $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the stage cost function, and $\ell_{\mathrm{T}} : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost function. The initial condition (44b) is expressed by setting $\boldsymbol{x}_0$ equal to the state estimate $\hat{\boldsymbol{x}}_0$ received from an estimator (see Sec. IV-F).

Finally, the inequality constraints are included with (44d). They provide bounds on the decision variables and path constraints to implement obstacle avoidance. The nonlinear rover dynamics (1) is integrated to obtain the states in a single-shooting fashion via a fourth-order method (i.e. Runge Kutta 4) starting from the initial state at sample 0: $\boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0$.

To reduce the impact of integration errors, we integrate on a finer grid, performing a number of integration $N_{\text{sub}}$ sub-steps within two adjacent knots. This has the advantage of improving the integration accuracy, without increasing the problem size. In our NMPC formulation we do not set any terminal cost but only a stage cost of the form:

$$\ell\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) = \ell_{\mathrm{p}} + \ell_{\mathrm{v}} + \ell_{\mathrm{s}}, \tag{45a}$$

$$\ell_{\mathrm{p}} = \| \boldsymbol{x}_k - \boldsymbol{x}_{d,k} \|^2_{\mathbf{Q}_p} \tag{45b}$$

$$\ell_{\mathrm{v}} = \| \boldsymbol{u}_k - \boldsymbol{u}_{d,k} \|^2_{\mathbf{Q}_v}, \tag{45c}$$

$$\ell_{\mathrm{s}} = \| \boldsymbol{u}_k - \boldsymbol{u}_{k-1} \|^2_{\mathbf{Q}_s}, \tag{45d}$$

where:

- the first term (45b) is a tracking cost to encourage the states $\boldsymbol{x}_k$ to track their references $\boldsymbol{x}_{d,k}$, at each sample k
- the second term (45c) is a tracking cost to track desired inputs $\boldsymbol{u}_{d,k} = [u_d(t_k) \;\; r_d(t_k)]^{\mathsf{T}}$, and
- the third term (45d) is a cost to smooth the inputs (i.e. minimize their first derivatives).

The positive definite weight matrices $\mathbf{Q}_p \in \mathbb{R}^{3\times3}$, $\mathbf{Q}_v \in \mathbb{R}^{2\times2}$, and $\mathbf{Q}_s \in \mathbb{R}^{2\times2}$ contain the weights of the different cost terms. For the inequalities (44d) we encode: 1) the bounds on the input variables ($|u| \leq u_{\max}$ and $|r| \leq r_{\max}$) and 2) implement obstacles avoidance as

$$\left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \end{bmatrix} \right\|^2 \geq r^2_{\text{safe}}, \tag{46}$$

where $\begin{bmatrix} x_o & y_o \end{bmatrix}^{\mathsf{T}}$ is the position of the closest obstacle and $r_{\text{safe}}$ is the minimum distance we want the rover to keep from the obstacle. NMPC is based on solving the Optimal Control Problem (OCP) (44) given the current state estimate $\hat{\boldsymbol{x}}_0$ of the system. Only the first element of the optimized input trajectory is applied to the system, then the state is measured and the OCP is solved again (shifting the horizon forward) based on the new state measurement, to close the loop.

### *F. State Estimation*

The controllers detailed in the previous section do not have direct access to the robot's state $\boldsymbol{x}$. To make an estimated state $\hat{\boldsymbol{x}}$ available to the controllers, we fuse Inertial Measurement Unit (IMU) and Global Navigation Satellite System (GNSS) data. A 9-axis IMU provides the robot's heading at 200 Hz via an internal filter running in hardware. The robot's position is provided by a GNSS receiver at 10 Hz with Real Time Kinematics (RTK) corrections via Networked Transport of RTCM via Internet Protocol (NTRIP), which guarantees 20 mm accuracy.

A simple custom ROS node directly combines the two datastreams at 10 Hz after converting the absolute coordinates provided by the GNSS to Cartesian coordinates via a fixed transform. In our experience, there was no need to perform any data fusion (e.g. via filtering).

This approach reduces the complexity, computational load, and uncertainty associated to the processing of high bandwidth data (e.g., cameras, lidars) in agricultural settings. At the same

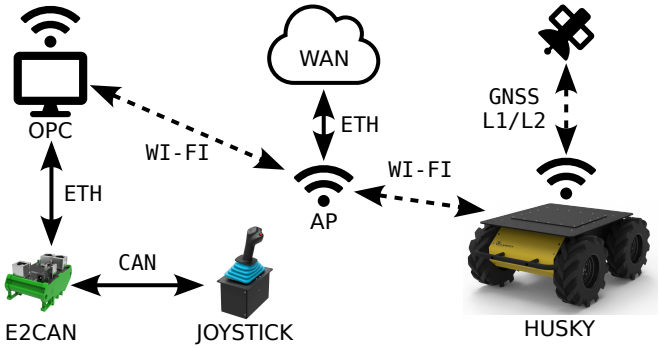



Fig. 5. Overview of the experimental setup. An Operator PC (OPC) is connected to the joystick via an Ethernet-to-CAN gateway, an Access Point (AP) provides internet access to the OPC and the robot via Wi-Fi at 5 GHz. The robot communicates to the GNSS over L1 and L2 bands and computes its RTK position using NTRIP corrections from the internet.

time, it leverages the prior information (e.g., maps of the orchard) and GNSS coverage that are available in most agriculture applications. However, in GNSS-denied environments, adding a SLAM system is necessary; this is left for future work.

## V. Experimental Setup

The human-robot shared control setup consists of a mobile platform, a teleoperation base station, and a communication infrastructure. An overview of the setup is shown in Fig. 5. The operator's PC at the base station communicates with a Joystick via an Ethernet-to-CAN gateway. The Wi-Fi range between the PC and the robot is extended by placing an Access Point (AP) at the center of the field, which also provides internet access to the robot and the Operator's PC (OPC).

### A. Mobile platform

The mobile platform used for our experiments is the Clearpath Husky A200, a 1 m long, four-wheel differential drive robot with a 75 kg payload capacity and a weight of 50 kg. The robot is shown in Fig. 6. It was equipped with an ArduSimple simpleRTK2B real-time kinematic (RTK) GNSS receiver, and a 9-axis Xsens Mti-630 IMU. A 5 Ghz Wi-Fi antenna is placed at the top for communication with the local RTK corrections network, telemetry, and teleoperation from the base station.

### B. Teleoperation station and communication infrastructure

The teleoperation station consists of the OPC connected to a CLS-P Side Stick Active Force Joystick by Brunner Elektronik. The Joystick communicates with the PC via a E2CAN Gateway and is configured to output pitch and roll signals. The pitch and roll are converted into ROS messages and mapped to command the surge speed $u$ and yaw rate $r$ of the robot, respectively.

When the trajectory tracking controller is active, the velocity corresponding to the desired trajectory is sent to the joystick, which is driven in autopilot mode. The joystick uses the in-built force feedback to indicate the user deviations from the velocity associated with the reference trajectory.

To maintain a stable 5 GHz Wi-Fi connection between the operator's PC and a robot, a WAVLINK-N300 dual-band Access Point (AP) was mounted on a support column in the middle of the experimental site. The AP also provided the robot with NTRIP RTK access, connected to the internet via a CAT6 Ethernet cable.

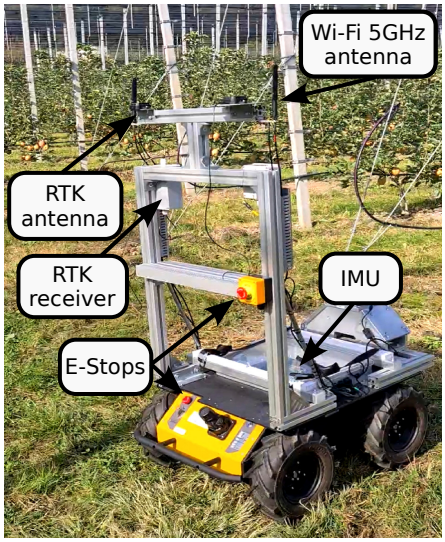



Fig. 6. The Husky A200 used for the experiments. The system was equipped with two Wi-Fi antennas, an RTK GNSS, an industrial-grade IMU, and two emergency stops. To limit interference, only the 5 GHz antenna was used.

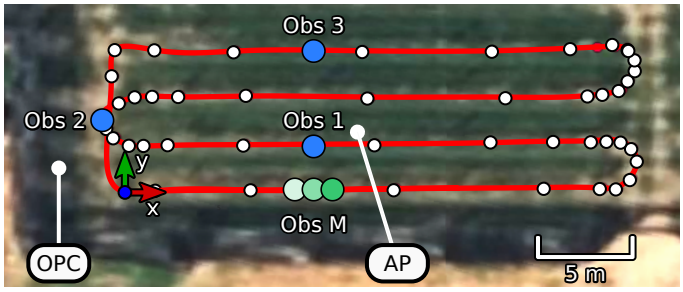



Fig. 7. Aerial view of the of the experimental site. The reference trajectory is indicated in red, the GNSS waypoints in white, the static obstacles in blue, and the moving obstacle (from left to right) in green. The robot starts at the origin of the inertial reference frame (the $X$-$Y$ axes) and moves towards the moving obstacle first and the static ones in sequential order.

### C. Experimental site

The experiments were conducted at the Laimburg Integrated Digital Orchard (LIDO), an experimental apple orchard at the Laimburg Research Center in Ora (BZ), Italy. The test site consists of five 25 m-long rows of trees, separated by 2.7 m-wide aisles. A satellite view of the site, with the desired trajectory performed in the experiments and the obstacles (tagged *in situ* with RTK GNSS), is shown in Fig. 7.

## VI. Experimental Evaluation

To evaluate our proposed controllers, we performed a series of eleven experiments, each one consisting of one or more trials, in a simulated and a real orchard environment (see Table I). For all tests, we consider the scenario in which the robot autonomously navigates the orchard, e.g., for a routine

TABLE I
SUMMARY OF THE EXPERIMENTAL VALIDATION

| Type | Appr. | Exp. N | Trials | Sh. C | Obs. A | $\lambda$-S | t-S |
|---|---|---|---|---|---|---|---|
| Sim. | Lyap. | Exp 1 | 5 | | | ✓ | ✓ |
| | | Exp 1A | 5 | ✓ | | ✓ | |
| | | Exp 1B | 5 | ✓ | | ✓ | ✓ |
| | | Exp 2 | 5 | ✓ | | ✓ | ✓ |
| | | Exp 2A | 5 | ✓ | ✓ | ✓ | ✓ |
| | MPC | Exp 3 | 5 | | | N/A | ✓ |
| | | Exp 4 | 5 | ✓ | | N/A | ✓ |
| Real | Lyap. | Exp 5 | 10 | | | | ✓ |
| | | Exp 6 | 10 | | | ✓ | ✓ |
| | | Exp 7 | 5 | | ✓ | ✓ | ✓ |
| | | Exp 8 | 1 | ✓ | ✓ | ✓ | ✓ |

spraying or monitoring operation in the potential presence of moving and stationary obstacles, as shown in Fig. 7.

The operator follows the robot's progress from the OPC at a distance, with the ability to manually stop it or move it away from the reference trajectory by using the joystick (e.g. to inspect a tree of interest). For the field experiments, the reference trajectory was created from a set of GNSS waypoints (white dots in Fig. 7). The same trajectory was replicated in the simulated environment.

Unless otherwise stated, we placed three static obstacles (blue dots in Fig. 7) on the reference trajectory, and one virtual obstacle moving along the $X$-axis with speed $0.35\,\mathrm{m/s}$ (darkening green dots in Fig. 7). All obstacles are modeled as infinitely tall circular cylinders with radius $r_{\text{safe}} = 0.6\,\mathrm{m}$.

We mentioned that the collision avoidance proposed in this work is a reactive controller; thus, prior knowledge of the number of obstacles in the experimental environment is not required. However, when the UGV gets close to an obstacle, a safe area around it is required.

### A. Simulation Results

*Experiment 1 – Lyapunov-based approach, $\lambda$-Scaling:* In this experiment, we validate the trajectory tracking controller with the scaling function $\lambda$ (18). We placed the starting point of the reference trajectory at (0, 10) and the robot at (0, 0) to have a position error at the start of the simulations. First, the robot tracks the reference trajectory represented by the red line (see Fig.8) without the scaling $\lambda$, in which the blue line represents the robot trajectory, second the robot tracks the reference trajectory (see Fig.8) with the $\lambda$-Scaling active, where the green line represents the robot trajectory. The actual and the reference trajectories are presented in Fig. 8, with the surge speed and the yaw-rate, which shows how the $\lambda$-Scaling reduces the magnitude of the control input when the function (17) is greater than 1 (from time 0 s to time 24 s), in which the robot without geometric scaling reached the reference trajectory faster than when the scaling function is active, but the maximum surge speed is 5.8 m/s while the maximum scaled surge 1m/s, similarly the maximum yaw rate 6.5 rad/s while the maximum scaled yaw rate is 1 rad/s .

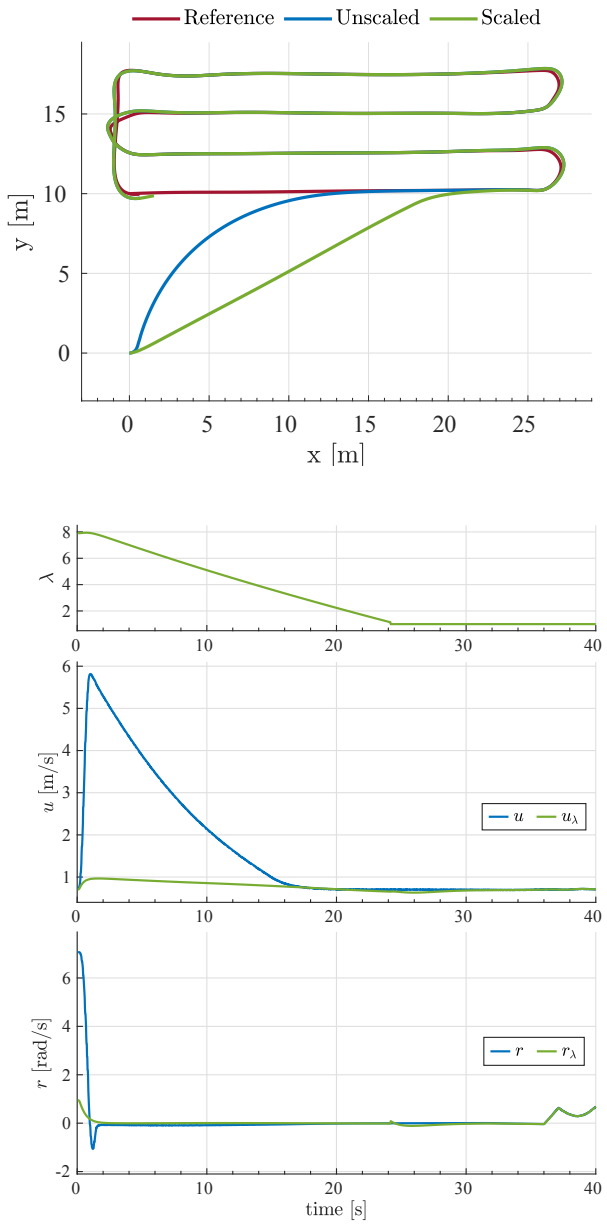



Fig. 8. Experiment 1 results. From top to bottom: a) reference and actual trajectories (Unscaled: is the trajectory with no scaling, Scaled: is trajectory with scaling) b) scaling function $\lambda$ c) unscaled command surge speed $u$ and scaled surge speed $u_\lambda$ d) unscaled yaw rate $r$ and scaled yaw rate $r_\lambda$

*Experiment 1A – Shared control Lyapunov-based approach, without t-Scaling:* In this experiment (see Fig. 9), we want to show a scenario of the shared controller with no time scaling in the reference trajectory. In this scenario the human operator first commands the UGV to stop for 15s between $t$=25s and $t$=40s, then again for 10s between $t$=70s and $t$=80s, and finally for 20s from $t$=140s and $t$=160s. When the human operator commands the UGV as indicated when $e^{-V_h} \approx 0$ (Fig. 9, top time-series plot), the reference trajectory keeps evolving.

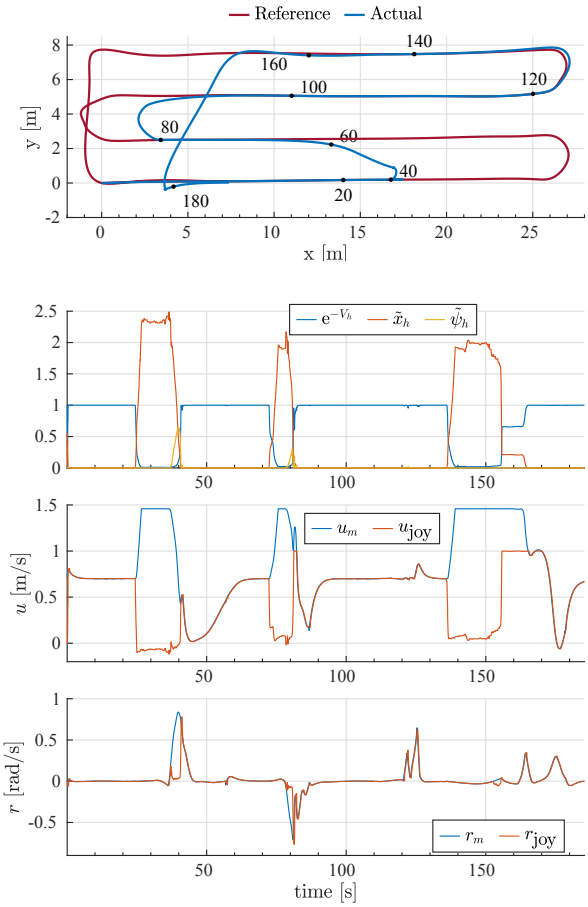



Fig. 9. Experiment 1A results. From top to bottom: a) reference and actual trajectories b) Negative exponential of the activation function $V_h$ and $\tilde{x}_h$, $\tilde{\psi}_h$ errors between the mapped automatic control and the mapped joystick position c) the mapped surge speed $u_m$ and the mapped joystick pitch $u_{joy}$ d) the mapped yaw-rate $r_m$ and the mapped joystick roll $r_{joy}$. The operator commands the UGV when the negative exponential of the activation function $V_h$ is less than 1.

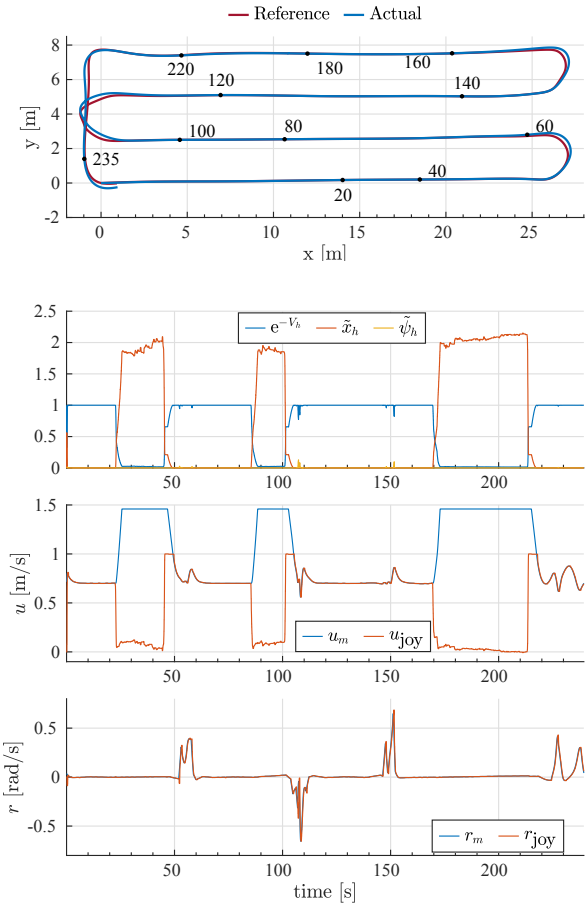



Fig. 10. Experiment 1B results. From top to bottom: a) reference and actual trajectories b) Negative exponential of the activation function $V_h$ and $\tilde{x}_h$, $\tilde{\psi}_h$ errors between the mapped automatic control and the mapped joystick position c) the mapped surge speed $u_m$ and the mapped joystick pitch $u_{joy}$ d) the mapped yaw-rate $r_m$ and the mapped joystick roll $r_{joy}$. The operator commands the UGV when the negative exponential of the activation function $V_h$ is less than 1.

When the UGV is released, the automatic controller takes a shortcut to reduce the error between the UGV position and the reference trajectory. We illustrate the experiment results in Fig. 9, in which you can see the shortcuts the UGV is taking.

*Experiment 1B – Lyapunov-based approach, with t-Scaling:* In this experiment, we repeat Experiment 1A with the time scaling, showing the benefits of its use. As in the previous experiment we stopped the UGV, but first for 25s between $t$=25s and $t$=50s, then again for 15s between $t$=85s and $t$=100s, and finally for 45s between $t$=170s and $t$=215s. The experiment results illustrated in Fig. 10 demonstrate the practical advantage of time scaling: allowing a human operator to stop the UGV for an extended period, while allowing the UGV to return to the desired trajectory without undesirable shortcuts–which would, in practice, cross rows of trees.

*Experiment 2 – Lyapunov-based approach, Shared Control:* In this experiment, we tested the *shared* trajectory tracking controller (39) in presence of obstacles. The automatic part is the trajectory tracking controller (18). The static and moving obstacles on the reference trajectory were manually avoided by the user with the joystick. This is reflected in the activation function, which indicates the human intention (when $e^{-V_h} \approx 0$, see Fig. 11). Without user input, the joystick follows the trajectory-tracking controller commands. Thanks to the force feedback, the user feels a resistance proportional to the tracking errors $\tilde{x}_h, \tilde{\psi}_h$ when the robot is commanded through the joystick.

*Experiment 2A – Lyapunov-based approach, Shared Control with Collision Avoidance:* This demonstration shows how the shared trajectory tracking *with* collision avoidance control performs in a more complex scenario with two moving obstacles. The first one moves at a speed of 0.35m/s in the same direction as the UGV, and the second is moves a speed of 1.0m/s towards the UGV. The static obstacles Obs2 and Obs3 are unchanged from the other experiment (e.g. Fig. 11), while Obs1 is moved to the end of the first row at (27, 1.8). The first moving obstacle follows a straight line on the X-axis direction starting from (0, 5), the same for the second moving obstacle, but it starts from (-10, 2.6).

The simulation scenario is as follows: the UGV first

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3554368



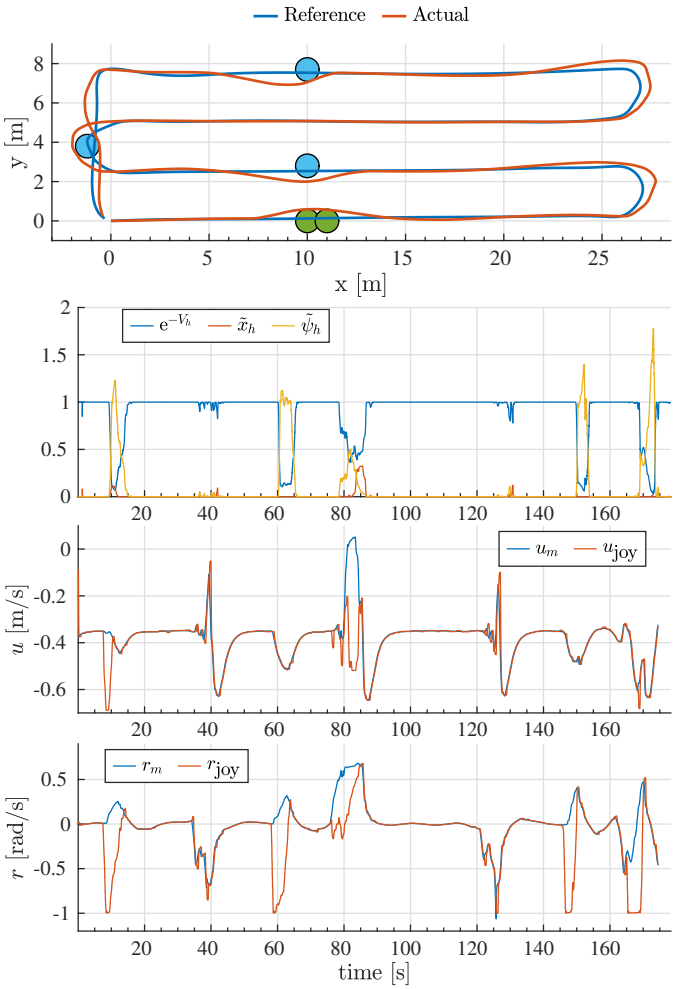



Fig. 11. Experiment 2 results. From top to bottom: a) reference and actual trajectories with obstacles. b) Negative exponential of the activation function $V_h$ and $\tilde{u}_{\text{joy}}$, $\tilde{r}_{\text{joy}}$ errors between the human inputs and the outputs of the controller, respectively c) surge speed $u$ and joystick position $u_{\text{joy}}$ d) yaw rate $r$ and joystick position $r_{\text{joy}}$. The intention of the operator trying to avoid the obstacles is visible when the roll becomes negative while $r$ remains positive.

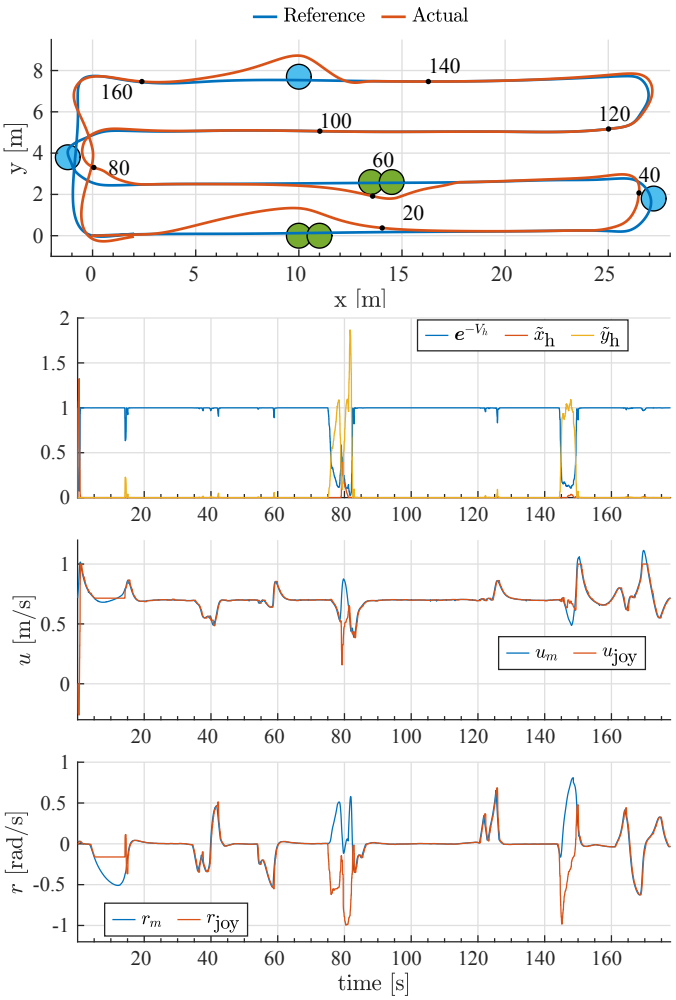



Fig. 12. Experiment 2A results. From top to bottom: a) reference and actual trajectories with obstacles. b) Negative exponential of the activation function $V_h$ and $\tilde{x}_h$, $\tilde{\psi}_h$ errors between joystick positions and mapped control inputs, respectively c) mapped surge speed $u_m$ and joystick pitch $u_{\text{joy}}$ d) mapped yaw rate $r_m$ and joystick roll $r_{\text{joy}}$. The operator's intention to avoid obstacles is evident when the $r_{\text{joy}}$ becomes negative while $r_m$ remains positive.

encounters the moving obstacle with a speed of 0.35 m/s, then the static obstacle Obs1, and then the second moving obstacle, where the UGV autonomously avoids collision. After that, the UGV encounters Obs2 and Obs3; in both cases, we *manually* force the UGV to avoid collision to the right of the obstacle. (By design, the controller avoids collisions to the left). The result of the demonstration is illustrated in Fig. 12, where we can see the trajectory of the UGV avoiding collision with the obstacle

*Experiments 3 and 4 – Model Predictive Controller:* For comparison, we repeated the conditions of Experiments 1 and 2, but this time with the NMPC presented in Section IV-E as the trajectory tracking controller. In both cases we used an horizon of $N = 15$ and a time step of $T_s = 0.1s$. The cost weights have been tuned to achieve the desired performance and are reported in Table II.

The optimization is implemented in Matlab using the Sequential Quadratic Programming (SQP) solver and a $4^{\text{th}}$ order Runge–Kutta integration method. To improve performance, we used a C++ implementation of the OCP. The NMPC

TABLE II
MODEL PREDICTIVE CONTROL PARAMETERS

| Parameter | Value | Description |
|---|---|---|
| $w_x$ | 10 | Weight for the $x$ tracking cost |
| $w_y$ | 5 | Weight for the $y$ tracking cost |
| $w_\psi$ | 10 | Weight for the $\psi$ tracking cost |
| $w_u$ | 1 | Weight for tracking desired lin. vel. |
| $w_r$ | 1 | Weight for tracking desired ang. vel. |
| $w_s$ | 0.01 | Weight for the smoothing cost |
| $N_{\text{sub}}$ | 5 | Number of integration sub-steps |
| $u_{\max}$ | 1 | Bound on surge speed |
| $r_{\max}$ | 0.5 | Bound on yaw rate |
| $r_{\text{safe}}$ | 0.6 | Safe obstacle radius [m] |

optimization takes on average 0.1 s to be computed, therefore we limit the control loop rate to 10 Hz. The results of Experiment 3 are presented in Fig. 13, where the UGV is following the reference trajectory using the NMPC.

In Experiment 4, we test the shared control (39) and the NMPC as trajectory tracking controller. As in Experiment 2, we

This article has been accepted for publication in IEEE Transactions on Automation Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TASE.2025.3554368



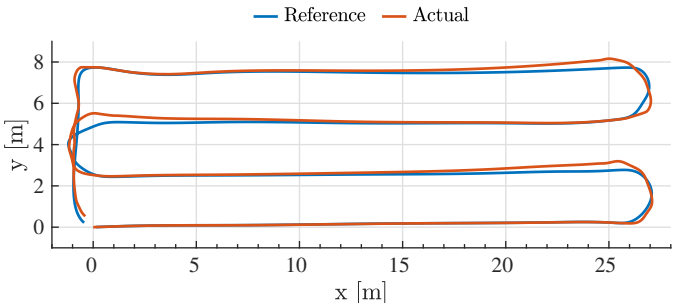



Fig. 13. Reference and actual trajectories for Experiment 3.

placed static and moving obstacles on the reference trajectory and used the joystick to avoid collision with these obstacles. Figure 14 presents the actual and reference trajectories. We consider this a validation of the proposed Lyapunov approach, which achieves a comparable level of performance with a significantly lower computational time: the simulation update time is 0.001s for the Lyapunov controller, whereas it is 0.1s for the NMPC.

The hard constraints on the actuation variables achieve an effect equivalent to the $\lambda$-scaling in the Lyapunov controller. Figure 14 shows that the $r$ variable quickly reaches its maximum upper bound $r_{\max} = 0.5$ rad/s but the actuator constraint is still respected.

*Discussion:* The NMPC allows the UGV to both follow the reference trajectory, and upon human intervention, to avoid obstacles and quickly return to the reference trajectory after the human releases it. However, the computational cost required to implement the NMPC limits the real-time update (i.e. to 2.5 Hz or 10 Hz, with or without obstacle avoidance, respectively). This makes practical implementation more difficult, especially if wireless communication is delayed or disrupted. In contrast, the implementation of the Lyapunov-based shared controller achieves comparable tracking performance with a computation time on the order of milliseconds, making it more suitable for practical implementation. Figure 13 shows that the NMPC takes longer to converge back on the trajectory after the turn at the end of the row, relative to the results in Fig. 8. This is related to the fact that, in the cost function (45), the position tracking is in trade-off with velocity tracking. Therefore, the tracking behavior depends on the tuning of the weights of the various terms in the cost function (Table II). The velocity term is mostly relevant in the obstacle avoidance test, to avoid "getting stuck" when the robot is moving directly toward the obstacle. For sake of simplicity, we kept the same weight settings for the tests with vs. without obstacles, resulting in a slower response at the position level.

## B. Field Experiment Results

In this section, we test the proposed Lyapunov-based controller on experiments with a real platform, in the field. The controller gains obtained from the simulations were manually refined while conducting the experiments, starting from the values obtained in the simulation. The final controller gains are presented in Table III and the joystick parameters in Table IV.

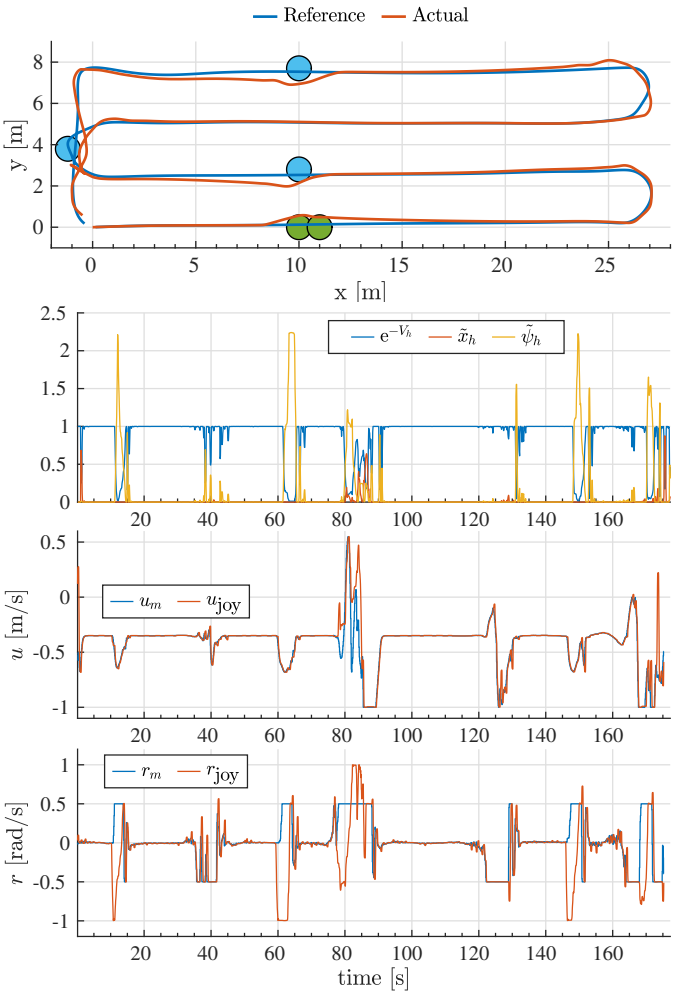



Fig. 14. Experiment 4 results. From top to bottom: a) reference and actual trajectories and obstacles b) Negative exponential of the activation function $V_h$ and $\tilde{x}_h$, $\tilde{\psi}_h$ errors between the mapped automatic control and the mapped joystick position c) the mapped surge speed $u_m$ and the mapped joystick pitch $u_{joy}$ d) the mapped yaw-rate $r_m$ and the mapped joystick roll $r_{joy}$. As for Experiment 2, the operator's attempts to avoid the obstacles are visible when $r_m$ and $r_{joy}$ have opposite signs.

TABLE III
CONTROL PARAMETERS.

| Parameter | Value | Description | Dimension |
|---|---|---|---|
| $k_x$ | 0.6 | Surge speed gain | [1/s] |
| $k_\psi$ | 1.7 | Yaw rate gain | [rad/s] |
| K | 0.08 | CBF boundary constant | |
| C | 0.01 | CBF relaxed constant | |
| $k_{hx}$ | 1 | Human surge speed control gain | [m/s] |
| $k_{h\psi}$ | 0.5 | Human yaw rate control gain | [rad/s] |
| $k_s$ | 5 | joystick spring constant | |
| c | 0.8 | Scaling factor constant | |
| $u_d$ | 0.7 | Desired surge speed | [m/s] |
| $r_{\text{safe},i}$ | 0.6 | Min. safe distance to $i^{\text{th}}$ obstacle | [m] |

Experiments were conducted from September to the end of November (Laimburg, Bolzano, Italy) under different weather conditions, such as long grass and muddy terrain. Some experiments continued through rainy conditions where we had to stop occasionally.

TABLE IV
JOYSTICK PARAMETERS.

| Parameter | Value | Description | Dimension |
|---|---|---|---|
| $k_{jx}$ | 1.5 | surge speed to joystick mapping | [m/s] |
| $k_{j\psi}$ | 2 | yaw rate to joystick mapping | [rad/s] |
| $k_{fx}$ | 1000 | Pitch Stiffness | |
| $k_{f\psi}$ | 1000 | Roll Stiffness | |

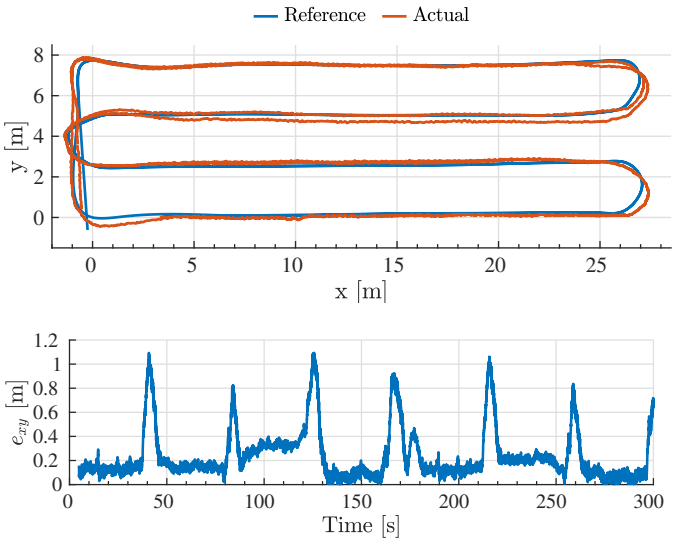



Fig. 15. Experiment 6 results. Lyapunov-based approach: the scaled trajectory tracking (18). Bottom: Norm of the position error. The tracking error increases rapidly during the fast turning actions and decreases again within 10 s when the robot continues the straight portions of the path.

*Experiments 6 and 7 – Trajectory tracking controller:* For the first tests of the Lyapunov-based trajectory tracking (6) (Experiment 5) and the scaled trajectory tracking (18) (Experiment 6), ten complete trials were conducted for each controller. We define the position error between the robot's actual position and the reference trajectory in the world frame as

$$\|e_{xy}\| = \sqrt{(x - x_d)^2 + (y - y_d)^2}, \tag{47}$$

where $(x - x_d)$ and $(y - y_d)$ are the position errors along $X$ and $Y$ direction, respectively. A representative error profile from Experiment 6 is presented at the bottom of Fig. 15, where the error $\|e_{xy}\| \approx 0.2$ except when the robot turns and recovers quickly.

The parameter $c$ in the scaling function (17) in the simulation Exp 1 was set to 0.7. However, in the field, such limitation on the control input magnitude caused the vehicle to turn with a very large radius, resulting in frequent collisions with the next row of trees. To correct this, we increased $c$ to 0.8. As the Lyapunov function $V_1$ never exceeded this threshold of $c = 0.8$, the scaling function (17) was not active during the entire duration of the trial.

*Experiment 7 – Collision avoidance controller:* Here, we tested the collision avoidance controller (32). Three traffic safety cones were used to represent static obstacles, where Obs1 is located at (10, 2.8), Obs2 is located at (-1.2, 3.8), and Obs3 is located at (10, 7.7). The test included one *spoofed* virtual

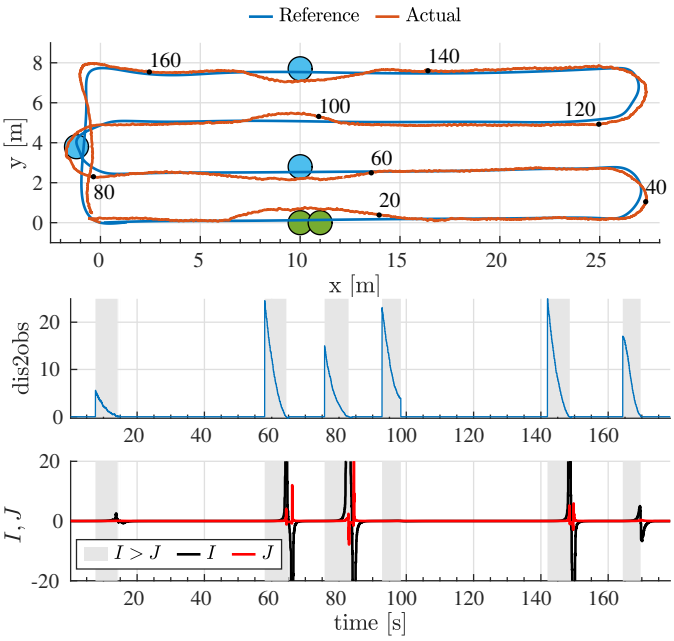



Fig. 16. Experiment 7 results. From top reference and actual trajectory, with obstacle locations. The second plot from the top shows the distance between the robot and the closest obstacle less the radius of the unsafe area; the bottom plot shows (30)–(31).

moving obstacle, which moves at a speed of 0.35 m/s. The safe area around each obstacle is a circle of radius $r_{\text{safe}}$=0.6 m.

The collision avoidance controller (32) is activated when the robot is near the safety area around an obstacle. The distance from the obstacle at which (32) is activated depends depends on the gains K and C in Table III because they affect when when (30) is greater than (31). Thus, we define

$$\text{dis2obs} = \begin{cases} 0, & I \leq J \\ (x - x_{oi})^2 + (y - y_{oi})^2 - r_{\text{safe}}^2, & I > J \end{cases} \tag{48}$$

where $(x, y)$ is the robot position, $(x_{oi}, y_{oi})$ is the position of the closes obstacle. Equation (48) provides a measure of safety wherein dis2obs is less than zero when the robot enters the safety area around an obstacle.

The uppermost plot in Fig. 16 is the collision avoidance trajectory. The robot violates the safety area around Obs-2. This violation is caused by the muddy terrain, in which the robot slips sideways while trying to follow the sharp turn of the reference trajectory. The second plot from the top is the dis2obs metric related to the distance of the robot from the closest obstacle, and the third plot presents functions (30)–(31).

*Experiment 8 – Shared control with collision avoidance:* In this experiment, we test shared trajectory tracking with collision avoidance control (39) using the real robot. The experiment shows that (39) allows the mobile robot to follow a predefined trajectory and avoid collision with static and moving obstacles. In addition, the human operator can manually drive the robot off the reference trajectory. The robot quickly returns to the reference trajectory when the human operator releases it. The force-feedback joystick follows the trajectory tracking controller output (33), where the joystick pitch $u_{\text{joy}}$ follows

the mapped surge speed (34), and the $r_{\text{joy}}$ follows the mapped yaw (35). Thus, the joystick will resist when the human operator tries to move the robot away from the reference trajectory.

We consider the scenario in which we first removed the cones placed at the position of Obs1 and Obs2 and forced the robot to go through the safe area. We put back the cone placed at Obs2 and drove the robot to avoid it by moving it to the right (the automatic collision avoidance controller, by default, avoids the obstacle by turning to the left). In contrast, the collision avoidance for Obs-3 and the moving obstacle is preformed by the collision avoidance controller (32), as the human provided no input.

The robot trajectory is shown in the uppermost plot of Fig. 17. In the second plot from the top of Fig. 17, we present the human intent, related to the exponential of the activation function $e^{-V_h}$ and the respective $\tilde{u}_{\text{joy}}$ and $\tilde{r}_{\text{joy}}$ errors. The third plot from the top represents the surge control and the joystick, where we can see how the surge control increases to $u >$1 m/s when the human operator stops the robot at the time $t =$100 s by manually forcing the $u_{\text{joy}}$ to zero. The fourth plot from the top represents the yaw rate and the $r_{\text{joy}}$. The human operator forces the $r_{\text{joy}}$ to deviate from following the yaw rate such that the robot goes through the safe area around Obs1 and Obs2, and avoids collision with Obs2 to the right rather than to the left, before coming to a stop.

The effect of the scaling function (17) can be seen in Fig.18 when the user stops the robot at time 100 s. In this case, the tracking error $V_1$ increases above the threshold $c = 0.8$, and the scaling is activated (gray area) to reduce the control input magnitude.

Finally, the effect of the reference time re-scaling can be seen by inspecting the evolution of the reference signal along the $X$ (forward) direction shown in Fig. 19. At t = 100 s, when the user asks the robot to slow down, the norm $e_{xy}$ of the tracking error increases, exceeding the chosen threshold of 2 m while the reference changes at a lower rate to limit the increase of the tracking error. We report only the plot in the $X$ direction, where the effect is more apparent because it has a higher variation over time.

*Discussion:* The field experiments were conducted from early September to the end of November 2023, under different weather conditions, including sunny and rainy days. The robot traversed the same terrain in different conditions, with long grass in September and short grass with muddy terrain in October and November [1].

The position error was around 0.2 m except when the robot turned, and it recovered quickly. Given that the RTK-GPS provides the robot localization with 0.1 m accuracy, the actual position error is 0.1 m.

We conducted twenty trajectory-tracking experiments, which each experiment took around 5 minutes, during all these experiments, when the robot was moving on straight lines (the line between the orchard tree lines) around 25 m, the position error was around 0.1 m; thus, we can conclude that the exogenous and model disturbances (1) bound is $C_d \approx 0.1$.

[1]A video of the experiment can be found at `https://youtu.be/OfC-wzxXLpg` and in the supplementary material.

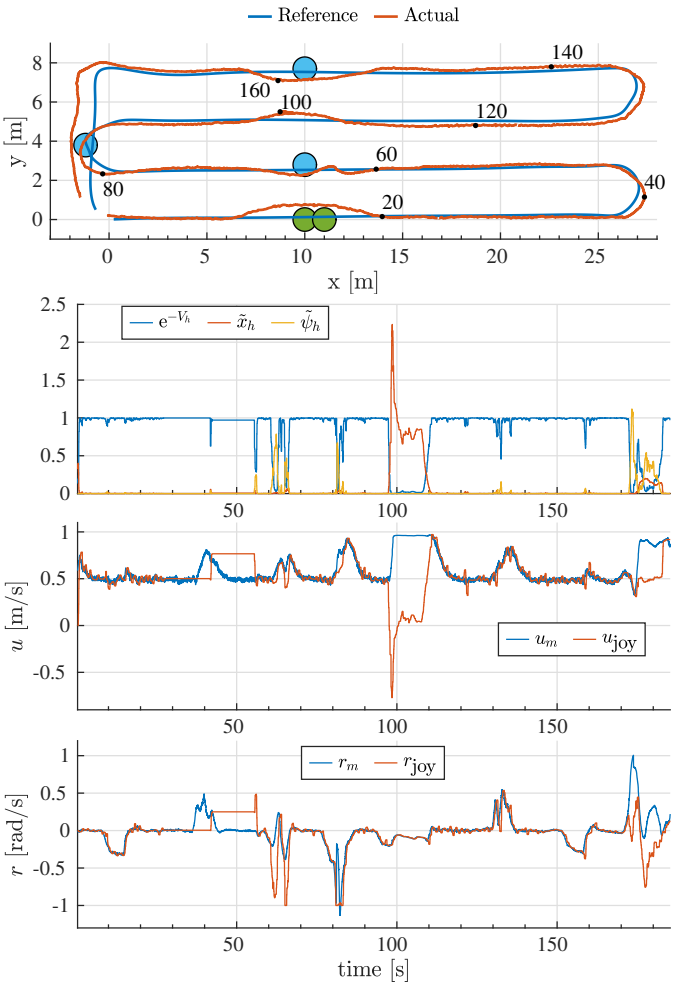



Fig. 17. Experiment 8 results. From top to bottom: a) reference and actual trajectories, with obstacle locations b) Negative exponential of the activation function $V_h$ and $\tilde{x}_h$, $\tilde{\psi}_h$ errors between the mapped automatic control and the mapped joystick position c) the mapped surge speed $u_m$ and the mapped joystick pitch $u_{joy}$. Note that $u_{\text{joy}}$ is forced to zero at time t = 100 s when the user forces the robot to stop fully, d) yaw rate $r$ and joystick position $r_{\text{joy}}$. As in the other experiments, the $r_{\text{joy}}$ has the opposite sign of $r$ when the user manually controls the robot.

The placement of the obstacles provides physical proof of the good performance of the shared trajectory tracking with the collision avoidance controller. Given that the obstacle diameter is 0.2 m, the distance between the tree lines is 2.7 m, and the UGV width is 1 m, the distance between the obstacle edge and the tree line is 1.25 m. The UGV could autonomously or with shared control avoid collision with the obstacles without colliding with tree lines. Also, the time-rescaling allows the UGV to follow the trajectory without colliding with the tree lines after the user commands it to stop for a few seconds.

However, when the terrain was muddy, the UGV experienced slipping during turns. To assess the performance of the collision avoidance controller under these conditions, we placed Obs2 at the final turn. The results showed that the UGV crossed on the safety area around Obs2 by approximately 0.1 m.

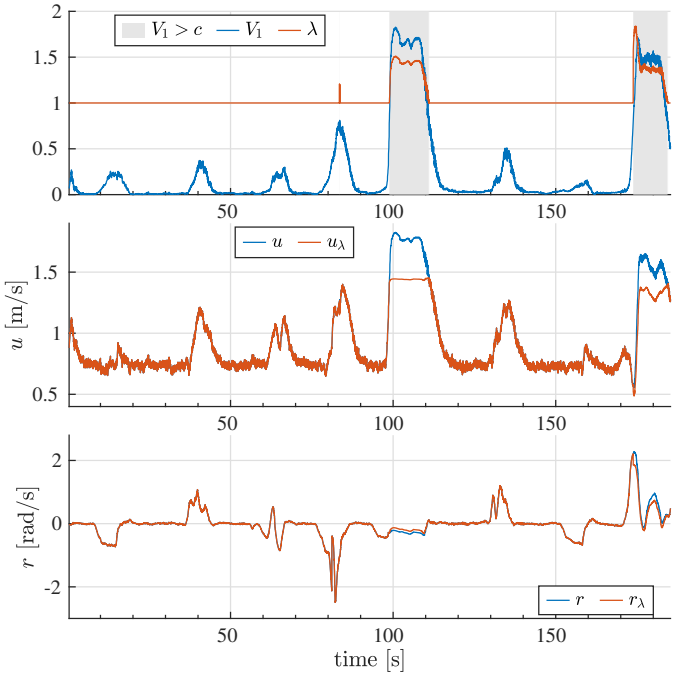



Fig. 18. Experiment 8 analysis on the effects of the $\lambda$ scaling. From top to bottom: a) Lyapunov function $V_1$ and $\lambda$ the scaling function. The gray area indicates when scaling is activated. b) unscaled surge speed $u$ and scaled surge speed $u_\lambda$. Note the magnitude reduction of $u$ when $\lambda$ was active. c) unscaled yaw rate $r$ and scaled yaw rate $r_\lambda$ which are the same since the robot is trying only to move forward.

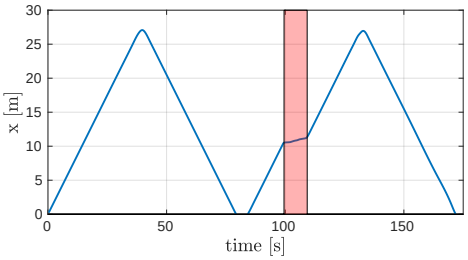



Fig. 19. Experiment 8 analysis on the time scaling. The solid line is the reference trajectory, and the shaded area shows the time interval when the time rescaling is active.

## VII. Conclusions and Future Work

The shared trajectory tracking and collision avoidance controller presented in this paper allows a UGV to maneuver through a semi-structured environment around static and moving obstacles, with smooth variations between the percentage of human operator and automatic control inputs. In our simulations, we demonstrate that our shared control approach allows the UGV to: a) recover from extreme user deviations from the desired trajectory without exceeding the actuator limits, b) avoid static and moving obstacles, and c) transition smoothly between human commands and the automatic control inputs. Additionally, we demonstrate our shared control approach can operate with both optimization- and analytical-based control (i.e. NMPC and Lyapunov, respectively).

We deployed shared control in the field with the "analytical" controllers on a differential-drive UGV using minimal perceptual inputs.

For future research, the challenge of muddy terrain in our experiments prioritizes slippage-aware trajectories, e.g. by considering the sway speed. Thanks to the versatility of our blending solution, future research could also investigate the adoption and stability analysis of optimization-based controllers, such as Linear Quadratic Controler (LQR) [52] and real-time NMPC implementations.

## VIII. Acknowledgment

We thank Dr. Walter Guerra and Ing. Elias Holzknecht from the Laimburg Research Center for logistical support and for providing access to the LIDO field test area.

## References

[1] M. Johns, B. Mok, D. Sirkin, N. Gowda, C. Smith, W. Talamonti, and W. Ju, "Exploring shared control in automated driving," in *2016 11th ACM/IEEE Intl Conf Human-Robot Interaction*. IEEE, 2016, pp. 91–98.

[2] D. A. Abbink, T. Carlson, M. Mulder, J. C. De Winter, F. Aminravan, T. L. Gibo, and E. R. Boer, "A topology of shared control systems—finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018.

[3] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, 2020.

[4] F. Galluppi, C. Urdiales, I. Sanchez-Tato, F. Sandoval, and M. O. Belardinelli, "A study on a shared control navigation system: human/robot collaboration for assisting people in mobility," *Cognitive processing*, vol. 10, no. 2, pp. 215–218, 2009.

[5] J. Mouchoux, S. Carisi, S. Dosen, D. Farina, A. F. Schilling, and M. Markovic, "Artificial perception and semiautonomous control in myoelectric hand prostheses increases performance and decreases effort," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1298–1312, 2021.

[6] J. P. Vasconez, G. A. Kantor, and F. A. Auat Cheein, "Human–robot interaction in agriculture: A survey and current challenges," *Biosystems Engineering*, vol. 179, pp. 35–48, 2019.

[7] L. Benos, A. Bechar, and D. Bochtis, "Safety and ergonomics in human-robot interactive agricultural operations," *Biosystems Engineering*, vol. 200, pp. 55–72, 2020.

[8] J. Han, J. Park, T. Kim, and J. Kim, "Precision navigation and mapping under bridges with an unmanned surface vehicle," *Autonomous Robots*, vol. 38, no. 4, pp. 349–362, 2015.

[9] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current opinion in biotechnology*, vol. 45, pp. 76–84, 2017.

[10] J.-P. Ore and C. Detweiler, "Sensing water properties at precise depths from the air," *J. Field Robotics*, vol. 35, no. 8, pp. 1205–1221, 2018.

[11] R. F. Carpio, C. Potena, J. Maiolini, G. Ulivi, N. B. Rosselló, E. Garone, and A. Gasparri, "A navigation architecture for ackermann vehicles in precision farming," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1103–1110, 2020.

[12] K. Lochan, A. Khan, I. Elsayed, B. Suthar, L. Seneviratne, and I. Hussain, "Advancements in precision spraying of agricultural robots: A comprehensive review," *IEEE Access*, 2024.

[13] R. Abhiram and R. K. Megalingam, "Autonomous fertilizer spraying mobile robot," in *2022 IEEE 19th India Council International Conference (INDICON)*. IEEE, 2022, pp. 1–6.

[14] A. Pal, A. C. Leite, and P. J. From, "A novel end-to-end vision-based architecture for agricultural human–robot collaboration in fruit picking operations," *Robotics and Autonomous Systems*, vol. 172, p. 104567, 2024.

[15] Y. Park, J. Seol, J. Pak, Y. Jo, C. Kim, and H. I. Son, "Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector," *Computers and Electronics in Agriculture*, vol. 212, p. 108116, 2023.

[16] L. Guevara, M. Hanheide, and S. Parsons, "Implementation of a human-aware robot navigation module for cooperative soft-fruit harvesting operations," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2184–2214, 2024.

[17] K. Zhang, K. Lammers, P. Chu, Z. Li, and R. Lu, "An automated apple harvesting robot—from system design to field evaluation," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2384–2400, 2024.

[18] G. Gil, D. E. Casagrande, L. P. Cortés, and R. Verschae, "Why the low adoption of robotics in the farms? challenges for the establishment of commercial agricultural robots," *Smart Agricultural Technology*, vol. 3, p. 100069, 2023.

[19] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi, and Z. Diao, "Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review," *Computers and Electronics in Agriculture*, vol. 205, p. 107584, 2023.

[20] L. Ye, F. Wu, X. Zou, and J. Li, "Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bi-directional rrt approach," *Computers and Electronics in Agriculture*, vol. 215, p. 108453, 2023.

[21] R. R. Shamshiri, E. Navas, V. Dworak, F. A. A. Cheein, and C. Weltzien, "A modular sensing system with canbus communication for assisted navigation of an agricultural mobile robot," *Computers and Electronics in Agriculture*, vol. 223, p. 109112, 2024.

[22] G. Adamides and Y. Edan, "Human–robot collaboration systems in agricultural tasks: A review and roadmap," *Computers and Electronics in Agriculture*, vol. 204, p. 107541, 2023.

[23] M. O. Yerebakan and B. Hu, "Human–robot collaboration in modern agriculture: A review of the current research landscape," *Advanced Intelligent Systems*, p. 2300823, 2024.

[24] Z. Fang, J. Wang, Z. Wang, J. Liang, Y. Liu, and G. Yin, "A human-machine shared control framework considering time-varying driver characteristics," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3826–3838, 2023.

[25] J. Li, C. Yang, L. Ding, B. You, W. Li, X. Zhang, and H. Gao, "Trilateral shared control of a dual-user haptic tele-training system for a hexapod robot with adaptive authority adjustment," *IEEE Transactions on Automation Science and Engineering*, 2024.

[26] J. Luo, D. Huang, Y. Li, and C. Yang, "Trajectory online adaption based on human motion prediction for teleoperation," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3184–3191, 2021.

[27] H. Su, W. Qi, Y. Schmirander, S. E. Ovur, S. Cai, and X. Xiong, "A human activity-aware shared control solution for medical human–robot interaction," *Assembly Automation*, vol. 42, no. 3, pp. 388–394, 2022.

[28] A. Gottardi, S. Tortora, E. Tosello, and E. Menegatti, "Shared control in robot teleoperation with improved potential fields," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 410–422, 2022.

[29] X. Deng, Z. L. Yu, C. Lin, Z. Gu, and Y. Li, "Self-adaptive shared control with brain state evaluation network for human-wheelchair cooperation," *Journal of neural engineering*, vol. 17, no. 4, p. 045005, 2020.

[30] L. Cao, G. Li, Y. Xu, H. Zhang, X. Shu, and D. Zhang, "A brain-actuated robotic arm system using non-invasive hybrid brain–computer interface and shared control strategy," *Journal of Neural Engineering*, vol. 18, no. 4, p. 046045, 2021.

[31] Z. Fang, J. Wang, J. Liang, Y. Yan, D. Pi, H. Zhang, and G. Yin, "Authority allocation strategy for shared steering control considering human-machine mutual trust level," *IEEE Transactions on Intelligent Vehicles*, 2023.

[32] L. Saleh, P. Chevrel, F. Claveau, J.-F. Lafay, and F. Mars, "Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 974–983, 2013.

[33] S. M. Erlien, S. Fujita, and J. C. Gerdes, "Shared steering control using safe envelopes for obstacle avoidance and vehicle stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 441–451, 2015.

[34] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342–354, 2017.

[35] K. D. von Ellenrieder, H. C. Henninger, and R. Belotti, "Homogeneity for shared control in the presence of disturbances," *IFAC-PapersOnLine*, vol. 52, no. 15, pp. 235–240, 2019.

[36] K. D. von Ellenrieder, S. C. Licht, R. Belotti, and H. C. Henninger, "Shared human–robot path following control of an unmanned ground vehicle," *Mechatronics*, vol. 83, p. 102750, 2022.

[37] Y. He, Y. Zhao, and G. Xu, "Double-loop human-robot shared control for trajectory tracking of an underwater robot," in *2023 China Automation Congress (CAC)*. IEEE, 2023, pp. 6276–6281.

[38] C. M. El Bou, F. Beck, K. D. von Ellenrieder, and S. K. Gupta, "Shared control with obstacle avoidance for ugvs," *TechRxiv*, Mar. 2024. [Online]. Available: http://dx.doi.org/10.36227/techrxiv.171166552.24374037/v1

[39] P. Morin, R. M. Murray, and L. Praly, "Nonlinear rescaling of control laws with application to stabilization in the presence of magnitude saturation," *IFAC Proceedings Volumes*, vol. 31, no. 17, pp. 663–668, 1998.

[40] W. Si, N. Wang, and C. Yang, "Design and quantitative assessment of teleoperation-based human–robot collaboration method for robot-assisted sonography," *IEEE Transactions on Automation Science and Engineering*, 2024.

[41] Y. Yang, Y. Yan, C. Hua, J. Li, and K. Pang, "Prescribed performance control for teleoperation system of nonholonomic constrained mobile manipulator without any approximation function," *IEEE Transactions on Automation Science and Engineering*, 2023.

[42] K. D. von Ellenrieder, *Control of Marine Vehicles*. Springer Nature, 2021.

[43] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annual Reviews in Control*, vol. 45, pp. 76–86, 2018.

[44] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[45] S. Zhu and B. Aksun-Guvenc, "Trajectory planning of autonomous vehicles based on parameterized control optimization in dynamic on-road environments," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1055–1067, 2020.

[46] A. Wang, A. Jasour, and B. C. Williams, "Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6041–6048, 2020.

[47] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on mechatronics*, vol. 21, no. 2, pp. 740–753, 2015.

[48] C. Wuthishuwong and A. Traechtler, "Vehicle to infrastructure based safe trajectory planning for autonomous intersection management," in *2013 13th international conference on ITS telecommunications (ITST)*. IEEE, 2013, pp. 175–180.

[49] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Elsevier, 2013.

[50] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*. prentice-Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

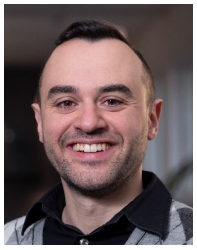
[51] F. Kuhne, W. F. Lages, and J. G. da Silva Jr, "Model predictive control of a mobile robot using linearization," in *Proceedings of mechatronics and robotics*, vol. 4, no. 4. Citeseer, 2004, pp. 525–530.

[52] Z. Su, H. Yao, J. Peng, Z. Liao, Z. Wang, H. Yu, H. Dai, and T. C. Lueth, "Lqr-based control strategy for improving human–robot companionship and natural obstacle avoidance," *Biomimetic Intelligence and Robotics*, vol. 4, no. 4, p. 100185, 2024.


**Cheikh Melainine El Bou** (S'21) received the B.S. degree in electrical engineering from the University of Nouakchott, Nouakchott, Mauritania, in 2011 and the M.S. degrees in control engineering from Sapienza University of Rome, Rome, Italy, in 2019. From 2012 to 2016 he was working as an automation engineer in SNIM(Societe National Industriel et Miniere), Zouerate Mauritania. Since 2020 he has been a PhD student at the Faculty of Science and Technology, at the Libera Università di Bolzano, BZ, Italy. His research topic studies the stability of human-robot shared control for remotely controlled uncrewed vehicle systems.

**Michele Focchi** received both the B.Sc and the M.Sc. degree in control systems engineering from Politecnico di Milano, Italy.in 2007. He received the Ph.D. degree in robotics from Istituto Italiano di Tecnologia (IIT), Genova, Italy, in 2013. From 2014 to 2021 he worked as a researcher at IIT. His research has been concerned with the software development of planning and control strategies for quadruped robots. On 2022 he joined University of Trento, Italy where he works as a lecturer. Currently his research interests are focused on pushing the performances of quadrupeds in traversing unstructured environments, by using optimization-based techniques and devising novel robotic platforms.

**Michael R. Chang** (S'22) joined the FiRST Lab at the Free U. of Bolzano in 2022 as a PhD student in Advanced Systems Engineering. He received his B.A. in Botany from the U. of Wisconsin, Madison, USA, and M.S. in Biology from the U. of New Mexico in 2013. His research involved satellite image processing of agricultural landscapes and biophysical forest simulation modeling. Previously, he worked in education, agriculture, and GIS and GNSS consulting for farms. Current research deploys semantic perception and SLAM in agroforestry contexts.



**Marco Camurri** (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer engineering from the University of Modena and Reggio Emilia, Modena, Italy, in 2009 and 2012, respectively. He received the Ph.D. degree in Bioengineering and Robotics from the University of Genoa and the Istituto Italiano di Tecnologia (IIT), Genoa, Italy, in 2017. After one year as Postoc at IIT, from 2018-2022 he was Posdoc/Senior Research Associate with the Oxford Robotics Institute, University of Oxford, Oxford, U.K. Since 2023, he is RTD (equiv. to Associate Professor) with the Faculty of Engineering, at the Libera Università di Bolzano, BZ, Italy. His research interests include field robotics, mobile robot perception, MPC, autonomous navigation, sensor fusion, and legged robotics.

**Karl D. von Ellenrieder** (M'08–SM'14) received the B.S. degree in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1990 and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 1992 and 1998, respectively. From 1998-2002 he was a Research Fellow at Monash University in Melbourne, VIC Australia and from 2003-2016 he was a Professor of Ocean Engineering at Florida Atlantic University, Dania Beach, FL, USA. Since 2016 he has been a Professor of Automation with the Faculty of Science and Technology, at the Libera Università di Bolzano, BZ, Italy. His research interests include topics related to the control of uncrewed vehicle systems. Prof. von Ellenrieder is a member of the IEEE Control Systems Society and the IEEE Oceanic Engineering Society.