Shared control with obstacle avoidance for UGVs

Cheikh Melainine El Bou,[1] *Student Member, IEEE*, Florian Beck[2],
Karl D. von Ellenrieder[1], *Senior Member, IEEE* and Satyandra K. Gupta[3], *Fellow, IEEE*

***Abstract*—A shared human-robot trajectory tracking controller with collision avoidance is designed for a differentially steered uncrewed ground vehicle. A mixed-initiative interaction is considered, in which a linear blending law combines the human commands with the automatic controller using a passive measurement of the human intent. The automatic control input includes a trajectory tracking controller, designed using a Lyapunov-based approach, and a reactive collision avoidance controller, designed using a relaxed control barrier function. The asymptotic stability of the closed-loop shared human-robot system is analytically proven. An experimental demonstration for the proposed shared control was implemented in a crowded indoor environment with static and moving obstacles. The robot's dimensions are 72 cm x 58 cm x 55 cm (length x width x height). The crowded indoor environment is a 7 m x 15 m room with two doors with a 90 cm width and a 2 m x 20 m corridor. The tracking error was less than 0.03 m, and the robot avoided collision with three static obstacles placed on the reference trajectory and one virtual moving obstacle. The human user can drive the robot away from the reference trajectory when desired, and when he/she relinquishes control, the robot quickly returns to the reference trajectory.**

***Index Terms*—Shared human–robot control, Human-robot interaction, Control barrier functions, Trajectory tracking, Uncrewed ground vehicles**

***Note to Practitioners*—The methods presented in this paper can be easily adapted to a new application that uses a deferentially-steered uncrewed ground vehicle. The approach is designed to work well for an arbitrarily complex, smooth reference trajectory. The practitioner should develop an accurate simulation model and tune the controller parameters to optimize the system's performance. The simulation model should be used to ensure that the system can meet the performance requirements of the given application. The trajectory tracking controller can be tuned to minimize the mean-squared-error between the robot's position and the the desired position, where the positioning error depends on the robot's kinematics and the accuracy of its localization system. The system is expected to work well with obstacles approximated with circles. We recommend using a force feedback joystick to improve the ease of use. To achieve good performance, the system should have a reliable communication channel with a time lag that is considerably shorter than typical human reaction time.**

I. INTRODUCTION

HUMAN-ROBOT shared control can provide a safe and efficient means of deploying uncrewed ground vehicles (UGVs) in unstructured environments by permitting human awareness to compensate for deficiencies in sensing, perception and planning. For example, the pre-planning required to deploy a fully autonomous uncrewed ground vehicle in a time-sensitive search and rescue mission may be prohibitive, whereas the use of a semi-automatic controller would be more feasible. Further, communication problems can occur when a UGV is remotely controlled by a human operator [1], [2], [3]. A shared control approach can mitigate these issues by permitting a human to be part of the closed-loop system when he/she wants, and the automatic controller to take over when needed, e.g. when communication problems occur. Here, we will focus on the shared human-robot trajectory tracking control of a UGV [4], [5].

The shared control of UGVs can benefit a myriad of important applications, as it can be used to permit a human expert to monitor the progress of a primarily automatic mission and only intervene as necessary. For example, the use of semiautomatic shared control can assist in search and rescue operations [6] where the fidelity of sensing systems may be partially compromised by extreme environmental conditions, such as smoke, dust, water, and snow, or when an expert's experience can be used to guide a search to areas where victims are most likely to be found in a complex, cluttered disaster scene, such as after an earthquake, avalanche or building collapse. The use of shared control can also reduce mission times and limit risks to both first responders and victims by providing a means to assess the stability of damaged structures and preplan a rescue mission using visual observations from thermal cameras [7], or measurements of gas/chemical concentrations [8] obtained during a rapid UGV survey. Other applications which can greatly benefit from the shared control of uncrewed ground vehicles by taking advantage of the presence of a human expert include: 1) precision agriculture, where a farmer's experience can be used to guide spraying, inspection, and harvesting operations [9], [10], [11]; 2) infrastructure inspection [12]; 3) firefighting, where UGVs can be used to determine the location of fires and the presence of dangerous gases to assist with intervention planning [13]; 4) mining [14]; and 5) in humanitarian/relief operations, such as the detection and removal of land mines [15].

The presence of moving or stationary obstacles is a significant challenge to the operation of UGVs in unstructured environments. Since it is not possible to guarantee that the human will always have sufficient situational awareness and response time to avoid obstacles when the UGV is remotely located, an automatic collision avoidance method is needed to help ensure the safe operation of the UGV with shared control. A collision avoidance controller based on the use of control barrier functions [16] is developed as a part of the automatic controller.

---

[1] Facoltà di Ingegneria, Libera Università di Bolzano, 39100 BZ Italy (email: cheikhmelainine@ieee.org, karl.vonellenrieder@ieee.org)
[2] Automation and Control Institute (ACIN), Technische Universität Wien, 1040 Vienna, Austria (email: beck@acin.tuwien.ac.at)
[3] Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089 USA (email: skgupta@usc.edu)

This work was sponsored in part by the Autonomous Province of Bolzano, Italy (Recoaro Project #4122), and by a PhD scholarship from the Italian Ministry of Universities and Research DOT19L38RN.

Here, we propose a shared human-robot trajectory tracking with collision avoidance controller, which possesses the following features:

1) The magnitude of the control input is bounded.
2) The controller permits the UGV to maneuver in an environment that includes static and moving obstacles.
3) The ratio of human input to automatic control input depends on human intent.
4) The safe set is forward invariant.

Given these desired features, we seek a control-based collision avoidance method that does not require *a priori* knowledge of the environment. The controller is designed using a control barrier function.

The human-robot shared control system consists of a differentially steered UGV and a human operator. The human interacts with the UGV using a joystick. The human commands and the trajectory tracking control with collision avoidance inputs are mixed using a linear blending law [17]. The trajectory tracking control input is designed using a Lyapunov-based approach [18], while the collision avoidance controller is developed using a control barrier function [16], in particular, the *relaxed control barrier function* (rCBF) introduced in [19], [20]. The asymptotic stability of the closed-loop human-robot shared control system is proven.

The main contributions of this paper are:

1) A model of human-robot shared control that supports a continuously varying ratio of human input to automatic control input depending on the human's intent.
2) A novel trajectory tracking controller based on the model described above permits a remote (non-collocated) UGV to operate within an environment with static and moving obstacles safely.
3) Theoretical foundations to establish the stability of the closed-loop shared control system.
4) Experiment-based case studies to demonstrate the effectiveness of the proposed controller described above.

II. RELATED WORK

The shared human-robot system presented in [17] consists of one human operator and one robot, which is modeled using a second order plant with exogenous disturbances. The automatic part of the shared controller is developed using a back-stepping technique and a disturbance observer. The human part is designed using a storage function to measure human intent. The work in [17] is extended in [21], in which the robot considered is a four wheeled vehicle with front wheel sterring. The automatic part of the shared control is a sliding mode controller based on the dynamic model of the vehicle. In both works the asymptotic stability of the human-robot shared control is proven, while in [21] the developed shared controller is experimentally evaluated using an uncrewed ground vehicle in an outdoor environment.

Collision avoidance methods can be broadly classified in two categories:

1) Planner-based methods, where the desired path is updated to avoid collisions using a path planning algorithm. These methods include RRT [22] and its extensions, RRT* and RRT*smart [23]. Other planner-based approaches include Velocity Obstacles Methods [24], [25], [26]. In VO approaches a cone-shaped obstacle is formed by the set of lines which pass through the center of the robot and are tangent to a buffer region surrounding an obstacle. The VO itself is this cone with its vertex shifted from the center of the robot by the robot's velocity. If the relative velocity between the robot and obstacle lies within the VO, they will collide [27]. By selecting a "best" velocity that lies outside of the VO, e.g. by using an online optimization to minimize the deviation from the desired path, etc., a collision can be avoided. Once computed, the desired velocity can be sent to a trajectory tracking controller as a time-dependent set of waypoints to be tracked until the next planner update is available [28], [29], [30], [31].
2) Controller-based methods, where the control input is updated to avoid a collision. For example in the Artificial Potential Field (APF) Method [32], [33], [34] the collision avoidance control input is determined by considering the robot to be a point in a potential field affected by the attractive forces generated by the goals, and the repulsive forces generated by obstacles. Another approach to control-based collision avoidance involves the use of Control Barrier Functions (CBF) [16], [35], [36], [19]. In this approach a collision-avoidance control input is designed using a control barrier function to keep the robot inside some safe set of states/conditions. The advantage of CBF-based collision avoidance methods is that the controller is designed using techniques very similar to those employed when designing controllers using Control Lyapunov Functions (CLFs). Specifically, CBFs permit the safe operation of the closed-loop system to be explicitly proven by showing the forward invariance of solution trajectories to a safe set.

While several collision avoidance methods have been developed for the trajectory tracking control of UGVs [37], [38], there are few studies involving the shared trajectory tracking control or shared path following of a UGV. One such study is [21], in which a shared control path following system for UGVs is proposed without considering the collision avoidance problem. Another such study is [39] in which a shared controller is proposed, in part, to avoid the deficiencies with APF methods for collision avoidance. In this latter work, the human commands are mixed with a collision avoidance controller, which is designed using the Virtual Force Field (VFF) Method. However, the ratio of human input to automatic control input is fixed.

III. PROBLEM FORMULATION

In a three-dimensional configuration space, the kinematic equations of a mobile robot are given by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{v}, \tag{1}$$

where ψ is the heading angle of the vehicle, $\boldsymbol{R}(\psi)$ is the transformation matrix from the body-fixed system to a North-

East-Down (NED) inertial coordinate system, which is given by

$$\boldsymbol{R}(\psi) := \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SO(3), \quad (2)$$

and

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

are the position and orientation (pose) vector and velocity vector (in body-fixed coordinates), respectively (Fig. 1). The variables appearing in (3) include the position northward x, the position eastward y, the surge speed u, the sway speed v and the yaw rate r.

Throughout this work, the symbol $\mathbb{R}^n$ is the Euclidean space of dimension n, $\mathcal{S}$ the set of Euler angles defined on the interval $[-\pi \quad \pi]$, and $SO(3)$ is the Special Orthogonal Group of order 3, vectors are indicated using bold symbols and scalar value are written with regular math fonts.

The differentially steered mobile robot is modeled as shown in Fig. 1, where the speeds of the two larger motorized wheels can be independently controlled to provide a desired surge speed u and yaw rate r. Thus, we take the control inputs to be u and r.

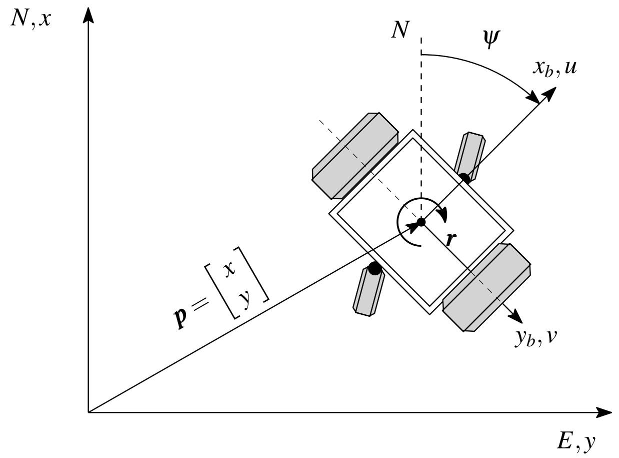

Fig. 1: Top view of a differentially steered mobile robot. The two larger side wheels are driven by motors, while the small fore and aft wheels of the vehicle are unpowered and can freely roll and pivot. The three degree of freedom maneuvering coordinate system definitions are also provided.

Let the state of the system be its pose, i.e. $\boldsymbol{\eta} = \boldsymbol{\eta}$. Then, (1) can be written as

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} -v\sin\psi \\ v\cos\psi \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix}. \quad (4)$$

Suppose the robot's wheels roll without slipping in either the longitudinal or lateral directions, the sway speed v would be identically zero. In this case, the motion of the robot is nonholonomic with first-order kinematic constraints [40]. In the more general case when there might be slipping, v can be related to the time derivatives of $\boldsymbol{\eta}$ using (1), where we note that $\boldsymbol{R}(\psi)^{-1} = \boldsymbol{R}(\psi)^T$. To avoid a singularity in the mapping between the control inputs and the motors actuators, we take the sway speed to be $v = 0$. Thus, (4) has the form of a nonlinear system, which is affine in its control inputs. In addition, we add a collision avoidance controller $\boldsymbol{u}$ to the control input, and define the mapping $\boldsymbol{g} : \mathcal{S} \rightarrow \mathbb{R}^3 \times \mathbb{R}^2$ as

$$\boldsymbol{g} = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix}, \quad (5)$$

then (4) becomes

$$\dot{\boldsymbol{\eta}} = \boldsymbol{g}(\boldsymbol{\eta})(\boldsymbol{\tau} + \boldsymbol{u}). \quad (6)$$

where $\boldsymbol{\tau}$ is the shared trajectory tracking control input.

Here, we explore the shared human-robot control of (6), where the shared trajectory tracking control input $\boldsymbol{\tau}$ is the combination of the signals coming from an automatic trajectory tracking controller $\boldsymbol{\tau}_c$, a human $\boldsymbol{\tau}_h$. The collision avoidance control input $\boldsymbol{u}$ prevents the system from assuming unsafe states.

Let

$$\boldsymbol{\eta}_d(t) = \begin{bmatrix} x_d \\ y_d \\ \psi_d \end{bmatrix} \in \mathbb{R}^2 \times \mathcal{S} \quad (7)$$

be the trajectory to be tracked. The trajectory $\boldsymbol{\eta}_d(t)$ and its first and second derivatives, $\dot{\boldsymbol{\eta}}_d(t)$ and $\ddot{\boldsymbol{\eta}}_d(t)$, are assumed to be smooth and bounded. The trajectory could be designed using a simple point-to-point planning method (e.g. [41]), or more advanced motion planning techniques that take energetic requirements or risk into account [42], [43], [44], [45], [46], [47].

The output of the system is the pose of the robot in the NED frame,

$$\boldsymbol{y}(t) = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \in \mathbb{R}^2 \times \mathcal{S} \quad (8)$$

.

We consider n obstacles represented by the position vector $\boldsymbol{\eta}_o(t) = [\boldsymbol{\eta}_{01}(t), ..., \boldsymbol{\eta}_{on}(t)]^T \in \mathbb{R}^{2n}$ in the NED reference frame, where

$$\boldsymbol{\eta}_{oi}(t) = \begin{bmatrix} x_{oi} \\ y_{oi} \end{bmatrix} \in \mathbb{R}^2, \quad \text{for} \quad i \in \{1, ..., n\}. \quad (9)$$

The sensing and perception of obstacles in unstructured environments is challenging. Recent approaches have implemented combinations of RADAR, LiDAR, sonar and cameras, often together with sensor fusion algorithms to estimate the positions and velocities of obstacles [48], [49], [50], [51], [52], [53].

The states are typically measured using a GPS/GNSS system, in combination with an inertial measurement unit and other sensors, such as LiDAR or RADAR. The sensor measurements are often further refined using state estimation and sensor fusion techniques [54], [55], [56], [57].

Here, clean measurements of the obstacle's pose are assumed to be available for control.

IV. CONTROL DESIGN

Here, we introduce the proposed trajectory-tracking shared controller with collision avoidance for a human-robot system composed of one human operator and one UGV. We start by designing the trajectory tracking controller (the automatic control input) using a Lyaponuv-based approach, then we design the human control input using the passivity of the system, and lastly, we use a blending law to combine the automatic control input and the human control input. The collision avoidance control input, which is considered a part of the automatic control input, will be designed based on a relaxed control barrier function.

A. Kinematic Trajectory Tracking Controller

Here, we restrict the desired trajectory (7) to the set of kinematically feasible trajectories, such that

$$\dot{\boldsymbol{\eta}}_d = \begin{bmatrix} u_d \cos(\psi_d) \\ u_d \sin(\psi_d) \\ r_d \end{bmatrix}, \tag{10}$$

where u_d and ψ_d are the desired surge speed and heading of the vehicle, and $r_d = \dot{\psi}_d$ is the desired yaw rate. For the desired trajectory to be kinematically feasible, the desired sway speed must be $v_d = 0$. This requires the velocity components in the NED frame to satisfy the relation

$$\dot{y}_d \cos\psi_d = \dot{x}_d \sin\psi_d, \tag{11}$$

so that the desired heading angle can be determined from the time-dependent desired pose $\boldsymbol{\eta}_d(t) = [x_d \quad y_d \quad \psi_d]^T$ as

$$\psi_d = \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right). \tag{12}$$

The desired reference trajectory is designed using a spline interpolation technique [58], [59]. Spline interpolation can be used to design a continuous curve between a starting point

$$a = \begin{bmatrix} X_a \\ Y_a \end{bmatrix}, \quad a \in \mathbb{R}^2,$$

and a goal point

$$b = \begin{bmatrix} X_b \\ Y_b \end{bmatrix}, \quad b \in \mathbb{R}^2.$$

Consider the set of way-points $M = [X \quad Y]^T \in \mathbb{R}^{2m}$ where

$$X = \begin{bmatrix} X_1 \\ . \\ . \\ . \\ X_m \end{bmatrix} \in \mathbb{R}^m,$$

$$Y = \begin{bmatrix} Y_1 \\ . \\ . \\ . \\ Y_m \end{bmatrix} \in \mathbb{R}^m,$$

and $[X_1, Y_1]^T = [X_a, Y_a]^T$ is the starting point and $[X_m, Y_m]^T = [X_b, Y_b]^T$ is the endpoint. The time it takes the robot to travel from point $[X_i, Y_i]^T$ to point $[X_{i+1}, Y_{i+1}]^T$, $i \in \{1, ..., m-1\}$ is given by

$$T_{i+1} - T_i = \frac{\sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2}}{u_d}, \quad \text{for } i \in \{2, ..., m\},$$

where u_d is the constant desired surge speed. The x-y components of the desired trajectory x_d and y_d are obtained from a cubic spline interpolation of the sets $\{(X_1, T_1), ..., (X_m, T_m)\}$ and $\{(Y_1, T_1), ..., (Y_m, T_m)\}$, respectively. To obtain the spline interpolation consider a cubic polynomial S_{X_i} between the points (X_i, T_i) and (X_{i+1}, T_{i+1}) for $i \in \{1, ..., m-1\}$. We have an $(m-1)^{\text{th}}$ order cubic polynomial, which requires us to compute $4(m-1)$ unknown parameters. To determine these unknown parameters, we assume the following boundary conditions

1) $S'_{X_i}(T_{i+1}) = S'_{X_{i+1}}(T_{i+2})$,
2) $S''_{X_i}(T_{i+1}) = S''_{X_{i+1}}(T_{i+2})$,

for $i \in \{1, ..., m-2\}$, and

1) $S_{X_i}(T_i) = X_i$,
2) $S_{X_i}(T_{i+1}) = X_{i+1}$,

for $i \in \{1, ..., m-1\}$. There are $4(m-1) - 2$ boundary conditions, so to have a unique solution for the problem we add two more boundary conditions

1) $S''_{X_1}(T_1) = 0$,
2) $S''_{X_{m-1}}(T_m) = 0$,

where S'_{X_i} is the first derivative of S_{X_i}, and S''_{X_i} its second derivative. Similarly, we build S_{Y_i} between the points (Y_i, T_i) and (Y_{i+1}, T_{i+1}) for $i \in \{1, ..., m-1\}$. The resulting reference trajectory for the differential drive robot is

$$\boldsymbol{\eta}_d(t) = \begin{bmatrix} x_d \\ y_d \\ \psi_d \end{bmatrix} = \begin{bmatrix} S_X \\ S_Y \\ \tan^{-1}\left(\dfrac{S'_Y}{S'_X}\right) \end{bmatrix}. \tag{13}$$

The tracking errors for system (6) with respect to the desired trajectory (13) are represented in the body-fixed reference frame of the robot using the transformation matrix (2), therefore

$$\begin{aligned} \tilde{x}_b &= \tilde{x}\cos\psi + \tilde{y}\sin\psi, \\ \tilde{y}_b &= -\tilde{x}\sin\psi + \tilde{y}\cos\psi, \\ \tilde{\psi}_b &= \tilde{\psi}, \end{aligned} \tag{14}$$

where $\tilde{x} = x_d - x$, $\tilde{y} = y_d - y$, and $\tilde{\psi} = \psi_d - \psi$. Define the kinematic trajectory tracking control input $\boldsymbol{\tau}_c$ as

$$\boldsymbol{\tau}_c = \begin{bmatrix} \tau_{cx} \\ \tau_{c\psi} \end{bmatrix}, \tag{15}$$

where

$$\tau_{cx} = u_d \cos(\tilde{\psi}_b) + k_x \tilde{x}_b, \tag{16}$$

$$\tau_{c\psi} = r_d + k_\psi \sin\tilde{\psi}_b + \tilde{y}_b u_d, \tag{17}$$

and $k_x > 0$, $k_\psi > 0$ are constants.

The trajectory tracking control input (15) renders the closed-loop kinematic tracking control system (14) uniformly globally exponentially stable according to [60, Section 5.4].

B. Safety-Critical Control

We consider n obstacles represented by the position vector $\boldsymbol{\eta}_o(t) = [\boldsymbol{\eta}_{01}(t), ..., \boldsymbol{\eta}_{on}(t)]^T \in \mathbb{R}^{2n}$ in the NED reference frame, where

$$\boldsymbol{\eta}_{oi}(t) = \begin{bmatrix} x_{oi} \\ y_{oi} \end{bmatrix} \in \mathbb{R}^2, \quad \text{for} \quad i \in \{1, ..., n\}. \tag{18}$$

The safety-critical collision avoidance controller is designed using a control barrier function (CBF). Since we are considering an environment with dynamic obstacles, we will use the time-varying control barrier function introduced in [20] for system (6). Consider the connected graph space

$$\mathcal{G} = \{(\boldsymbol{\eta}, t) \in \mathbb{R}^2 \times \mathcal{S} \times \mathbb{R} \quad | \quad (x - x_{oi})^2 + (y - y_{oi})^2 > r_i^2\}, \tag{19}$$

for all $i \in \{1, ..., n\}$, where r_i is the minimum safe distance between the robot and the i-th obstacle. Suppose that $(\boldsymbol{\eta}(0), 0) \in \mathcal{G}$, and the time-dependent set

$$\mathcal{X}(t) = \{\boldsymbol{\eta} \in \mathbb{R}^2 \times \mathcal{S} \quad | \quad (\boldsymbol{\eta}, t) \in \mathcal{G}\}. \tag{20}$$

Assumption 1: The speed of each moving obstacle $\dot{\boldsymbol{\eta}}_{oi}$ is bounded.

Definition 1 (Time Varying Control Barrier Function): We consider the graph space (19), the time-dependent set (20) for $t \geq 0$, and the system (6). The continuous and differentiable function $\boldsymbol{B} : \mathcal{G} \to \mathbb{R}$ is said to be a time-varying control barrier function if it satisfies the following conditions

1) $\boldsymbol{B}$ is a positive semi-definitive function, such that for all $t \geq 0$, and $(\boldsymbol{\eta}, t) \in \mathcal{G}$, $B(\boldsymbol{\eta}, t) \geq 0$;
2) for any fixed time $t_0 \geq 0$, $\boldsymbol{B}$ is proper function with respect to the state $\boldsymbol{\eta}$; that is the set $\{\boldsymbol{\eta} \mid B(\boldsymbol{\eta}, t_0) \leq L\}$ is compact for all $L > 0$;
3) there exist constants $C, K > 0$ such that

$$\inf_{\boldsymbol{u} \in \mathbb{R}^2} \dot{B}(\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{u}, t) < KB + C. \tag{21}$$

Consider the safe graph set (19) and n obstacles, where the position of the i^{th} obstacle is located at $\boldsymbol{\eta}_{oi}(t)$, as defined in (18).

Suppose the robot is located at

$$\boldsymbol{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \in \mathbb{R}^2 \times \mathcal{S},$$

and the minimum safe distance between the robot and the i^{th} obstacle is r_i.

To design the collision avoidance control input $\boldsymbol{u} \in \mathbb{R}^2$, the first step is to start with the design of the relaxed time-varying control barrier function (rCBF).

Proposition 1: The function

$$B(\boldsymbol{\eta}, t) = \sum_{i=1}^{n} \left(\frac{1}{(x - x_{oi})^2 + (y - y_{oi})^2 - r_i^2} \right) + \cos^2 \psi, \tag{22}$$

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

To prove this proposition we will shown that $B(\boldsymbol{\eta}, t)$ satisfies the three conditions given in Definition 1.

Proof of Proposition 1: The first condition is $B(\boldsymbol{\eta}, t) \geq 0$ for all $(\boldsymbol{\eta}, t) \in \mathcal{G}$. If $(\boldsymbol{\eta}, t) \in \mathcal{G}$, then $(x - x_{oi})^2 + (y - y_{oi})^2 > r_i^2$ for all $i \in \{1, ..., n\}$, which implies from (22) that $B(\boldsymbol{\eta}, t) \geq 0$ since $\cos^2 \psi \geq 0$.

To prove the second condition, we use a strategy similar to that of [19]. Thus, the function (22) is proper if its domain is compact, which means is closed and bounded. For every fixed time t_0 and $(\boldsymbol{\eta}, t) \in \mathcal{G}$ we can find a positive value $L > 0$ such that the following set

$$\mathcal{X}(t_0) = \{\boldsymbol{\eta} \mid B(\boldsymbol{\eta}, t) \leq L\}, \tag{23}$$

is not empty. The set $\mathcal{X}$ is bounded for all $L > 0$ if $\boldsymbol{\eta} \in \mathcal{G}$, since $\cos^2 \psi \leq 1$, and it's complement in $\mathbb{R}^n$

$$\mathcal{X}^c(t_0) = \{\boldsymbol{\eta} \mid B(\boldsymbol{\eta}, t) > L\}, \tag{24}$$

is an open set, so that $\mathcal{X}(t_0)$ is a closed set. Therefore, $\mathcal{X}(t_0)$ is a compact set.

To prove the third condition, we firstly compute the derivative of the function $B(\boldsymbol{\eta}, t)$, which is

$$\begin{aligned} \dot{B}(\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{u}, t) &= \sum_{i=1}^{n} \left(\frac{\partial B}{\partial x} \dot{x} + \frac{\partial B}{\partial y} \dot{y} + \frac{\partial B}{\partial \psi} \dot{\psi} + \frac{\partial B}{\partial \boldsymbol{\eta}_{oi}} \dot{\boldsymbol{\eta}}_{oi} \right) \\ &= L_g B(\boldsymbol{\tau} + \boldsymbol{u}) + \sum_{i=1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} \end{aligned} \tag{25}$$

with

$$L_g B = \sum_{i=1}^{n} \left(\begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} & \frac{\partial B}{\partial \psi} \end{bmatrix} \boldsymbol{g}(\boldsymbol{\eta}) \right) \in \mathbb{R}^2, \tag{26}$$

and

$$\sum_{i=1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} = \sum_{i=1}^{n} \frac{-2((x - x_{oi})\dot{x}_{oi} + (y - y_{oi})\dot{y}_{oi})}{((x - x_{oi})^2 + (y - y_{oi})^2 - r_i^2)^2}, \tag{27}$$

where $\dot{x}_{oi}$ and $\dot{y}_{oi}$ are the speed of the ith moving obstacle in the x direction and y direction respectively.

Suppose we have m moving obstacles, than (27) became

$$\sum_{i=n-m+1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} = \sum_{i=n-m+1}^{n} \frac{-2((x - x_{oi})\dot{x}_{oi} + (y - y_{oi})\dot{y}_{oi})}{((x - x_{oi})^2 + (y - y_{oi})^2 - r_i^2)^2}. \tag{28}$$

We have that for every fixed time t_0 and $(\boldsymbol{\eta}, t) \in \mathcal{G}$, then $(x - x_{oi})^2 + (y - y_{oi})^2 - r_i^2 > 0$ for all $i \in \{1, ..., m\}$. Given that $\dot{x}_{oi}$ and $\dot{y}_{oi}$ are bounded, this implies that

$$\sum_{i=n-m+1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} < \infty. \tag{29}$$

Thus there exists a $M > 0$ such that

$$\sum_{i=n-m+1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} < M.$$

Which means there exist $K > 0$ and $C > 0$ such that

$$\sum_{i=n-m+1}^{n} \frac{\partial B}{\partial \boldsymbol{\eta}}_{oi} \dot{\boldsymbol{\eta}}_{oi} < KB + C. \tag{30}$$

Case 1: $L_g B = 0$, thus

$$\dot{B}(\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{u}, t) = \sum_{i=n-m+1}^{n} {\frac{\partial B}{\partial \boldsymbol{\eta}}}_{oi} \dot{\boldsymbol{\eta}}_{oi} < KB + C,$$

then the third condition is satisfied. Case 2: $L_g B \neq 0$
Let $I \in \mathbb{R}$ and $J \in \mathbb{R}$ to be defined as

$$I := L_g B \boldsymbol{\tau} + \sum_{i=1}^{n} {\frac{\partial B}{\partial \boldsymbol{\eta}}}_{oi} \dot{\boldsymbol{\eta}}_{oi} \tag{31}$$

and

$$J := KB + C. \tag{32}$$

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

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

If $I > J$ with

$$\boldsymbol{u} = \begin{cases} 0, & I \leq J \\ -\dfrac{(I-J)}{\|L_g B\|^2} L_g B^T - \begin{bmatrix} 0 \\ k_{CBF} \tanh\left(\dfrac{1}{h}\right) \end{bmatrix}, & I > J, \end{cases}$$

then

$$\dot{B}(\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{u}, t) = k_{CBF} \sin(2\psi) \tanh\left(\frac{1}{h}\right),$$

and we have that the hyperbolic tangent $|\tanh(\cdot)| \leq 1$ and $|\sin(\cdot)| \leq 1$, by choosing $C > k_{CBF}$, this implies that $\dot{B}(\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{u}, t) < KB + C$, then the third condition is satisfied.

□

Remark 1: The term $\cos^2(\psi)$ is introduced in the control barrier function (22) to ensure that the yaw rate part of the collision avoidance controller is nonzero. In general, any continuously differentiable, positive function of ψ can be used to achieve this objective.

Based on the proof of Proposition 1 we have the additional proposition.

Proposition 2: Consider the system (6), the safe time-varying set (20), the safe graph set (19) and the time-varying rCBF (22). Given the constants $K, C > 0$, the obstacle avoidance control input

$$\boldsymbol{u} = \begin{cases} 0, & I \leq J \\ -\dfrac{(I-J)}{\|L_g B\|^2} L_g B^T - \begin{bmatrix} 0 \\ k_{CBF} \tanh\left(\dfrac{1}{h}\right) \end{bmatrix}, & I > J, \end{cases} \tag{33}$$

ensures that the graph set (19) is an invariant set.

C. Shared Control

We consider a mixed-initiative interaction between the human and the UGV [61], in which the human can take control of the robot at any time and drive it anywhere.

Denote the human control input as $\boldsymbol{\tau}_h(t)$ and the input from an automatic controller as $\boldsymbol{\tau}_c(t)$. A linear blending law is often used, such as

$$\boldsymbol{\tau}(t) = K_h \boldsymbol{\tau}_h(t) + K_c \boldsymbol{\tau}_c(t), \tag{34}$$

where $\boldsymbol{\tau}(t)$ is the shared control input, $K_h \geq 0$ and $K_c \geq 0$ are arbitration matrices defined such that they form a convex combination, i.e. $K_h + K_c = \mathbf{1}$, with $\mathbf{1}$ being the identity matrix [61].

1) Human control: A two-axis spring-loaded joystick is used to provide human input. By actuating the joystick handle, a user can signal his/her intent to give the UGV a different speed or steering direction from those commanded by the automatic controller. When this occurs, the spring-loaded joystick imparts a force, which is linearly proportional to the zero position of the joystick

$$\begin{aligned} f_x &= k_s \tilde{x}_h, \\ f_\psi &= k_s \tilde{\psi}_h, \end{aligned} \tag{35}$$

where $k_s > 0$ is a spring constant, $\tilde{x}$ is the displacement along the surge direction, and $\tilde{\psi}$ is the displacement along the steering direction. The surge speed control input from the human is then

$$\tau_{hx} = k_{hx} \tilde{x}_h, \tag{36}$$

where $k_{hx} > 0$ is a constant, and the heading control input from the human is

$$\tau_{h\psi} = k_{h\psi} \tilde{\psi}_h, \tag{37}$$

where $k_{h\psi} > 0$ is a constant.

Using (36) and (37), the total control input from the human is given by

$$\boldsymbol{\tau}_h = \begin{bmatrix} \tau_{hx} \\ \tau_{h\psi} \end{bmatrix}. \tag{38}$$

The following assumption is required to prove the stability of the proposed shared control system.

Assumption 2 (Passivity): The Human-joystick system is passive.

A passivity assumption is often used when analyzing the stability of teleoperated systems [62]. In general, a system is passive if every bounded input produces a bounded output. In shared control, what we mean by passivity is that the human does not intentionally apply an input that causes the output of the system (i.e., the tracking error) to become unbounded.

Assumption 2 can be mathematically expressed as

$$\int_0^t \left(f_h \dot{x}_{1h} - f_e \dot{x}_1\right) \mathrm{d}t \geq 0, \tag{39}$$

where f_h is the force applied by a human, f_e is an external force applied by the environment, $\dot{x}_{1h}$ is the speed at which the teleoperated system is being displaced (e.g. the handle of a joystick) and $\dot{x}_1$ is the speed at which the controlled system is moving. Here, since the system is not manipulating its environment, no external forces act on the system (apart from disturbances, which are assumed to require minimal human input to counteract), so that (39) becomes

$$\int_0^t f_h \dot{x}_{1h} \mathrm{d}t \geq 0. \tag{40}$$

Measures of a human's intent to control the speed V_{hx} and direction $V_{h\psi}$ of the system can be defined using Assumption 2. Let $\dot{\tilde{x}}$ and $\dot{\tilde{\psi}}$ be the rates at which the joystick handle is moved away from the tracked position. Then, as shown in [21], using (35) in (40), the measures become

$$V_{hx} := \int_0^t f_x \dot{\tilde{x}}_h \mathrm{d}t = \int_0^t k_s \tilde{x}_h \frac{\mathrm{d}\tilde{x}_h}{\mathrm{d}t} \mathrm{d}t = \frac{1}{2} k_s \tilde{x}_h^2 \geq 0 \quad (41)$$

and

$$V_{h\psi} := \int_0^t f_\psi \dot{\tilde{\psi}}_h \mathrm{d}t = \int_0^t k_s \tilde{\psi}_h \frac{\mathrm{d}\tilde{\psi}_h}{\mathrm{d}t} \mathrm{d}t = \frac{1}{2} k_s \tilde{\psi}_h^2 \geq 0. \quad (42)$$

Note that $V_{hx} > 0$ and $V_{h\psi} > 0$ whenever the joystick handle is displaced from the tracked position in either the front-to-back $\tilde{x} \neq 0$ or side-to-side directions $\tilde{\psi} \neq 0$, and $V_{hx} = 0$ and $V_{h\psi} = 0$ only when $\tilde{x} = 0$ or $\tilde{\psi} = 0$, making it a convenient measure of the human's intent to control the system.

2) Control input blending: As mentioned above, a mixed-initiative approach is used where, V_{hx} and $V_{h\psi}$ are used to measure of the human's intent to control the plant. The two matrices $\boldsymbol{K}_h$ and $\boldsymbol{K}_c$ in (34), are defined as

$$\boldsymbol{K}_h := \begin{bmatrix} (1 - \mathrm{e}^{-V_{hx}}) & 0 \\ 0 & (1 - \mathrm{e}^{-V_{h\psi}}) \end{bmatrix} \quad (43)$$

and

$$\boldsymbol{K}_c := \begin{bmatrix} \mathrm{e}^{-V_{hx}} & 0 \\ 0 & \mathrm{e}^{-V_{h\psi}} \end{bmatrix}, \quad (44)$$

so that as V_{hx} and $V_{h\psi}$ increase, the elements of $\boldsymbol{K}_c$ decrease, and the level of automatic control is reduced in favor of human input. A block diagram of the shared control system is given in Fig. 2.

Remark 2: In preliminary simulations, it was observed that when separate terms involving $\exp(-V_{hx})$ and $\exp(-V_{h\psi})$ were used in the arbitration matrices the system exhibited unstable behavior. The automatic control input is designed to reduce the error between the actual state $\boldsymbol{\eta}(t)$ and the desired state $\boldsymbol{\eta}_d(t)$. If the human activates the heading control input $\boldsymbol{\tau}_{h\psi}$ only, the automatic controller will try to drive the robot back to the desired trajectory using the surge speed control input $\boldsymbol{\tau}_{cx}$, and if the human activates only the surge speed control input $\boldsymbol{\tau}_{hx}$ to move the robot away from the desired trajectory, the automatic controller will try to drive the robot back to the desired trajectory using the heading control input $\boldsymbol{\tau}_{c\psi}$. In both cases, the system exhibited unstable behavior.

For this reason, when the human intends to drive the robot using either the yaw rate $\tau_{\psi h}$ or the surge speed τ_{xh}, in other words, when either V_{hx} or $V_{h\psi}$ are activated, the effect of the automatic part of the controller is reduced.

Assumption 3: Based on Remark 2, we assume that the human operator has full control over the robot if either V_{hx} or $V_{h\psi}$ is activated.

Definition 2: We consider a joystick with two axes, Forward-Backward and Left-Right. From Assumption 3 we can define the human input $\boldsymbol{\tau}_h = [\tau_{xh} \quad \tau_{\psi h}]^T$ as

$$\tau_{xh} = \begin{cases} 0, & |\tilde{x}_h| \leq C_{Joy}, \quad |\tilde{\psi}_h| \leq C_{Joy}, \\ k_{hx} \tilde{x}_h u_d, & |\tilde{x}_h| > C_{Joy}, \quad |\tilde{\psi}_h| \leq C_{Joy}, \\ k_{hx} u_d, & |\tilde{x}_h| \leq C_{Joy}, \quad |\tilde{\psi}_h| > C_{Joy}. \end{cases} \quad (45)$$

$$\tau_{\psi h} = \begin{cases} 0, & |\tilde{\psi}_h| \leq C_{Joy}, \\ k_{h\psi} \tilde{\psi}_h, & |\tilde{\psi}_h| > C_{Joy}. \end{cases} \quad (46)$$

where $|\cdot|$ represents the absolute value of a quantity, k_{hx}, $k_{h\psi}$, and C_{Joy} are positive constants, $\tilde{x}_h$ the is joystick displacement along the Forward-Backward axis, and $\tilde{\psi}_h$ is the joystick displacement along the Left-Right axis. The joystick displacements are scaled by the maximum positive displacement along each axis so that their magnitudes are constrained as $|\tilde{x}_h| \leq 1$ and $|\tilde{\psi}_h| \leq 1$. From experimental observation we found that when we release the joystick the values of $\tilde{x}_h$ and $\tilde{\psi}_h$ are not exactly equal zero. Thus, we use a deadband of magnitude $C_{joy} > 0$ and take $\tau_{xh} = 0$ when $|\tilde{x}_h| < C_{joy}$ and $\tau_{\psi h} = 0$ when $|\tilde{\psi}_h| < C_{joy}$. The Forward-Backward human control input τ_{xh} in (45) is proportional to the product of $\tilde{x}_h$ and the desired speed u_d when activated, and the Left-right human control input $\tau_{\psi h}$ in (46) is proportional to $\tilde{\psi}_h$ when activated.

The arbitration matrices introduced in (34) become

$$\boldsymbol{K}_h = \begin{bmatrix} (1 - \mathrm{e}^{-V_h}) & 0 \\ 0 & (1 - \mathrm{e}^{-V_h}) \end{bmatrix} \quad (47)$$

and

$$\boldsymbol{K}_c = \begin{bmatrix} \mathrm{e}^{-V_h} & 0 \\ 0 & \mathrm{e}^{-V_h} \end{bmatrix}. \quad (48)$$

where $V_h = V_{hx} + V_{h\psi}$.

D. Main Results

The trajectory tracking problem for the human-robot shared control system (6) can be solved by combining the human control input $\boldsymbol{\tau}_h$ with the automatic control input $\boldsymbol{\tau}_c$ using the blending law (34), which results in the shared control input (34). While the collision avoidance problem can be solved using the control input (33).

The magnitude of the human input is proportional to the joystick displacement, which is bounded by design. The boundedness of $(\partial B / \partial \boldsymbol{\eta}_o) \cdot \dot{\boldsymbol{\eta}}_o$ does not contradict any of the three conditions that require the function B to be a relaxed time-varying control barrier function.

D.1. Stability analysis

The shared trajectory tracking controller with collision avoidance (49) is the result of the combination of the Lyapunov-based trajectory tracking controller (15), the human control input (38), and the collision avoidance controller (33), thus the stability analysis of the closed-loop human-robot shared control system (6) can be divided into:

1) the stability of the closed-loop system (6) when we apply only the Lyapunov-based trajectory tracking controller (15);
2) the stability of (6) when we apply only human control input (38);
3) the stability of (6) when we combined (15) and (38); and
4) the forward invariant of the safe graph space (19).

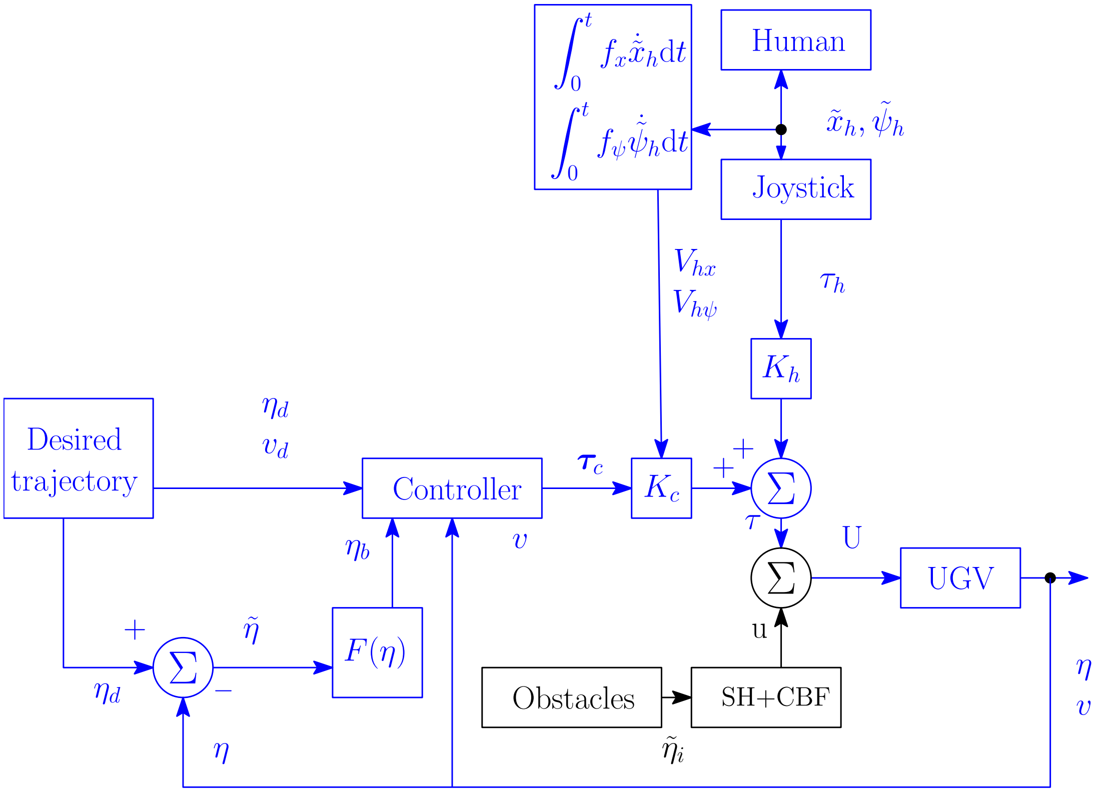

Fig. 2: Block diagram of the shared control system with collision avoidance, the blue part of the figure represents the block diagram of the shared control system without collision avoidance, where $\boldsymbol{\eta}$ is the robot pose in the NED-frame, $\boldsymbol{v}$ the robot velocities in the body-frame, $F(\boldsymbol{\eta})$ is the transformation function from the NED-frame to the body-frame, $\tilde{\boldsymbol{\eta}}$ is the robot pose errors in the NED-frame, $\tilde{\boldsymbol{\eta}}_b$ is the robot pose errors in the body frame, in the block $SH + CBF$ we compute B, $L_g B$, and using the shared control input $\boldsymbol{\tau}$ and the collision avoidance control input $\boldsymbol{u}$ we compute $\boldsymbol{U}$.

The closed loop system (6) is asymptotically stable when we apply only the Lyapunov-based trajectory tracking controller (15) is asymptotically stable according to [60, Section 5.4].

The assumption of passivity (Assumption 2) implies that the human operator will not intentionally drive the UGV far away from the operating region, such that the tracking errors become unbounded. Thus, when only the human control input (38) is applied, the output of the closed-loop system (6) is bounded.

To prove the stability of the closed-loop system (6) when (15) and (38) are combined, we need to prove that the shared controller (34) is bounded. Using the results of [21], if (15) is bounded, and (38) is bounded, then (34) is also bounded. The boundedness of (38) is ensured by assuming that the speed of the moving obstacles is bounded.

The forward invariance of the safe graph space (19) is ensured by the collision avoidance controller (33) in Proposition 2.

Based on this stability analysis we can derive the following theorem.

Theorem 1: Consider the closed-loop human-robot shared control system (6). Under Assumptions 1–3, the safe graph space (19) is an invariant set if the shared trajectory tracking controller with collision avoidance $\boldsymbol{U}$ is defined as

$$\boldsymbol{U} \quad = \quad \boldsymbol{\tau} + \boldsymbol{u}, \tag{49}$$

with $\boldsymbol{\tau}$ in (34) and $\boldsymbol{u}$ in (33).

Proof 1: The control input (49) is the sum of the collision avoidance controller (33), which ensures that the robot will stay in the safe graph set $\mathcal{G}$, and the shared trajectory tracking control input (34), i.e. $\boldsymbol{\tau} = K_c \boldsymbol{\tau}_c + K_h \boldsymbol{\tau}_h$, which is the convex combination of the trajectory tracking control input (15) and the human control input (38). It is proven in [60, Section 5.4] that the trajectory tracking control input (15) asymptotically stabilizes the closed-loop system (6). Owing to Assumption 2, the human does not intentionally drive the robot to instability so that the human input (38) also stabilizes (6). Assumption 3 guarantees that the human can take control of the system when he/she wants. Thus, the closed-loop shared control system (6) is asymptotically stable.

□

V. EXPERIMENTAL SETUP

The mobile platform is a Sally robot produced by DS-Automation of Linz, Austria (Figure 3). The Sally robot has four wheels, where the two right and left wheels are driven by electrical motors, and the fore and aft wheels freely turn and pivot and are used to stabilize the robot. Table I summarizes the main parameters of the Sally robot. A USB camera is mounted on top of the robot and used by the human operator to monitor the robot's surroundings. A Light Detection and

Ranging (LiDAR) sensor is mounted at the front of the robot for self-localization. Two odometry sensors are mounted on the left and right wheels. A picture of the Sally robot with the indication of the USB camera and LiDAR is shown in Figure 3.

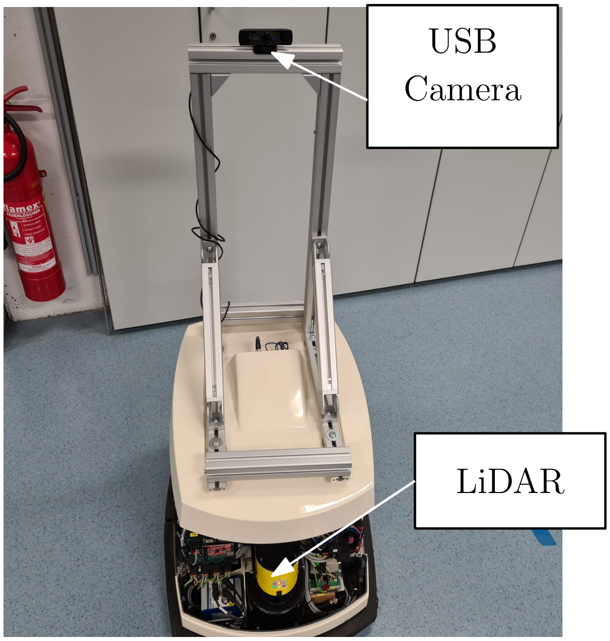

Fig. 3: The DS Automation Sally robot: the human operator uses the USB camera to monitor the environment in front of the robot, while the LiDAR is for self-localization.

TABLE I: Main parameters of the Sally robot.

Parameter	Value [m]
Length	1.0
Width	0.7
Distance between left and right wheels	0.5
Wheel radius	0.075

The ground station consists of two monitors, a graphical processing unit (GPU)-based server, and a CLS-P Side Stick Active Force Joystick produced by Brunner Elektronik AG of Hittnau, Switzerland. The fore-aft movement of the joystick corresponds to a surge control input, and the side-to-side motion of the joystick corresponds to a yaw control input. The ground station and Sally robot communicate via WiFi, which is connected directly to the server unit through an Ethernet cable. The human user supervises the progress of the experiments using the ROS graphical interface RVIZ. The ROS master with the control nodes and the USB camera node run onboard the Sally robot, while the RVIZ node with the joystick node runs on the server unit. A schematic of the experimental setup is shown in Figure 4.

A. Mapping

The experimental demonstration of the proposed shared control system was implemented in a crowded indoor environment, in which the robot's dimensions are 72 cm x 58 cm x 55 cm (length x width x height). The crowded indoor environment is a 7 m x 15 m room with two doors with a 90 cm width and a 2 m x 20 m corridor.

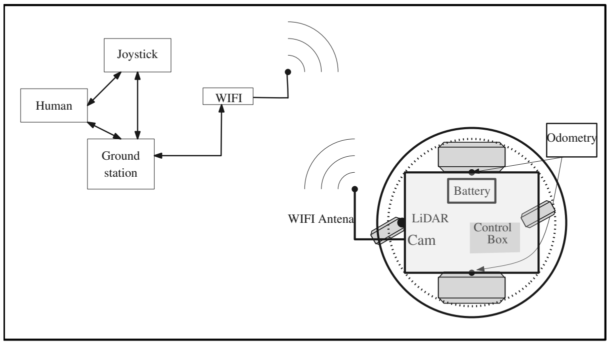

Fig. 4: Block diagram of the ground station and the Sally robot. The arrows indicate the direction of the flow of information between components. The Sally robot has a USB camera, Odometry, and LiDAR sensor.

A map of the laboratory in which the experiment was conducted was created by manually teleoperating the robot within the test area using a joystick. During this process, the output of the LiDAR and odometry sensors is used by the `gmapping` node to construct a map. The ROS graphical interface RVIZ provides visualization of the map in real-time. Figure 5 presents a screenshot of the created map.

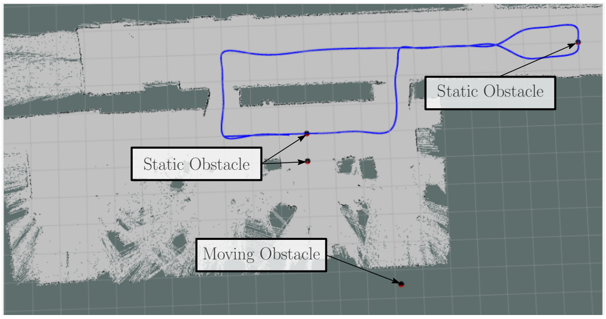

Fig. 5: Screenshot of the map created for the floor in RVIZ. The blue line is the reference trajectory, and the black dots are the obstacles.

B. Control gains tuning

The Lyaponuv-based trajectory tracking controller (15) is designed to reduce the error between the robot pose and the reference trajectory, while the collision avoidance controller (33) is designed to steer the robot away from obstacles. Since the reference trajectory is time-dependent, the error between the robot pose and the reference trajectory can become large during an obstacle avoidance maneuver. When this happens in a crowded environment, the robot may not be able to return to the reference trajectory after the avoidance maneuver if the controller is not properly tuned. For this reason, the trajectory tracking controller (15) and the collision avoidance controller (33) are carefully tuned to ensure the robot continues to follow the reference trajectory when the tracking errors are large.

TABLE II: Control parameters.

Parameter	Value	Description
k_x	0.9	Surge speed gain
k_ψ	1	Yaw rate gain
K	0.5	Collision avoidance constant
C	0.5	Collision avoidance constant
k_{hx}	0.5	Human surge speed control gain
$k_{h\psi}$	0.5	Human yaw rate control gain
k_s	5	Joystick spring constant
c	0.25	Scaling factor constant
u_d	0.5 [m/sec]	Desired surge speed
r_i	0.2 [m]	Min. safe distance to i^{th} obstacle

A similar problem with large tracking errors occurs when the human operator drives the robot far from the reference trajectory. We must ensure that the robot will return to the reference trajectory when the joystick is released.

To provide a starting point for experimentally tuning the controllers, they were first manually tuned using a ROS Gazebo simulation model of the Sally robot. The various controllers were tuned in the following order:

1) the trajectory tracking controller (15);
2) the trajectory tracking with collision avoidance controller (33); and
3) the shared trajectory tracking controller (34);

The controller gains obtained from the simulations were then experimentally refined (manually) using the physical platform. Tuning experiments were conducted in the same order as the tuning simulations. The final controller gains selected are presented in Table II.

VI. EXPERIMENTAL RESULTS

Consider a scenario in which the Sally robot is used as a service robot [63], [64]. For instance, in a search-rescue scenario, where the robot is used to survey the floor of a building. A user can give the robot a predefined task, which is performed while following the reference trajectory and remotely monitoring its progress using the camera mounted on top of the robot. The user can also manipulate the joystick to move the robot away from the reference trajectory, as needed, for example, to better see a dark region of the survey area more closely, or more slowly, examining the dark region. When the user releases the joystick the robot must return to the reference trajectory to resume its assigned task.

A spline interpolation on a set of points $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^m$, where m is the number of points, is used to generate the reference trajectory, as explained in Section III. The reference trajectory used in the experiments is shown in Figure 5. The desired surge speed is $u_d = 0.5$ [m/sec], and the desired yaw rate is the derivative of ψ_d where

$$\psi_d = \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right), \tag{50}$$

in which $\dot{y}_d$ and $\dot{x}_d$ are the first derivative of resulting trajectory from the sets $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^m$, respectively.

We consider four obstacles, including three static obstacles represented by cones with 0.2 meters diameter located at $\eta_{o_1} = [0,1]$, $\eta_{o_2} = [0,0]$, $\eta_{o_3} = [10,4]$, and one virtual moving obstacle, which tracks an L-shaped trajectory. The obstacle trajectory is created using spline interpolation, as described in Section III, on the set of points $X_{obs} \in \mathbb{R}^m_o$ and $Y_{obs} \in \mathbb{R}^m_o$ at a constant speed of $u_o = 0.2$ [m/sec], where m_o is the number of points. The safe distance is taken to be $r_i = 0.2$ [m] for $i \in \{1,2,3,4\}$. Thus, the distance between the robot and any obstacle should be greater than r_i. This is checked by monitoring the denominator of the control barrier function (22) during the experiments. In addition, the error between the robot pose $\boldsymbol{\eta}$ and the desired trajectory $\boldsymbol{\eta}_d$ is used as a metric for measuring the effectiveness of the trajectory tracking controller. The pose error is defined as

$$\begin{aligned} E_P &:= \|\boldsymbol{\eta} - \boldsymbol{\eta}_d\|, \\ &= \sqrt{(x-x_d)^2 + (y-y_d)^2 + (\psi-\psi_d)^2}, \end{aligned} \tag{51}$$

where $(x - x_d)$ and $(y - y_d)$ are the position error in meters, while $(\psi - \psi_d)$ is the angular error in radians. The testing of the shared trajectory tracking control with collision avoidance (49) is performed in two stages: 1) in the first stage, the trajectory tracking with collision avoidance, and 2) in the second stage, the shared trajectory tracking controller.

A. Trajectory tracking with collision avoidance

The first experimental testing is for the trajectory tracking controller (15). The trajectory of the Sally robot and the desired trajectory are presented in Figure 6. The Sally robot is controlled by the trajectory tracking controller (15). The Sally robot can follow the desired trajectory with the pose errors shown in Figure 7, where the jumps on the error are caused by the sharp turns in the trajectory in which the difference between the robot heading angle and the trajectory angle become greater than $\pi/2$. However, the trajectory tracking controller recovers from this error in less than 1 second.

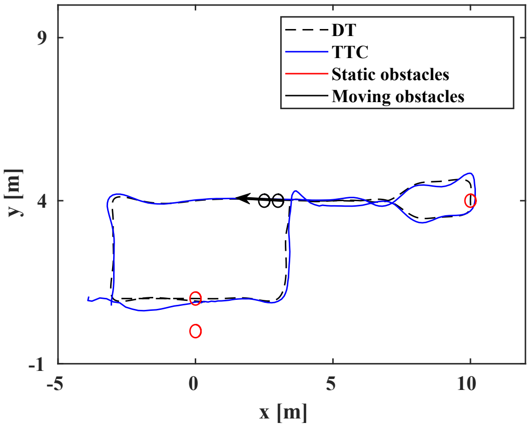

Fig. 6: DT is the desired trajectory, TTC is the actual trajectory, the red circles represent the static obstacles, the black circle represents the moving obstacle, and the arrow indicates the direction of the moving obstacle.

Now we consider a combined controller in which the trajectory tracking controller (15) drives the Sally robot to follow the reference trajectory, while the collision avoidance

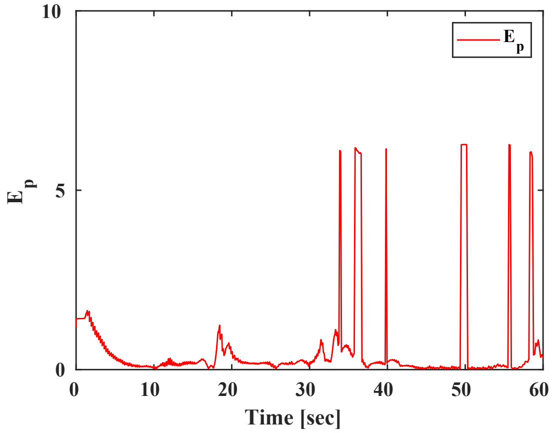

Fig. 7: The pose error (E_p) defined in (36) is the error between the desired trajectory and the Sally robot's actual trajectory, the jumps in the error value are caused by 90° turn, in which the angular error becomes greater than $\pi/2$.

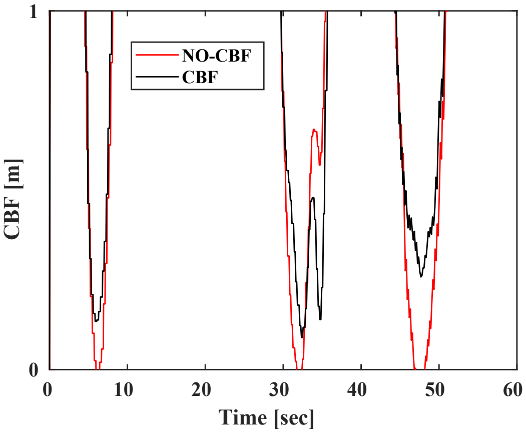

Fig. 9: NoCBF the variation of the control barrier function denominator when there is no collision avoidance controller, and CBF when we consider the collision avoidance controller.

controller (33) prevents it from colliding with the four obstacles (three static obstacles Obs_1, Obs_2, Obs_3 and one moving obstacle Obs_4). The Sally robot trajectory and the desired trajectory are plotted in Figure 8, where we can see that the trajectory tracking controller with collision avoidance allows the robot to follow the reference trajectory and avoid collision with the obstacles. In Figure 9, we show the denominator of the control barrier function (22) in both, which indicates how the collision avoidance controller can keep the robot in the safe zone by keeping the denominator greater than zero.

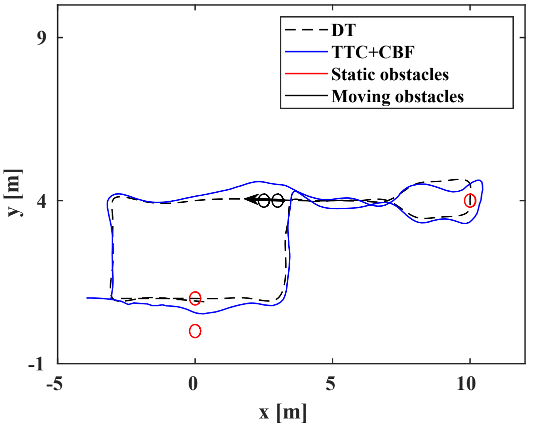

Fig. 8: DT the desired trajectory, TTC+CBF the Sally robot's actual trajectory, the red circles represent the static obstacles, the black circle represents the moving obstacle, and the arrow indicates the direction of the moving obstacle.

B. Shared trajectory tracking controller

Here, we explore shared control in which the trajectory tracking controller (34) is combined with input from a human operator. The human uses the joystick to make the robot avoid collision with the first two static obstacles obs_1, obs_2 and the moving obstacle obs_4. The trajectory of the robot and the desired trajectory are shown in Figure 10. Figure 11 shows the activation function, which is activated as the robot approaches the first two static and moving obstacles.

Test Outcomes: The two tests demonstrate that the proposed shared trajectory tracking controller with collision avoidance can allow a UGV to perform tasks in a confined environment such that:

1) The pose error E_p is less than 0.03, except in $\pi/2$ turn in which the angular error became greater than $\pi/2$, and the trajectory tracking controller reduces the pose error to $E_p = 0.02$ within time less than 2 seconds.
2) The human user drives the robot between the time 2.5 seconds and 6.3 seconds to avoid collision with the static obstacles located at $(0, 0)$ and $(0, 1)$, and between the times 43.2 seconds and 46.6 seconds to avoid colliding with the moving obstacle. During both these time intervals, the switching between the human authority and trajectory tracking controller does not cause any bad behavior of the robot. The trajectory tracking controller brings the robot back to the reference trajectory each time the human user releases the joystick.
3) The collision avoidance controller drives the robot to the right of an obstacle to avoid a collision. The tuning for the collision avoidance controller allows the robot to avoid collision with obstacles without going very far from the reference trajectory; for instance, the robot avoids collision with the two obstacles $(0, 0)$ and $(0, 1)$ by going between them.

VII. CONCLUDING REMARKS

Here, we have shown that a closed-loop shared control system consisting of a human operator and an uncrewed ground vehicle can safely operate in an environment with static and moving obstacles. A Lyapunov-based trajectory tracking controller combined with a collision avoidance controller based on a relaxed control barrier function is the automatic

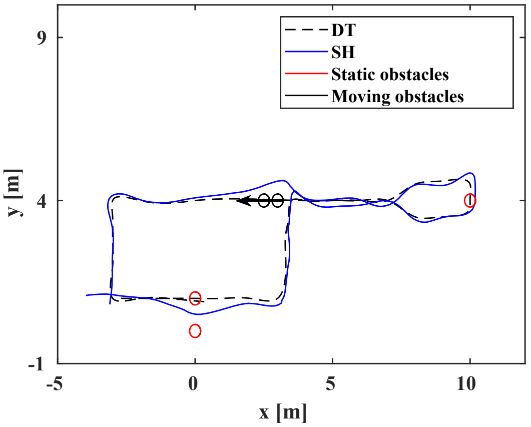

Fig. 10: DT is the desired trajectory, SH is the shared trajectory, the red circles are the static obstacles, the black circle is the moving obstacle, and the arrow indicates the moving direction of the obstacle.

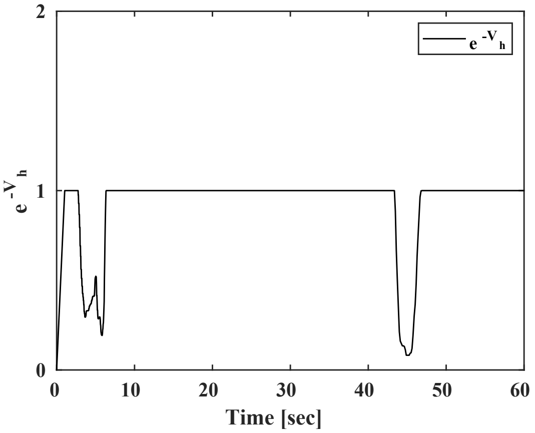

Fig. 11: The exponential of the minus of the activation function.

input to a differentially steered UGV. A linear blending law permits the system to switch between automatic control and human input. Of special significance is the fact that the human and the machine are not collocated. It is hoped that the results of the work can be used as a comprehensive example of how shared control between a human and non-collocated machine can be implemented for many important applications, such as agricultural inspection, monitoring, spraying, and harvesting, as well as for other applications that occur in remote or difficult to reach environments, such as search and rescue. We note that when the human and robot are not collocated, additional challenges are introduced, such as providing the human with adequate situational awareness and ensuring that communication delays do not cause instability. We plan to explore these issues further in future work.

ACKNOWLEDGMENTS

The experiments reported in this paper were conducted while the first author was performing his Ph.D. study period abroad as a visiting researcher at the Technical University of Vienna (Wien), Austria. The authors would like to thank Prof. Andreas Kugi and the Automation and Control Institute members at TU Vienna for their kind support and technical input.

REFERENCES

[1] R. R. Murphy and J. L. Burke, "Up from the rubble: Lessons learned about hri from search and rescue," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3. SAGE Publications Sage CA: Los Angeles, CA, 2005, pp. 437–441.

[2] M. Baker, R. Casey, B. Keyes, and H. A. Yanco, "Improved interfaces for human-robot interaction in urban search and rescue," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 3. IEEE, 2004, pp. 2960–2965.

[3] A. Norton, W. Ober, L. Baraniecki, E. McCann, J. Scholtz, D. Shane, A. Skinner, R. Watson, and H. Yanco, "Analysis of human–robot interaction at the darpa robotics challenge finals," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 483–513, 2017.

[4] L. Li, J. Li, and S. Zhang, "State-of-the-art trajectory tracking of autonomous vehicles," *Mechanical Sciences*, vol. 12, no. 1, pp. 419–432, 2021.

[5] D. Calzolari, B. Schürmann, and M. Althoff, "Comparison of trajectory tracking controllers for autonomous vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–8.

[6] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *Ieee Access*, vol. 8, pp. 191 617–191 643, 2020.

[7] J. Quenzel, M. Splietker, D. Pavlichenko, D. Schleich, C. Lenz, M. Schwarz, M. Schreiber, M. Beul, and S. Behnke, "Autonomous fire fighting with a UAV-UGV Team at MBZIRC 2020," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 934–941.

[8] T. Lewis and K. Bhaganagar, "A comprehensive review of plume source detection using unmanned vehicles for environmental sensing," *Science of The Total Environment*, vol. 762, p. 144029, 2021.

[9] G. Quaglia, C. Visconte, L. S. Scimmi, M. Melchiorre, P. Cavallone, and S. Pastorelli, "Design of the positioning mechanism of an unmanned ground vehicle for precision agriculture," in *Advances in Mechanism and Machine Science*, T. Uhl, Ed. Cham: Springer International Publishing, 2019, pp. 3531–3540.

[10] M. Mammarella, L. Comba, A. Biglia, F. Dabbene, and P. Gay, "Cooperation of unmanned systems for agricultural applications: A case study in a vineyard," *Biosystems Engineering*, vol. 223, pp. 81–102, 2022.

[11] F. N. Ortataş, H. Ulutaş, M. E. Şahİn, and F. Çİftcİ, "Autonomous mapping and spraying in precision agriculture using unmanned ground vehicles," in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2023, pp. 1–5.

[12] H. Miura, A. Watanabe, M. Okugawa, T. Miura, and T. Koganeya, "Plant inspection by using a ground vehicle and an aerial robot: lessons learned from plant disaster prevention challenge in world robot summit 2018," *Advanced Robotics*, vol. 34, no. 2, pp. 104–118, 2020.

[13] N. F. Talavera, J. J. Roldán-Gómez, F. Martín, and M. C. Rodriguez-Sanchez, "An autonomous ground robot to support firefighters' interventions in indoor emergencies," *Journal of Field Robotics*, vol. 40, no. 3, pp. 614–625, 2023.

[14] T. Zou, J. Angeles, and F. Hassani, "Dynamic modeling and trajectory tracking control of unmanned tracked vehicles," *Robotics and Autonomous Systems*, vol. 110, pp. 102–111, 2018.

[15] G. M. Clayton, "Mechatronics for humanitarian explosive ordnance disposal in Cambodia," *Mechanical Engineering*, vol. 140, no. 09, pp. S4–S10, 2018.

[16] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[17] 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.

[18] A. Behal, W. Dixon, D. M. Dawson, and B. Xian, *Lyapunov-based control of robotic systems*. CRC Press, 2009.

[19] M. Igarashi and H. Nakamura, "Collision avoidance assist control for two-wheel vehicle robots by control barrier function," in *2018 International Automatic Control Conference (CACS)*. IEEE, 2018, pp. 1–6.

[20] M. Igarashi, I. Tezuka, and H. Nakamura, "Time-varying control barrier function and its application to environment-adaptive human assist control," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 735–740, 2019.

[21] 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.

[22] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[23] I. Noreen, A. Khan, and Z. Habib, "A comparison of RRT, RRT* and RRT*-smart path planning algorithms," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 10, p. 20, 2016.

[24] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 560–565.

[25] ——, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[26] D. Wilkie, J. Van Den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5573–5578.

[27] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.

[28] P. Svec, B. C. Shah, I. R. Bertaska, J. Alvarez, A. J. Sinisterra, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, "Dynamics-aware target following for an autonomous surface vehicle operating under colregs in civilian traffic," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3871–3878.

[29] B. C. Shah, P. Švec, I. R. Bertaska, W. Klinger, A. J. Sinisterra, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, "Trajectory planning with adaptive control primitives for autonomous surface vehicles operating in congested civilian traffic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2312–2318.

[30] B. C. Shah, P. Švec, I. R. Bertaska, A. J. Sinisterra, W. Klinger, K. von Ellenrieder, M. Dhanak, and S. K. Gupta, "Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic," *Autonomous Robots*, pp. 1–25, 2015.

[31] I. R. Bertaska, B. Shah, K. von Ellenrieder, P. Švec, W. Klinger, A. J. Sinisterra, M. Dhanak, and S. K. Gupta, "Experimental evaluation of automatically-generated behaviors for usv operations," *Ocean Engineering*, vol. 106, pp. 496–514, 2015. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2015.07.002

[32] S. Lee and J. Park, "Cellular robotic collision-free path planning," in *Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments*. IEEE, 1991, pp. 539–544.

[33] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings 2003 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2003)*, vol. 2. IEEE, 2003, pp. 735–740.

[34] D. H. Lee, S. S. Lee, C. K. Ahn, P. Shi, and C.-C. Lim, "Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 10, pp. 9998–10 006, 2020.

[35] M. Rauscher, M. Kimmel, and S. Hirche, "Constrained robot control using control barrier functions," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 279–285.

[36] S. Fukuda, Y. Satoh, and O. Sakata, "Trajectory-tracking control considering obstacle avoidance by using control barrier function," in *2020 International Automatic Control Conference (CACS)*. IEEE, 2020, pp. 1–6.

[37] H.-y. Zhang, W.-m. Lin, and A.-x. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.

[38] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[39] P. Pappas, M. Chiou, G.-T. Epsimos, G. Nikolaou, and R. Stolkin, "VFH+ based shared control for remotely operated mobile robots," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 366–373.

[40] A. De Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and dynamics of multi-body systems*. Springer, 1995, pp. 277–342.

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

[42] 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.

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 4, pp. 1599–1616, 2012.

[49] T. Dang, S. Kammel, C. Duchow, B. Hummel, and C. Stiller, "Path planning for autonomous driving based on stereoscopic and monoscopic vision cues," in *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 2006, pp. 191–196.

[50] P. Marin-Plaza, A. Hussein, D. Martin, and A. d. l. Escalera, "Global and local path planning study in a ROS-based research platform for autonomous vehicles," *Journal of Advanced Transportation*, vol. 2018, 2018.

[51] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, "Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm," *Energies*, vol. 12, no. 12, p. 2342, 2019.

[52] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[53] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 452–458.

[54] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[55] M. Rajasekhar and A. K. Jaswal, "Autonomous vehicles: The future of automobiles," in *2015 IEEE International Transportation Electrification Conference (ITEC)*. IEEE, 2015, pp. 1–6.

[56] J. A. Guerrero, M. Jaud, R. Lenain, R. Rouveure, and P. Faure, "Towards LIDAR-RADAR based terrain mapping," in *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, 2015, pp. 1–6.

[57] M. S. Samrat, M. F. Ali, M. A. Islam, M. Hasan, and M. A. Hasan, "Hybrid sensor based path-planning for autonomous vehicle," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*. IEEE, 2021, pp. 1–4.

[58] I. J. Schoenberg, *Cardinal spline interpolation*. SIAM, 1973.

[59] C. Habermann and F. Kindermann, "Multidimensional spline interpolation: Theory and applications," *Computational Economics*, vol. 30, pp. 153–169, 2007.

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

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

[62] P. F. Hokayem and M. W. Spong, “Bilateral teleoperation: An historical survey,” *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[63] D. Belanche, L. V. Casaló, C. Flavián, and J. Schepers, “Service robot implementation: a theoretical framework and research agenda,” *The Service Industries Journal*, vol. 40, no. 3-4, pp. 203–225, 2020.

[64] H. Gao, X. Zhang, J. Wen, J. Yuan, and Y. Fang, “Autonomous indoor exploration via polygon map construction and graph-based slam using directional endpoint features,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1531–1542, 2018.

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 Universitá di Bolzano, BZ, Italy. His research topic is studying the stability of human-robot shared control for remotely controlled uncrewed vehicle systems.

Florian Beck received a master’s degree and bachelor degree in Computer Engineering from TU Wien, Vienna, Austria, in 2018 and 2015, respectively. He is currently employed as a research assistant at the Automation and Control Institute (ACIN) at TU Wien, Vienna, Austria. His research interests include reactive motion planning and control for safe human-robot collaboration.

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 Universitá 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.

Dr. Satyandra K. Gupta holds the Smith International Professorship in the Viterbi School of Engineering at the University of Southern California. Dr. Gupta’s research interests are physics-informed artificial intelligence, computational foundations for decision making, and human-centered automation. He works on applications related to Computer-Aided Design, Manufacturing Automation, and Robotics. He has published more than four hundred technical articles in journals, conference proceedings, and edited books. He is a fellow of the American Society of Mechanical Engineers (ASME), Institute of Electrical and Electronics Engineers (IEEE), Solid Modeling Association (SMA), and Society of Manufacturing Engineers (SME).