Robotica (2018) volume 36, pp. 463–483. © Cambridge University Press 2017
doi:10.1017/S0263574717000510

Obstacle avoidance control of a human-in-the-loop mobile robot system using harmonic potential fields

C. Ton †*, Z. Kan‡ and S. S. Mehta§

†*Research and Engineering Education Facility, University of Florida, Shalimar, FL, USA*
‡*Department of Mechanical and Industrial Engineering, The University of Iowa, Iowa City, IA, USA. E-mail: kanzhen0322@gmail.com*
§*Department of Industrial and Systems Engineering, University of Florida, Shalimar, FL, USA. E-mail: siddhart@ufl.edu*

(Accepted October 14, 2017. First published online: November 16, 2017)

SUMMARY
This paper considers applications where a human agent is navigating a semi-autonomous mobile robot in an environment with obstacles. The human input to the robot can be based on a desired navigation objective, which may not be known to the robot. Additionally, the semi-autonomous robot can be programmed to ensure obstacle avoidance as it navigates the environment. A shared control architecture can be used to appropriately fuse the human and the autonomy inputs to obtain a net control input that drives the robot. In this paper, an adaptive, near-continuous control allocation function is included in the shared controller, which continuously varies the control effort exerted by the human and the autonomy based on the position of the robot relative to obstacles. The developed control allocation function facilitates the human to freely navigate the robot when away from obstacles, and it causes the autonomy control input to progressively dominate as the robot approaches obstacles. A harmonic potential field-based non-linear sliding mode controller is developed to obtain the autonomy control input for obstacle avoidance. In addition, a robust feed-forward term is included in the autonomy control input to maintain stability in the presence of adverse human inputs, which can be critical in applications such as to prevent collision or roll-over of smart wheelchairs due to erroneous human inputs. Lyapunov-based stability analysis is presented to guarantee finite-time stability of the developed shared controller, i.e., the autonomy guarantees obstacle avoidance as the human navigates the robot. Experimental results are provided to validate the performance of the developed shared controller.

KEYWORDS: Shared control; Robust control; Obstacle avoidance; Harmonic potential field; Adjustable autonomy.

1. Introduction

The integration of human intelligence, modern information technology, and robotics in human–robot interaction has been an active research area for decades. One of the goals is to fuse the expert knowledge of humans with the computational advantage of robots in well-learned situations to improve the performance of co-robotic systems. From the control perspective, the challenge is in fusing human inputs with autonomous commands to affect the behavior of the robot while guaranteeing safety and stability of the system. The early work by Sheridan (cf. ref. [1,2]) introduced supervisory control, where a human supervisor would direct and monitor activities of an autonomous process. The separation between human intervention and autonomous execution in supervisory control is overcome with shared control that enables both the autonomy and the human to participate in process- or task-level execution. Shared control uses the strengths of the robot and the human to overcome each other's weaknesses by combining a pre-determined or time-varying level of autonomy

* Corresponding author. E-mail: chau.t.ton@gmail.com

with the human input. Applications of shared control include medical and biomedical systems,[3–5] teleoperation,[6–9] assistive robotics,[10–16] and brain-machine interface.[17]

To this end, shared control can be divided into two categories: task-level shared control and servo- or execution-level shared control. Similar to supervisory control, in task-level shared control, human issues broad commands to a robot to perform general tasks. The robot carries out commanded tasks until a trigger is raised (e.g., encountered an obstacle), when the control switches to the robot. Discussions of the task-level shared control can be found in refs. [7, 18–22].

In execution-level shared control, the commanded control input to a robot is obtained by appropriately fusing the human and the autonomy control inputs. The robot continually integrates its own assessment of the environment with the influence human exerts on the system. The amount of human influence on the robot depends on the designed control allocation weights, which can be fixed or adaptive. Fixed control allocation is used when prior knowledge of the quality of human and robot control inputs is known. For example, in dual-user haptic training system for medical surgery, the control allocation can be chosen according to the expertise of the trainee and the trainer.[4, 23–27] In contrast to fixed control allocation, adaptive control allocation dynamically allocates control authority to the human and the autonomy. The control allocation can be based on human performance,[14, 15, 28] duration of interaction,[29] safety of the system,[30] agreement between the human and the robot,[16] or the state of the robot in the environment.

Execution-level shared control has been successfully demonstrated in various applications. However, most of the existing results lack rigorous stability and robustness guarantees of the overall human–robot systems. Disparate and occasionally conflicting control objectives between the human and the robot pose great challenges in the design and analysis of shared control systems. For example, smart wheelchairs must be designed to counteract dangerous maneuvers and prevent collisions and falls that may result from the lapse in judgment or conflicting requests from an unsuspecting operator.[15, 30] Likewise, in applications such as assistive robotics[10–13] and surgical robotics,[4] system instability may put human life in danger. Therefore, from the control theory perspective, stability analysis of shared control systems is essential to ensure performance within safety limits. Few results (e.g., refs. [16, 31, 32]) provided rigorous mathematical analysis to guarantee the stability of shared control systems in their applications. In ref. [31], Ren and Beard considered the stability of shared control systems using control Lyapunov functions (CLFs). Wang and Liu[16] developed an adaptive shared controller and provided stability analysis for an assistive robot to track the time-varying human inputs and avoid obstacles. For a similar problem, Jiang et al.[33, 34] presented a shared controller with fixed control allocation and provided Lyapunov-based stability analysis. Recently, model predictive control and fuzzy logic control has been used for obstacle avoidance in shared control applications;[35–37] however, many of these approaches lack rigorous stability guarantees.

This paper considers execution-level shared control of a mobile robot using adaptive control allocation for applications where a human agent is navigating a semi-autonomous mobile robot in an environment with obstacles. The human agent has a desired navigation objective, e.g., a person on a smart wheelchair may want to go to a certain aisle in a shop; the objective may not be known to the robot. Based on an objective, the human applies control input to the robot. Further, the semi-autonomous robot can be programmed for obstacle avoidance, where the robot may detect obstacles using on-board sensors. The autonomous obstacle avoidance behavior provides autonomy control input that is necessary for avoiding collision with obstacles. A shared control structure allows the human to achieve his or her navigation objective while the autonomy steers the robot away from any obstacles. The presented work primarily focuses on two problems. First, how to appropriately fuse the human and the autonomy control inputs that conserves their individual control objectives. Second, how to design the autonomy control input for obstacle avoidance that also takes into account human inputs. To fuse the human and the autonomy control inputs, a near-continuous state-dependent control allocation function is developed that adaptively weighs the two control inputs based on the position of the robot relative to obstacles. In addition, the control allocation function ensures relatively smooth transition of control authority between the human and the autonomy. As opposed to traded or switched control in refs. [8, 18, 19, 22, 32, 38], where interaction between the robot and the operator exhibits a switching behavior, the presented adaptive shared controller reflects both the human and the autonomy inputs. Further, the collision avoidance objective is achieved by developing a sliding mode controller using harmonic potential fields to obtain the autonomy control input. The

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

obstacles are modeled using harmonic potential fields. The positive charge in the potential field is located at the center of the obstacle, and the location of the negative charge is chosen to reduce the deviation of the robot from its human-intended trajectory for improved user interaction. In contrast to refs. [16, 39], the presented controller includes a feed-forward robust element to compensate for adversarial human inputs, which is crucial to safety in assistive robotics (e.g., to prevent collision or roll-over of smart wheelchairs due to erroneous human inputs). Due to robustness to adverse human actions, the robot does not have to halt when human inputs contradict the collision avoidance objective, as in ref. [16], rather the autonomy ensures the robot reaches a safe state-space without interruption. The satisficing approach in ref. [31] requires that CLFs be obtained for each obstacle space to determine stabilizing and destabilizing control input sets for obstacle avoidance. However, finding CLFs can be a challenge. As opposed to ref. [31], the presented work employs a sliding mode controller to track a desired gradient of the potential field to avoid collision. Moreover, the stability of the closed-loop human–robot system in the presence of assistive as well as adversarial human control inputs guarantees finite-time stability, i.e., the robot will avoid obstacles in finite time to reach a safe state-space. The contributions of the paper can be summarized as follows: (a) a novel execution-level shared controller using a near-continuous adaptive control allocation function is developed to assist a human operator to navigate a mobile robot safely, i.e., without collision, in an environment with obstacles, (b) the sliding mode control structure includes feed-forward element to robustify against adverse human inputs, (c) Lyapunov-based stability analysis is provided in the presence of assistive and adversarial human inputs, and (d) the controller guarantees that the obstacle avoidance objective is achieved in finite time.

2. Problem Formulation

Consider the problem where a wheeled mobile robot (WMR) is navigating in a complex environment with obstacles. The motion of the WMR is governed by the following kinematics:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} \tag{1}$$

where $[x(t)\ y(t)]^T = p(t) \in \mathbb{R}^2$, $\theta(t) \in \mathbb{R}$ denote the position and orientation, respectively, of the WMR, and $\upsilon \in \mathbb{R}^+$ is the forward linear velocity. The angular velocity $\omega(t) \in \mathbb{R}$ assumes the shared control structure as

$$\omega = k_s u_a + (1 - k_s) u_h \tag{2}$$

where $u_a(t) \in \mathbb{R}$, $u_h(t) \in \mathbb{R}$ are the autonomy and the human control inputs, respectively, and the function $k_s(p)$ determines the allocation of control authority between the autonomy and the human, such that $0 \leq k_s \leq 1$. The human control input $u_h(t)$ is assumed to be bounded such that $|u_h(t)| \leq \mathcal{U}_h$, for $\mathcal{U}_h \in \mathbb{R}^+$.

The time-varying input $u_h(t)$ can be directly supplied by the human (e.g., using a joystick or keyboard). Alternately, $u_h(t)$ can be designed to achieve a human-specified positioning objective for the robot. For example, $u_h(t)$ can be designed to track the time-varying position of a human in scenarios where the robot is to follow the human,[16] or $u_h(t)$ can be obtained for the robot to follow a human-encoded desired trajectory.[32] Further, the obstacle avoidance controller on-board a WMR provides $u_a(t)$ to the robot based on the information gathered from the environment. Appropriate fusion of $u_h(t)$ and $u_a(t)$ in Eq. (2) is crucial to the performance and safety of the system.

Given the system in Eqs. (1) and (2) and the human input $u_h(t)$, the goal is to determine $u_a(t)$ and the control allocation weight $k_s(p)$ that guarantees obstacle avoidance.

3. Obstacle Avoidance Controller Development

The control objective is to design the autonomy control input $u_a(t)$ that will guarantee that the semi-autonomous robot safely avoids obstacles as the human navigates the robot through a complex environment. In adaptive shared control, careful attention must be given when the autonomy and

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

human inputs conflict. Due to environmental or physio-psychological factors, the human control input may deviate from a desirable action set to be deemed adversarial to the goal of obstacle avoidance. Such situations can be dangerous, and therefore the movement of the robot is stopped in ref. [16]. In contrast to ref. [16], in this paper, a robust control approach is taken to manage conflicting human inputs without the need to stop the robot. The properties of the presented obstacle avoidance controller are given below.

Property 1. The controller must guarantee obstacle avoidance with no assistance from the human, i.e., $u_h = 0$, when the robot is close to the obstacles.

Property 2. The controller must guarantee obstacle avoidance in the presence of adverse human inputs when the robot is close to the obstacles.

Property 3. The obstacle avoidance controller must be stable in the sense of Lyapunov.

Property 1 states that the autonomy can generate sufficient control effort to safely avoid obstacles in the absence of human inputs. Property 2 states that the obstacle avoidance controller can detect the presence of adversarial human inputs and maintains robustness against them. It is to be noted that the human action is called adversarial only when the robot is sufficiently close to the obstacles, and the selected human action may result in collision. The formal stability definition in Property 3 enables to derive necessary conditions for stability of the closed-loop system.

The subsequent sections are devoted to development and analysis of obstacle avoidance controller based on the control objective defined above and to satisfy Properties 1–3. A harmonic potential field-based obstacle avoidance controller is developed in Section 3.1. Section 3.2 analyzes the stability of the controller and provides a bound on the convergence time. Section 3.3 outlines an approach to select goal point for the robot as it approaches the obstacles. The case of multiple obstacles is studied in Section 3.4 by considering different obstacle geometries.

3.1. Control architecture

A harmonic potential field, also known as Coulomb potential, exhibits well-defined equilibrium points as against artificial potential fields that may contain local maxima or minima. Inspired by the work in ref. [40], harmonic potential fields are developed to create feasible non-holonomic trajectories to guarantee obstacle avoidance. Consider the WMR configuration space contains n obstacles, and let (x_j, y_j) be the center of the obstacle $j = \{1, 2, \ldots, n\}$. In this paper, the obstacles are considered to be circular for simplicity; however, other complex shapes can be accommodated through diffeomorphic mappings.[40,41] These shapes can include overlapping obstacles and other obstacle structures. Let (x_g, y_g) be the position of the goal point. Given the goal point and the obstacles, the harmonic potential field $\phi_j(t)$ for each obstacle can be designed as (see Fig. 1)

$$\begin{aligned} \phi_j &= q_g \ln\left(\frac{1}{d_g}\right) + q_j \ln\left(\frac{1}{d_j}\right) \\ d_g^2 &= (x - x_g)^2 + (y - y_g)^2 \\ d_j^2 &= (x - x_j)^2 + (y - y_j)^2 \end{aligned} \tag{3}$$

where $d_g(t) \in \mathbb{R}^+$ is the distance of the robot from the goal point, $d_j(t) \in \mathbb{R}^+$ is the distance of the robot from the obstacle j, $q_g \in \mathbb{R}^-$ denotes a unit negative charge at the goal point, $q_j \in \mathbb{R}^+$ is the electric charge at the obstacle j of magnitude

$$q_j = \frac{r_{oj}}{r_{oj} + D_j} \tag{4}$$

where $r_{oj} \in \mathbb{R}^+$ is the radius of the obstacle j, and $D_j \in \mathbb{R}^+$ denotes the distance from the obstacle j to the goal point.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

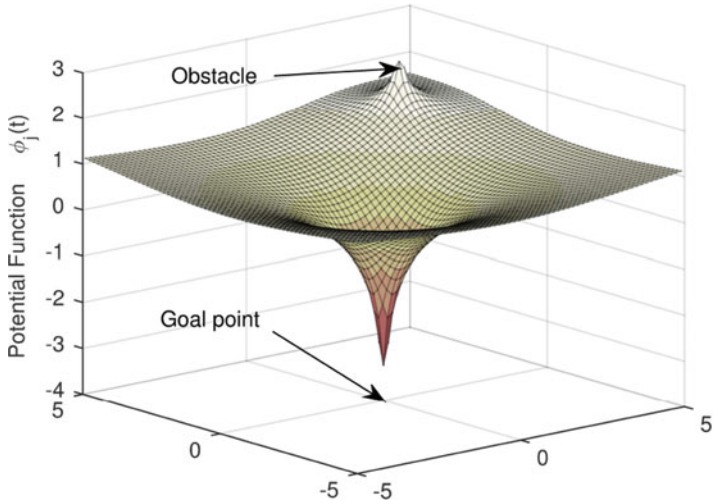


Fig. 1. Harmonic potential field $\phi_j(t)$ obtained using a negative charge at the goal point and a positive charge at the obstacle.

Based on Eq. (3), the desired orientation $\theta_d(t)$ of the robot can be obtained as the negative gradient of the potential field $\phi_j(t)$ with respect to $p(t)$ as

$$\theta_d(t) = \arctan 2\left(-\frac{\partial \phi_j}{\partial y}, -\frac{\partial \phi_j}{\partial x}\right) \tag{5}$$

In Eq. (5), the mapping $\arctan 2(\cdot) : \mathbb{R}^2 \to \mathbb{R}$ denotes the four quadrant inverse tangent function and $\theta_d(t) \in (-\pi, \pi]$. Based on the definition of the harmonic potential field in Eq. (3), if the angular trajectories $\theta(t)$ in Eq. (1) track $\theta_d(t)$ in Eq. (5), then collision with the obstacles can be avoided.

To design the obstacle avoidance controller, let $\tilde{\theta}_d(t)$ be the difference between the current orientation $\theta(t)$ and the desired orientation $\theta_d(t)$ as

$$\tilde{\theta} \triangleq \theta - \theta_d \tag{6}$$

Taking the time derivative of Eq. (6) and substituting Eqs. (1) and (2), the open-loop error system can be obtained as

$$\dot{\tilde{\theta}} = k_s u_a + (1 - k_s) u_h - \dot{\theta}_d \tag{7}$$

In Eq. (7), the time derivative $\dot{\theta}_d(t)$ of the continuous function in Eq. (5) is bounded, i.e., $\dot{\theta}_d(t) \in \mathcal{L}_\infty$. The expression for $\dot{\theta}_d$ can be obtained by taking the derivative of Eq. (5) along the trajectories of the dynamical system in Eq. (1) as

$$\dot{\theta}_d = \nabla_p \theta_d \dot{p} \tag{8}$$

where $\nabla_p \theta_d \in \mathbb{R}^{1\times 2}$ is defined as

$$\nabla_p \theta_d \triangleq \left[\frac{\partial \theta_d}{\partial x} \quad \frac{\partial \theta_d}{\partial y}\right]$$

Consider the obstacle geometry as shown in Fig. 2. The configuration space of the WMR is divided into safe set $\mathcal{M}_s$, danger set $\mathcal{M}_d$, and hysteresis set $\mathcal{M}_h$ defined as

$$\mathcal{M}_s \triangleq \left\{(x, y) \mid r_{hj} \leq d_j \ \forall j = 1, \ldots, n\right\} \tag{9}$$

$$\mathcal{M}_d \triangleq \left\{(x, y) \mid r_{oj} \leq d_j \leq r_{dj} \ \forall j = 1, \ldots, n\right\} \tag{10}$$

$$\mathcal{M}_h \triangleq \left\{(x, y) \mid r_{dj} \leq d_j < r_{hj} \ \forall j = 1, \ldots, n\right\} \tag{11}$$

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

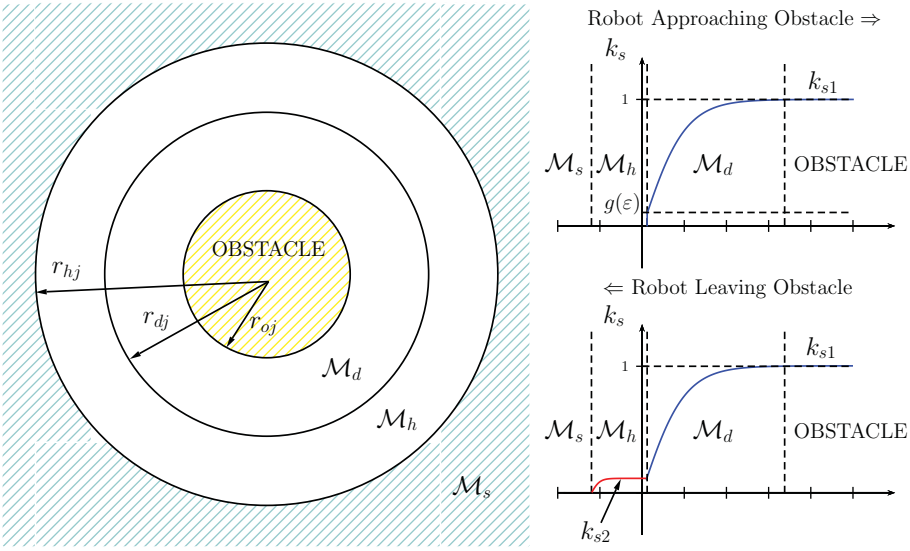


Fig. 2. Schematic diagram showing the control allocation weights k_{s1} and k_{s2} as a function of the state of the robot and the obstacle geometry, where $\mathcal{M}_s$ indicates the safe set, $\mathcal{M}_h$ is the hysteresis set, and $\mathcal{M}_d$ represents the danger set.

where $r_{dj}, r_{hj} \in \mathbb{R}^+$ are as shown in Fig. 2. $\mathcal{M}_h$ acts as a buffer preventing the robot from hunting, i.e., repeated entering and exiting $\mathcal{M}_d$, at the boundary between $\mathcal{M}_h$ and $\mathcal{M}_d$. The radius $r_{dj} > r_{oj}$ must be selected to be sufficiently large to ensure that a robot traveling at maximum linear speed can avoid collision with an obstacle given the limit on its angular velocity. It is relatively straightforward to obtain an expression for minimum r_{dj} using the bounds on the robot's velocities and the control allocation weights shown in Fig. 2. The radius r_{hj} is selected to be greater than r_{dj} to obtain a non-empty hysteresis set, so that any noise in the measurement of $d_j(t)$ or erroneous human inputs will not result in hunting. In practice, r_{hj} can be selected by trial and error to ensure satisfactory performance of the system. In addition, since the controller development assumes the robot to be a point mass, the radius (or maximum dimension) of the robot should be added to r_{oj}, r_{dj}, and r_{hj}.

Based on the open-loop error system in Eq. (7), the autonomy input can be designed as

$$u_a = -\left(k_{a1}\,\mathrm{sgn}(\tilde{\theta})\left|\tilde{\theta}\right|^{\alpha} - \frac{\dot{\theta}_d}{k_s} + \frac{k_{a2}}{k_s}|\,\mathrm{sgn}\,(u_h) + \mathrm{sgn}\,(\tilde{\theta})|\,\mathrm{sgn}\,(u_h)|\,\mathrm{sgn}\,(\tilde{\theta})\right) \tag{12}$$

where $k_{a1}, k_{a2} \in \mathbb{R}^+$ are control gains, $\alpha \in \mathbb{R}^+$ is a constant, and $\dot{\theta}_d(t)$ can be obtained using Eq. (8). The robust terminal sliding mode controller $u_a(t)$ in Eq. (12) is designed for the robot to track the desired potential gradient $\theta_d(t)$ to avoid obstacles. The first term in $u_a(t)$ is designed to regulate the tracking error $\tilde{\theta}(t)$ to origin. It should be noted that the first term in Eq. (12) is continuous in time. The last term in Eq. (12) maintains robustness against adverse human actions. In Eq. (12), adverse human actions are identified by comparing the direction of the human input, $|\,\mathrm{sgn}\,(u_h)| = 1$ for counter-clockwise angular velocity, $|\,\mathrm{sgn}\,(u_h)| = -1$ for clockwise angular velocity, and $|\,\mathrm{sgn}\,(u_h)| = 0$ for $u_h = 0$, with the sign of the angular error $|\,\mathrm{sgn}\,(\tilde{\theta})|$, where the angles measured counter-clockwise are positive. It can be seen from Eqs. (2) and (12) that if the control gain k_{a2} is selected sufficiently large, then adverse human actions will always be dominated to avoid obstacles. The condition on gain k_{a2} is derived in Section 3.1. Given the design of the last term in Eq. (12), the control input $u_a(t)$ is continuous when no human input is applied or when the direction of human input is the same as that of the autonomy. Alternatively, $u_a(t)$ is discontinuous only when human exerts an input that is adversarial to the goal of obstacle avoidance. Chattering of the $\theta(t)$ trajectory around origin is only possible when the human exerts *persistent* adversarial input with high frequency. However, such scenarios can be deemed unpractical, and therefore the controller in Eq. (12) will not exhibit chattering.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

Clearly, from Properties 1 and 2, the autonomy control input is warranted only when the robot is in the vicinity of obstacles, i.e., for $p(t) \in \mathcal{M}_d$ and when the robot enters the hysteresis $\mathcal{M}_h$ from $\mathcal{M}_d$. In $p(t) \in \mathcal{M}_s$ and when the robot travels from $\mathcal{M}_s$ to $\mathcal{M}_h$, the autonomy control input $u_a(t) = 0$. Therefore, the control allocation function $k_s(p) \in \mathbb{R}$ in Eqs. (2) and (12) can be defined as

$$k_s \triangleq \begin{cases} 0 & \text{if } p(t) \in \mathcal{M}_s \\ k_{s0} & \text{if } p(t) \in \mathcal{M}_h \\ k_{s1} & \text{if } p(t) \in \mathcal{M}_d \end{cases} \tag{13}$$

The function $k_{s0}(p)$ in Eq. (13) is defined based on the trajectory of the robot coming into the region $\mathcal{M}_h$ from $\mathcal{M}_d$ or $\mathcal{M}_s$ as

$$k_{s0} \triangleq \begin{cases} k_{s2} & \text{if robot enters } \mathcal{M}_h \text{ from } \mathcal{M}_d \\ 0 & \text{if robot enters } \mathcal{M}_h \text{ from } \mathcal{M}_s \end{cases} \tag{14}$$

$k_{s1}(p), k_{s2}(p) \in \mathbb{R}$ in Eqs. (13) and (14), respectively, are defined as the following continuous functions:

$$k_{s1} \triangleq \tanh\left(\beta \frac{r_{dj} - d_j}{r_{dj} - r_{oj}} + \varepsilon\right) \tag{15}$$

$$k_{s2} \triangleq \tanh(\varepsilon) \tanh\left(\gamma \frac{r_{hj} - d_j}{r_{hj} - r_{dj}}\right) \tag{16}$$

where the constants $\beta, \gamma > 0$, and $\varepsilon \in \mathbb{R}^+$ is a small constant. For sufficiently large β and γ, $k_{s1} \in [\tanh(\varepsilon), 1)$ and $k_{s2} \in [\varepsilon_1, \tanh(\varepsilon))$, where $\varepsilon_1 > 0$ since the definition of hysteresis set in Eq. (11) excludes the boundary set of radius r_{hj}. The constant ε ensures positive control gain when the robot enters $\mathcal{M}_d$, and maintains continuous transition of the gain k_s as the robot leaves $\mathcal{M}_d$ to enter $\mathcal{M}_h$. In practice, ε can be small but it should provide acceptable performance in $\mathcal{M}_h$ against hunting. It can be seen from Eqs. (13)–(16) that $k_s(p) \in [0, 1)$ assumes small values when the robot is away from the obstacles thus providing more control authority to the human, and the control authority slides to autonomy as the robot approaches obstacles. Specifically, the human agent assumes complete control when the robot is in the safe set. The functions $k_{s1}(p), k_{s2}(p)$ enable continuous allocation of the control authority between the human and autonomy (see Fig. 2) when $p(t) \in \mathcal{M}_d, \mathcal{M}_h$. In addition, using Eqs. (15) and (16), $k_{s2} \approx k_{s1}$ when $d_j = r_{dj}$. In other words, when the robot is leaving an obstacle, the allocation function $k_s(p)$ is almost continuous at the border of $\mathcal{M}_d$ and $\mathcal{M}_h$. The discontinuity in $k_s(p)$ only exists at the border of $\mathcal{M}_d$ and $\mathcal{M}_h$ when k_p goes from 0 to $\tanh(\varepsilon)$ as the robot approaches an obstacle, and at the border of $\mathcal{M}_h$ and $\mathcal{M}_s$ when k_p goes from ε_1 to 0 as the robot leaves $\mathcal{M}_h$ after traveling in $\mathcal{M}_d$. This near-continuous control allocation may help in faster learning and improving the user experience. The constants β and γ in Eqs. (15) and (16), respectively, can be designed to obtain the desired rate of change in the control authority, i.e., how fast the autonomy gains control over human and *vice versa*.

Substituting Eq. (12) in Eq. (7), the closed-loop error system can be obtained as

$$\begin{aligned} \dot{\tilde{\theta}} = &- k_s\left(k_{a1}\,\mathrm{sgn}(\tilde{\theta})\left|\tilde{\theta}\right|^{\alpha} - \frac{\dot{\theta}_d}{k_s} + \frac{k_{a2}}{k_s}|\,\mathrm{sgn}(u_h) + \mathrm{sgn}(\tilde{\theta})||\,\mathrm{sgn}(u_h)|\,\mathrm{sgn}(\tilde{\theta})\right) \\ &+ (1 - k_s)u_h - \dot{\theta}_d. \end{aligned} \tag{17}$$

The closed-loop error system in Eq. (17) expresses the dynamics of the tracking error $\tilde{\theta}(t)$ when the autonomy control input $u_a(t)$ is applied. Therefore, it is crucial to examine the stability of Eq. (17) to determine whether the developed controller can avoid obstacles. The following section provides rigorous stability analysis using Lyapunov theory.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

3.2. Convergence analysis

Lemma 1. *Given the system in Eqs. (1) and (2), for any human control input $u_h(t)$, the autonomy controller $u_a(t)$ in Eq. (12) tracks the orientation of the robot along the desired trajectory, i.e., $\theta(t) \to \theta_d(t)$. [Property 3]*

Proof. The stability of the controller in Eq. (12) is analyzed for the cases when $u_h(t)$ is assistive, $u_h(t)$ is adversarial, and $u_h(t) = 0$ when $p(t) \in \mathcal{M}_d$ or the robot enters $\mathcal{M}_h$ from $\mathcal{M}_d$.

Consider a Lyapunov candidate function $V(t) \in \mathbb{R}^+$ as

$$V \triangleq \frac{1}{2}\tilde{\theta}^2 \tag{18}$$

where $\tilde{\theta}(t)$ is defined in Eq. (6). Taking the time derivative of $V(t)$ along the system trajectory in Eq. (1) and substituting Eq. (17), the Lyapunov derivative can be obtained as

$$\dot{V} = -k_s k_{a1}\tilde{\theta}\,\mathrm{sgn}(\tilde{\theta})\left|\tilde{\theta}\right|^{\alpha} + (1-k_s)\tilde{\theta}u_h - k_{a2}|\,\mathrm{sgn}\,(u_h) + \mathrm{sgn}\,(\tilde{\theta})|\,\mathrm{sgn}\,(u_h)|\tilde{\theta}\,\mathrm{sgn}\,(\tilde{\theta}) \tag{19}$$

Consider the case when the human input $u_h(t)$ is assisting autonomy to track $\theta_d(t)$, i.e., when $\mathrm{sgn}\,(u_h) = -\,\mathrm{sgn}\,(\tilde{\theta})$. The Lyapunov derivative in Eq. (19) can be written as

$$\dot{V} = -k_s k_{a1}\tilde{\theta}\,\mathrm{sgn}(\tilde{\theta})\left|\tilde{\theta}\right|^{\alpha} - (1-k_s)|u_h|\tilde{\theta}\,\mathrm{sgn}\,(\tilde{\theta}) \tag{20}$$

where $u_h = |u_h|\,\mathrm{sgn}\,(u_h)$ along with $\mathrm{sgn}\,(u_h) = -\,\mathrm{sgn}\,(\tilde{\theta})$ is used. After substituting $\tilde{\theta} = \left|\tilde{\theta}\right|^{\alpha}\left|\tilde{\theta}\right|^{1-\alpha}\mathrm{sgn}\,(\tilde{\theta})$ and simplifying, the Lyapunov derivative can be obtained as

$$\dot{V} = -\left|\tilde{\theta}\right|^{\alpha}\left(k_s k_{a1}\left|\tilde{\theta}\right| + (1-k_s)|u_h|\left|\tilde{\theta}\right|^{1-\alpha}\right) \tag{21}$$

From Eqs. (11), (15), and (16), $k_s(p) > 0$. In Eq. (21), it can be seen that the bracketed quantity is a positive definite function of $\tilde{\theta}(t)$. Therefore, the Lyapunov derivative is negative definite, $\dot{V} < 0$. Based on $V > 0$ and $\dot{V} < 0$, it can be concluded that $\tilde{\theta}(t) \in \mathcal{L}_2$. From the facts that $\dot{\theta}_d(t) \in \mathcal{L}_\infty$ and $k_s(p) \neq 0$, the control input in Eq. (12) is bounded, $u_a(t) \in \mathcal{L}_\infty$. In addition, the constant α in Eq. (12) can be chosen to satisfy $0 < \alpha < 1$. Then, the closed-loop system in Eq. (17) is finite-time-stable.[42] Specifically, the developed obstacle avoidance controller working in collaboration with the human agent can guarantee that the tracking error $\tilde{\theta}(t)$ goes to zero in finite time.

Now consider the case when the human control input $u_h(t)$ is adverse to the goal of obstacle avoidance, i.e., $\mathrm{sgn}\,(u_h) = \mathrm{sgn}\,(\tilde{\theta})$. The Lyapunov derivative in Eq. (19) can be written as

$$\dot{V} \leq -k_s k_{a1}\tilde{\theta}\,\mathrm{sgn}(\tilde{\theta})\left|\tilde{\theta}\right|^{\alpha} - 2k_{a2}\tilde{\theta}\,\mathrm{sgn}\,(\tilde{\theta}) + \mathcal{U}_h\tilde{\theta}\,\mathrm{sgn}\,\tilde{\theta} \tag{22}$$

where $|u_h(t)| \leq \mathcal{U}_h$ is used. To dominate adverse human inputs, the robust gain k_{a2} in Eq. (12) can be designed as

$$k_{a2} > \frac{\mathcal{U}_h}{2} \tag{23}$$

Based on Eqs. (22) and (23), for $\mathcal{K} \in \mathbb{R}^+$, the Lyapunov derivative can be written as

$$\dot{V} \leq -\left|\tilde{\theta}\right|^{\alpha}\left(k_s k_{a1}\tilde{\theta}\,\mathrm{sgn}(\tilde{\theta}) + \mathcal{K}\left|\tilde{\theta}\right|^{1-\alpha}\right) \tag{24}$$

The bracketed term in Eq. (24) is positive definite. For $0 < \alpha < 1$, the closed-loop system in Eq. (17) is finite-time-stable, and it can be shown that $u_a(t) \in \mathcal{L}_\infty$. Specifically, the controller guarantees finite-time tracking of $\theta_d(t)$ and robustness against adverse human inputs provided that the robust gain in Eq. (12) is chosen according to Eq. (23).

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

In the absence of human input, i.e., $u_h = 0$, the Lyapunov derivative in Eq. (19) becomes

$$\dot{V} = -k_s k_{a1} \left|\tilde{\theta}\right|^{\alpha+1} \tag{25}$$

From Eq. (25), it is clear that $\dot{V} > 0$, and for any $0 < \alpha < 1$, the closed-loop system in Eq. (17) is finite-time-stable.

□

Lemma 1 proves that a WMR with shared angular velocity $\omega(t)$ tracks the desired trajectory $\theta_d(t)$. However, for collision avoidance, the robot must track the gradient of the potential field, i.e., $\tilde{\theta}(t) \to 0$, before reaching the obstacle boundary. For a robot moving with constant linear velocity, it is necessary to establish a lower bound on the angular velocity to guarantee collision avoidance.

Theorem 1. *The stabilizing shared controller $\omega(t)$, and the obstacle geometry, r_{oj} and r_{dj}, there exists a lower bound on the autonomy control input $u_a(t)$ in Eq. (12) that guarantees collision avoidance.*

Proof. The Lyapunov derivative in Eq. (21) can be written as

$$\dot{V} = -k_s k_{a1} \left|\tilde{\theta}\right|^{\alpha+1} - (1 - k_s)|u_h| \left|\tilde{\theta}\right| \leq -k_s k_{a1} \left|\tilde{\theta}\right|^{\alpha+1} \tag{26}$$

An inequality identical to Eq. (26) can be obtained using Eqs. (24) and (25) when the human input is adversarial and $u_h = 0$, respectively. From the solution of the differential equation in Eq. (26), the upper bound on the convergence time for $\theta(t) \to \theta_d(t)$ can be obtained as

$$t \leq \frac{\left|\tilde{\theta}_o\right|^2}{2^{\zeta+1}(1-\zeta)k_{a1}\tanh(\varepsilon)} \tag{27}$$

where $\tilde{\theta}_o \in \mathbb{R}$ is the initial tracking error and $\zeta = (\alpha + 1)/2$. In Eq. (27), the minimum value of $k_s = \tanh(\varepsilon)$ is used. For the goal point generation approach in Section 3.3, it can be shown that the initial error $\left|\tilde{\theta}_o\right| \leq \pi/2$. Let $t_c \in \mathbb{R}^+$ be the minimum time to reach any obstacle from the boundary of $\mathcal{M}_d$, and $\bar{\upsilon} \in \mathbb{R}$ be the maximum linear velocity of the robot. Hence, we have $t_c = \min_j(r_{dj} - r_{oj})/\bar{\upsilon}$. The sufficient condition to avoid collision is $t < t_c$, i.e., the robot must track the desired gradient curve before reaching obstacle. Based on the inequality $t < t_c$, the lower bound on $u_a(t)$ can be obtained by selecting the control gain as

$$k_{a1} > \frac{\bar{\upsilon}\left|\tilde{\theta}_o\right|^2}{2^{\zeta+1}(1-\zeta)\tanh(\varepsilon)(r_{dj} - r_{oj})} \quad \forall j \tag{28}$$

Theorem 1 and Lemma 1 prove that the shared control input (2) along with Eqs. (23) and (28) guarantee obstacle avoidance in the presence of no human inputs [Property 1] and adversarial inputs [Property 2]. □

It is known that the harmonic potential field has an unstable equilibrium point apart from the two singular points, i.e., the source and the sink. The unstable equilibrium point lies on the obstacle along the singular line joining the obstacle center and the goal point as shown in Fig. 3. The unstable equilibrium point can only be reached by traveling along the singular line. However, the curvature of the gradient lines along the singular line approaches infinity, $\dot{\theta}_d(t) \to \infty$, at the equilibrium point, and therefore $\theta_d(t)$ cannot be tracked with finite control resources. In addition, since the equilibrium point is on the obstacle, collision avoidance cannot be guaranteed in this case. To prevent the robot from traveling along the singular line, a small perturbation can be added to the angular velocity $\omega(t)$. The perturbation causes deviation in the robot's position from the singular line, enabling the robot track the gradient with finite control effort. In practice, the noise in the position and velocity measurements can be sufficient to avoid this situation.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

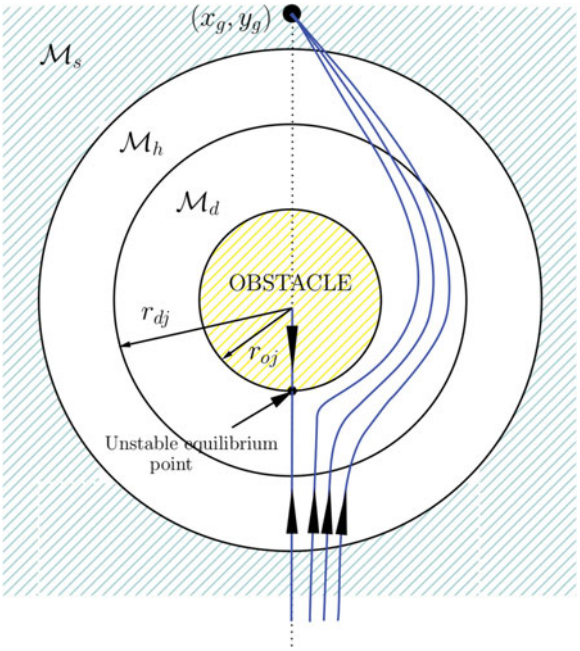


Fig. 3. Harmonic potential field showing the stable goal point and the unstable equilibrium point.

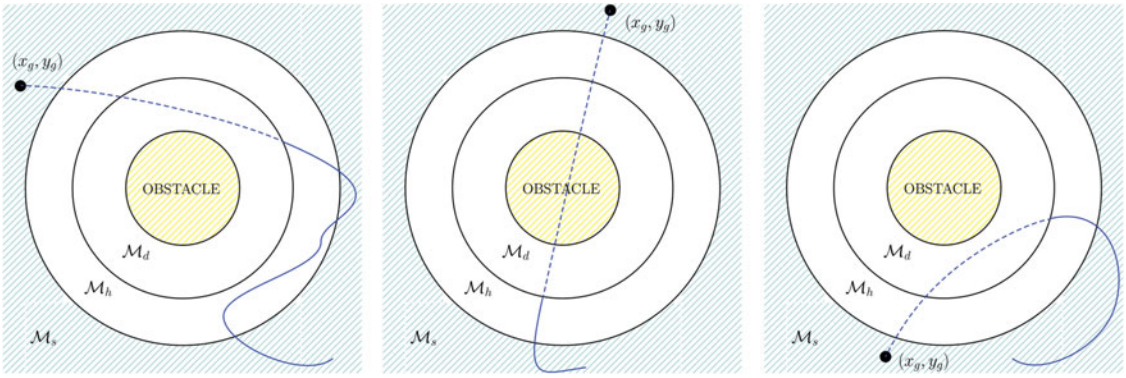


Fig. 4. The goal point for the harmonic potential field obtained by extrapolating the motion of the robot prior to entering $\mathcal{M}_d$.

3.3. Goal point generation

The human determines the trajectory of the robot in $\mathcal{M}_s$. As discussed earlier, when the robot enters the danger set, $p(t) \in \mathcal{M}_d$, the potential field corresponding to that obstacle is switched on for obstacle avoidance. The location of the goal point, where the negative charge of the potential resides, affects the trajectory of the robot in $\mathcal{M}_d$. In this paper, the goal point is selected based on human intention prior to the robot entering $\mathcal{M}_d$. The goal point is chosen to reduce the deviation of robot from the human intended trajectory as it leaves $\mathcal{M}_h$.

The digital control system is considered to have finite memory to store the position of the robot from the last m time steps, p_k for $k = 1, 2, \ldots, m$. Using p_k and assuming the WMR rolls without sliding (non-holonomic constraint), the goal point can be obtained by polynomial extrapolation as shown in Fig. 4. Given the known obstacle geometry, it can be ensured that the extrapolated goal point lies in $\mathcal{M}_s$. It is to be noted that the goal point is only required to generate harmonic potential field to avoid obstacles, and the goal point may not actually be visited since the human determines the motion of the robot in $\mathcal{M}_s$.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

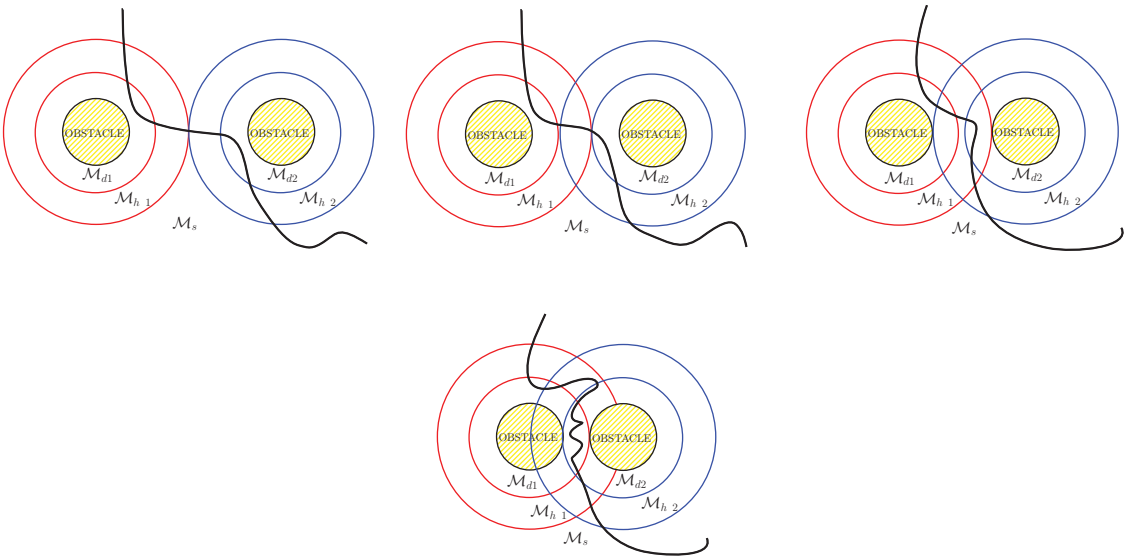


Fig. 5. Sketch showing robot trajectories for different obstacle geometries.

3.4. Multiple obstacles

Given the dependence of autonomy control input in Eq. (12) on the obstacle geometry, special consideration must be given to the case of multiple obstacles, especially when their obstacle geometry overlaps. In the case of multiple obstacles, the potential at any point (x, y) is the result of negative charge at the goal point and all positive charges at the obstacles. Special care must be taken in designing $q_j\ \forall j$ to ensure that the field lines do not go to infinity (resulting in instability), and the field lines do not enter the obstacles (resulting in collision). In ref. [43], a mathematical programming problem is formulated to design q_j to satisfy the above constraints. However, the additional computational burden is unwarranted in the presented application. The developed control allocation function $k_s(p)$ ensures that the potential field has no influence on the robot when $p(t) \in \mathcal{M}_s$. When the robot enters the danger set, $p(t) \in \mathcal{M}_d$, the potential field corresponding to that obstacle is switched on and it turns off only when the robot leaves $\mathcal{M}_h$ or enters $\mathcal{M}_d$ of another obstacle. Depending on the geometry of the obstacles, we consider four cases as shown in Fig. 5. Let $\mathcal{M}_{di}$ be the danger set and $\mathcal{M}_{hi}$ be the hysteresis set of the obstacle i. Consider the following cases: (a) $\mathcal{M}_{hi} \cap \mathcal{M}_{hj} = \emptyset$, (b) $\mathcal{M}_{hi} \cap \mathcal{M}_{hj} \neq \emptyset$ and $\mathcal{M}_{hi} \cap \mathcal{M}_{dj} = \emptyset$, (c) $\mathcal{M}_{hi} \cap \mathcal{M}_{dj} \neq \emptyset$, and (d) $\mathcal{M}_{di} \cap \mathcal{M}_{dj} \neq \emptyset$, for $i \neq j$. Under switching potential field, it is trivial to show that in (a), (b), and (c), the robot will move away from the obstacles. Moreover, in contrast to ref. [40], the designed controller due to inclusion of the hysteresis set will not exhibit hunting. Special consideration is given to the case when the danger sets of two obstacles overlap, i.e., $\mathcal{M}_{di} \cap \mathcal{M}_{dj} \neq \emptyset$. To avoid collision, when $p(t) \in \mathcal{M}_{di} \cap \mathcal{M}_{dj}$, the potential field of to the nearest obstacle can be used to obtain $u_a(t)$. The resultant controller will track the robot along the equidistant line between the obstacles by following a discontinuous gradient $\dot{\theta}_d(t)$ as shown in Fig. 5. The resulting fast switching of the controller can be avoided using various heuristics, e.g., see ref. [40]. To summarize, the design of $k_s(p)$ in Eqs. (13)–(16) enables to treat multiple obstacles case in a similar manner to that of the single obstacle. The potential field of the closest obstacle is used to obtain $u_a(t)$ whenever the robot enters a danger set or sets $\mathcal{M}_{di}$.

4. Experimental Validation

The experiment was conducted in an indoor environment to validate the performance of the developed adaptive execution-level shared controller. A human operator remotely operated the WMR, while the on-board autonomous controller ensured collision avoidance. A stricter scenario is considered where the forward linear velocity of the robot was considered to be constant. If the linear velocity can be varied, and in particularly reduced, in the vicinity of obstacles, then the collision avoidance problem becomes simpler, which can also be verified from Eqs. (12) and (28). Therefore, given the constant linear velocity, the operator signals the angular velocity commands to the robot in order to steer

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

Fig. 6. (a) Experimental setup showing the robot's workspace and the circular boundary of the three obstacles, (b) iRobot Create with on-board computer, and (c) the operator's GUI for remote control of the robot.

it in the workspace. The obstacle avoidance controller designed in Eq. (12) also issues angular velocity commands to avoid collision when the robot enters $\mathcal{M}_d$. To summarize, both the human and the autonomy control inputs are angular velocity or steering commands to the robot. Note that the developed controller does not require that the linear velocity of the robot be constant, and it can be varied by an operator. As proved in Section 3.1, obstacle avoidance is guaranteed if the gain k_{a1} is chosen according to Eq. (28) using the known upper bound on the time-varying linear velocity. The experimental testbed and the results are discussed in the following sections.

4.1. Testbed

The WMR under consideration is a differential drive iRobot Create. A low-level controller tracks the commanded linear and angular velocities by determining the required speed of each wheel. The workspace was about 5.5 m $\times$ 5.5 m which included three circular obstacles of radius 0.37 m as shown in Fig. 6. An OptiTrack motion capture system was used to obtain the time-varying position and orientation of the robot with respect to a known inertial frame. The robot carried a computer running Robotic Operating System (ROS) that received measurements from the OptiTrack system over a wireless network. The ground station included a computer with a user interface developed in MATLAB to display the position of the robot in the environment to enable human to navigate the robot.

The human could steer the robot by pressing the left and the right arrow keys to turn the robot in counter-clockwise and clockwise direction, respectively, to navigate the robot through the environment according to his or her objective. The ground station also used ROS to encode the human input to send it to the robot over the ROS network. The longer the human held down the button, the control input grew linearly in magnitude until it reached a designed saturation level. Further, the computer on-board the robot determined the autonomy control input $u_a(t)$ and the control allocation weight $k_s(p)$ based on its own position and orientation with respect to the obstacles in the map and the human input. Finally, the human input $u_h(t)$ was combined with $u_a(t)$ according to Eq. (2) to drive the robot. The constant linear velocity of the robot was measured to be about 0.3 m/s. The human input, the autonomy input, and the OptiTrack-measured robot position was recorded and time stamped so that the results can be analyzed and presented.

4.2. Results

Three trials were conducted to demonstrate the performance of the presented controller. The plots in Figs. 7–9 are obtained using the data recorded in actual trials. The obstacles are shown in solid yellow circles, and the dotted green lines around the obstacles represent the boundaries of the danger and the hysteresis sets. The trajectory traversed by the robot is shown in blue, where the hollow and the solid blue circles are the start- and end-points, respectively. The black arrows superimposed on the robot's trajectory in Figs. 7–9 indicate that the human exerted a turn velocity $u_h(t)$ to the robot at the corresponding position on its trajectory. Along the direction of the motion, the left and the right arrows indicate counter-clockwise and clockwise turns, respectively. As discussed earlier, the

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

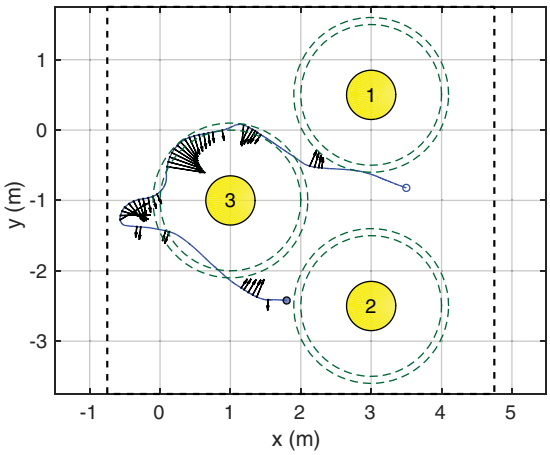


Fig. 7. Trial 1: The time-varying robot trajectory (blue line), obstacles (yellow), and the direction and the relative magnitude of the human's steering control input (black arrows).

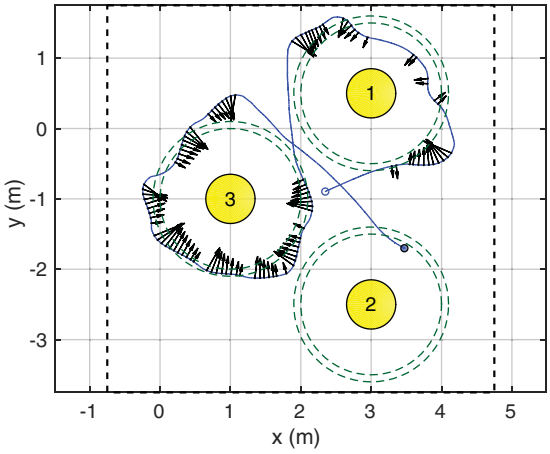


Fig. 8. Trial 2: The time-varying robot trajectory (blue line), obstacles (yellow), and the direction and the relative magnitude of the human's steering control input (black arrows).

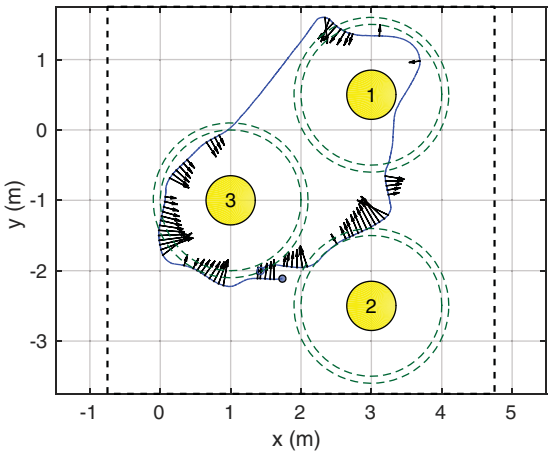


Fig. 9. Trial 3: The time-varying robot trajectory (blue line), obstacles (yellow), and the direction and the relative magnitude of the human's steering control input (black arrows).

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

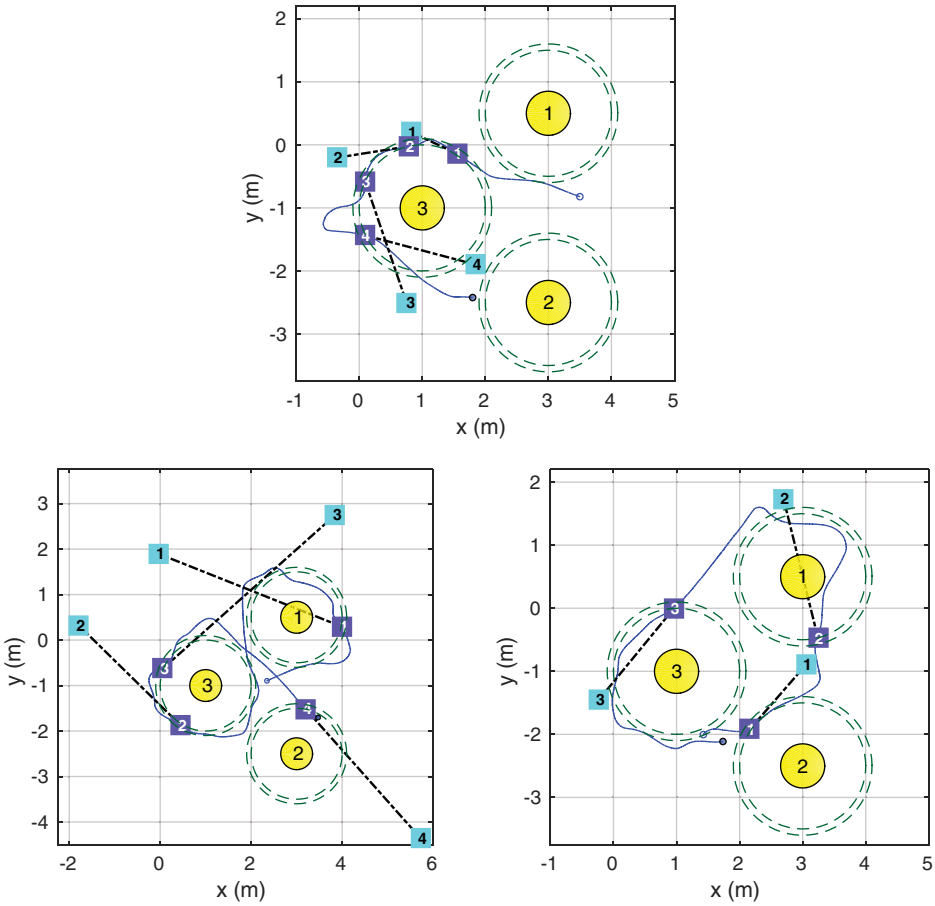


Fig. 10. Trials 1 (top), 2 (left bottom), 3 (right bottom): The time-varying robot trajectory (blue line), obstacles (yellow), and the goal points (cyan boxes) generated as the robot enters the danger set of the obstacles at points marked in purple boxes.

human control input was designed to grow in magnitude if the human held the key pressed for longer duration. Therefore, the length of the arrow is chosen proportional to the magnitude of $u_h(t)$ to clearly demonstrate when the human persistently exerted control on the robot, i.e., shorter arrows corresponds to small angular velocities and *vice versa*. Fig. 10 demonstrates the procedure for generating the goal points as outlined in Section 3.3. The instances when the robot enters the danger set $\mathcal{M}_d$ are shown in purple boxes, while the goal points are shown in cyan boxes. The boxes are connected by dotted black lines to show their correspondence. The numbers in the boxes merely identify a goal point from the rest (i.e., goal point identification number), and it is included for bookkeeping purposes only.

Figs. 7–9 show that occasionally the human input causes the robot to get close to the obstacle which may potentially result in collision. However, it can be seen that the autonomous controller avoids collision by tracking the desired potential gradient, ensuring robustness with respect to adverse human inputs, and varying the control allocation function by sliding the control authority to autonomy. Consider Figs. 7 and 10 corresponding to trial 1. As the robot enters $\mathcal{M}_d$ of obstacle 3, the first goal point is generated, and $u_a(t)$ causes the robot to move away from the obstacle. However, due to the persistent counter-clockwise turn commanded by the human, the robot turns back toward the obstacle as soon as it leaves $\mathcal{M}_h$, which results in the second goal point when the robot enters $\mathcal{M}_d$. Once again the autonomous controller ensures collision avoidance by steering the robot away from the obstacle thus demonstrating the performance of the obstacle avoidance controller. To show the robustness of the developed controller with respect to adversarial human inputs, consider Figs. 8 and 10 and refer to the part of the trajectory after goal point 3 is generated when the robot enters $\mathcal{M}_d$ of obstacle 3. The autonomous controller is steering the robot in counter-clockwise direction away from the obstacle, while the human is commanding the robot to turn clockwise. The robust element in

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

Table I. Minimum distance R_m for a single obstacle with $\beta \in \{.01, 0.1, 1.0\}$ and $\gamma = 1$.

Initial condition	$\beta = 0.01$	$\beta = 0.1$	$\beta = 1.0$
$\mathcal{Z}_1$	0.3756	0.5577	0.7281
$\mathcal{Z}_2$	0.5073	0.6216	0.7877
$\mathcal{Z}_3$	0.5998	0.6935	0.8202

$u_a(t)$ nullifies the effect of the adverse human inputs to guarantee obstacle avoidance. Similarly, the robustness of the developed controller can be observed in Fig. 9 when the robot enters $\mathcal{M}_d$ of obstacles 1 and 3. In Fig. 9, another interesting observation to be made is when the robot enters $\mathcal{M}_d$ of obstacle 1 (corresponding to goal point 2 in Fig. 10). The robot travels close to the singular line and does not steer away until it gets very close to the obstacle due to the structure of the gradient lines close to the unstable equilibrium point as shown in Fig. 3.

In the subsequent results, the performance of the controller was analyzed by changing the design parameters β and γ, and the control gains k_{a1} and k_{a2}. The trajectory of the robot was recorded by varying the parameters and control gains. In addition, the minimum distance $R_m \in \mathbb{R}^+$ from the robot to the center of the obstacle is obtained for the entire length of the trajectory. The minimum distance is defined as

$$R_m \triangleq \min_{(x,y)} \left((x - x_1)^2 + (y - y_1)^2 \right)^{1/2} \tag{29}$$

where (x_1, y_1) is the center of the obstacle, and $(x(t), y(t))$ is the recorded trajectory of the robot.

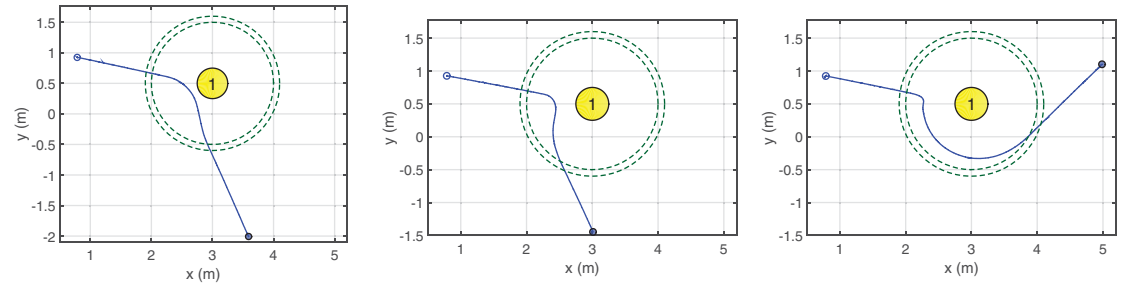
Fig. 11 shows the trajectory of the robot for various cases of $\beta \in \{0.01, 0.1, 1\}$ and $\gamma = 1$, $\mathcal{U}_h = 2$, and $k_{a2} = 1$. Three different initial conditions $(x(0), y(0), \theta(0))$ for the robot were chosen, which are denoted by $\mathcal{Z}_1 \in (0.785, 0.925, -0.2569)$, $\mathcal{Z}_2 \in (0.735, 0.712, -0.2569)$, and $\mathcal{Z}_3 \in (0.689, 0.502, -0.2569)$. The robot trajectories for different values of β for the initial conditions in set $\mathcal{Z}_1$ are shown in Figs. 11(a)–(c), similarly trajectories for the initial conditions in set $\mathcal{Z}_2$ are shown in Figs. 11(d)–(f), and trajectories for the initial condition in set $\mathcal{Z}_3$ are shown in Figs. 11(g)–(i). The initial condition $\mathcal{Z}_1$ is on the singular line with the robot pointing toward the center of the obstacle, while the initial conditions in $\mathcal{Z}_2$ and $\mathcal{Z}_3$ are parallel to the singular line.

The minimum distance R_m of the robot from the obstacle is given in Table I, where the row and column correspond to the row and column of Fig. 11. From Table I, it is clear that increasing β also increases R_m for all initial conditions. Moreover, the farther the initial conditions are from the singularity line, i.e., compare Figs. 11(d)–(f) to 11(g)–(i), the larger R_m becomes. This is expected since the control effort required to avoid obstacles is the largest when the robot is on the singular line and it reduces as the robot moves away from the singular line, which is due to the varying curvature of the gradient lines as shown in Fig. 3. Further, it can be seen from Table I that with increase in β the minimum distance R_m increases. This is because large β allows $k_{s1}(t)$ in Eq. (15) to approach 1 in less time, and when $k_{s1}(t) = 1$ the entire control effort is directed toward obstacle avoidance. Hence, with increase in β, the convergence time t reduces, and hence R_m increases. This is evident from the error plot in Fig. 12, corresponding to the initial conditions in set $\mathcal{Z}_3$, where it is shown that $\tilde{\theta}(t)$ converges faster with larger value of β. Figs. 12(a)–(b) show that, although obstacle avoidance was achieved, the radius of the danger zone was not large enough for $\tilde{\theta}(t)$ to decrease to zero with $\beta \in \{.01, .1\}$. When β was increased to 1, the error $\tilde{\theta}(t)$ converged to zero within the danger zone in about 6 s as shown in Fig. 12(c).

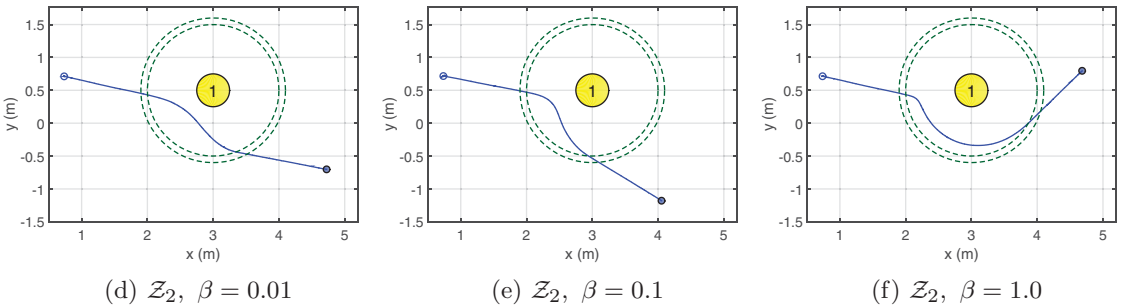
In the next experiment, γ is varied while maintaining β constant. Fig. 13 shows the trajectory of the robot for $\gamma \in \{.01, .1\}$ and $\beta = 1$. The initial condition of the robot is in the set $\mathcal{Z}_2$. Together with Figs. 13 and 11(f), for $\beta = 1$ and $\gamma \in \{.01, .1, 1\}$, the minimum distance is obtained as $R_m = \{0.7871m, 0.7883m, 0.7877m\}$. This shows that γ has little impact on the performance of the system.

To study the effect of control gain k_{a1} on the performance of the controller, k_{a1} is varied from 0.3 to 1.7. Additionally, adversarial human input is exerted persistently and intentionally causing the robot to collide with the obstacle. When the robot enters $\mathcal{M}_d$ from the left side of the workspace (see Fig. 14), the human holds down the left arrow key thus persistently commanding the robot to turn counter-clockwise to force the robot to collide with the obstacle. Fig. 14 shows the robot trajectories for the different values of k_{a1} and $\beta = 1, \gamma = 1, k_{a2} = 1$. The minimum distance R_m is shown in

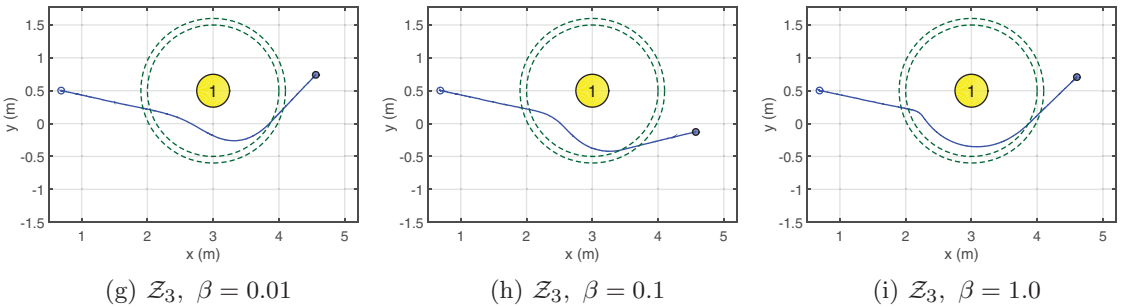
https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press



(a) $\mathcal{Z}_1,\ \beta = 0.01$ (b) $\mathcal{Z}_1,\ \beta = 0.1$ (c) $\mathcal{Z}_1,\ \beta = 1$

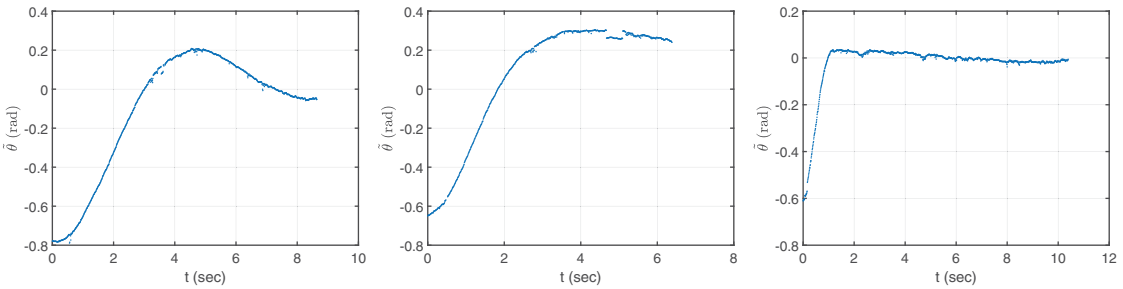


(d) $\mathcal{Z}_2,\ \beta = 0.01$ (e) $\mathcal{Z}_2,\ \beta = 0.1$ (f) $\mathcal{Z}_2,\ \beta = 1.0$



(g) $\mathcal{Z}_3,\ \beta = 0.01$ (h) $\mathcal{Z}_3,\ \beta = 0.1$ (i) $\mathcal{Z}_3,\ \beta = 1.0$

Fig. 11. Experimental results for a single obstacle with $\beta \in \{.01, 0.1, 1.0\}$ and $\gamma = 1$. (a) $\mathcal{Z}_1,\ \beta = 0.01$. (b) $\mathcal{Z}_1,\ \beta = 0.1$. (c) $\mathcal{Z}_1,\ \beta = 1$. (d) $\mathcal{Z}_2,\ \beta = 0.01$. (e) $\mathcal{Z}_2,\ \beta = 0.1$. (f) $\mathcal{Z}_2,\ \beta = 1.0$. (g) $\mathcal{Z}_3,\ \beta = 0.01$. (h) $\mathcal{Z}_3,\ \beta = 0.1$. (i) $\mathcal{Z}_3,\ \beta = 1.0$.



(a) $\mathcal{Z}_3,\ \beta = 0.01$ (b) $\mathcal{Z}_3,\ \beta = 0.1$ (c) $\mathcal{Z}_3,\ \beta = 1.0$

Fig. 12. Experimental results showing the error $\tilde{\theta}(t)$ for the duration that the robot is within the danger zone $p \in \mathcal{M}_d$. (a) $\mathcal{Z}_3,\ \beta = 0.01$. (b) $\mathcal{Z}_3,\ \beta = 0.1$. (c) $\mathcal{Z}_3,\ \beta = 1.0$.

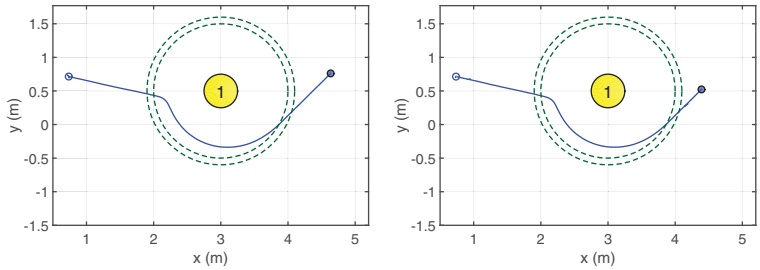
https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

Table II. Minimum distance R_m for a single obstacle where k_{a1} is varied while $\gamma = 1, \beta = 1, k_{a2} = 1$.

	$k_{a1} = 0.3$	$k_{a1} = 0.5$	$k_{a1} = 0.7$	$k_{a1} = 0.9$	$k_{a1} = 1.1$	$k_{a1} = 1.3$	$k_{a1} = 1.5$	$k_{a1} = 1.7$
R_m	0.0685	0.2529	0.4509	0.5693	0.6091	0.5910	0.6896	0.6460

Table III. Minimum distance R_m for a single obstacle where k_{a2} is varied while $\gamma = 1, \beta = 1, k_{a1} = 0.7$.

	$k_{a2} = 1.0$	$k_{a2} = 1.2$	$k_{a2} = 1.4$	$k_{a2} = 1.6$	$k_{a2} = 1.8$	$k_{a2} = 2.$
R_m	0.4683	0.4635	0.4789	0.4682	0.4714	0.4688



(a) $\mathcal{Z}_2,\ \gamma = 0.01$ (b) $\mathcal{Z}_2,\ \gamma = 0.1$

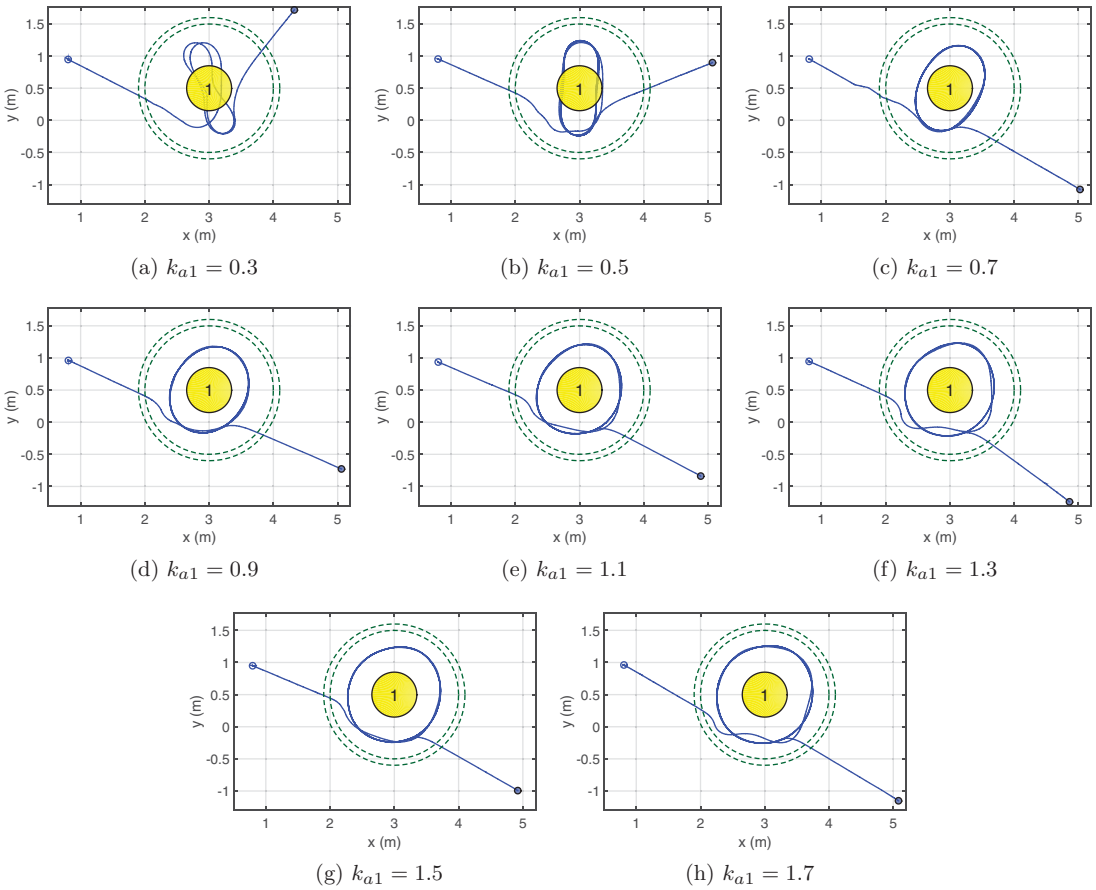
Fig. 13. Experimental results showing the robot trajectories for $\gamma \in \{0.01, 0.1\}$ and $\beta = 1.0$. (a) $\mathcal{Z}_2,\ \gamma = 0.01$. (b) $\mathcal{Z}_2,\ \gamma = 0.1$.

Table II for different values of k_{a1}. From Figs. 14(a)–(b), it is clear that insufficient gain k_{a1} causes the robot to collide with the obstacle. In Figs. 14(c)–(e), where k_{a1} goes from 0.7 to 1.1, there are noticeable improvements in R_m. For k_{a1} ranging from 1.3 to 1.7 as shown in Figs. 14(f)–(h), there is little improvement in the minimum distance.

Due to persistent command, the human input saturates at $u_h(t) = \mathcal{U}_h$. The autonomy control input, on the other hand, attempts to track the desired potential gradient by commanding the robot to turn clockwise. Mathematically, $\text{sign}(u_h) = -\text{sign}(u_a)$. Depending on the values of k_{a2}, β, and $\mathcal{U}_h$, and from Eq. (2), either the term $(1 - k_s)\mathcal{U}_h$ dominates $k_s u_a(t)$ or *vice versa*. If $(1 - k_s)\mathcal{U}_h$ dominates $k_s u_a(t)$, then the resulting angular velocity $w(t)$ will drive the robot close to the obstacle, which increases the control gain k_s (see Eq. (15)) and, as a result, $(1 - k_s)\mathcal{U}_h$ decreases. At a distance d_j^* from the obstacle, the value of the gain k_s is such that $(1 - k_s)\mathcal{U}_h = k_s u_a(t)$ (i.e., the human and the autonomy control efforts cancel each other). However, any deviation from d_j^* will cause either of the control inputs to dominate, which will change k_s forcing the robot to maintain distance d_j^* from the obstacle. In a crude sense, the trajectory at d_j^* can be considered as an equilibrium when persistent human input is applied. It is trivial to show that identical results are obtained if $k_s u_a(t)$ dominates $(1 - k_s)\mathcal{U}_h$. Further, the autonomy control input required to track the desired potential gradient varies depending on the position of the robot in the field. For small values of k_{a1}, the robot cannot track the desired potential gradient when it is in the vicinity of the singular line. Therefore, the robot traverses into the obstacle until the gain k_s becomes sufficiently large to dominate the human control input. As a result, for small k_{a1}, the robot exhibits irregular trajectories shown in Figs. 14(a)–(c). In summary, the control gain k_{a1} should be chosen sufficiently large for the robot to track the potential gradient almost everywhere. If the human persistently exerts adversarial control, then the robot will guarantee obstacle avoidance and follow a circular trajectory in $\mathcal{M}_d$.

The following results demonstrate the performance of the controller when the control gain k_{a2} is varied. It is to be noted that the gain k_{a2} corresponds to the feed-forward term in Eq. (12) that ensures robustness against adversarial human inputs. Fig. 15 shows the trajectories of the robot with persistent adversarial human input similar to the previous results in Fig. 14. The parameter k_{a2} is varied and $\gamma = 1, \beta = 1, k_{a1} = 0.7$. The minimum distance R_m is shown in Table III, where no obvious performance difference is seen by varying k_{a2}. This shows that k_{a2} only needs to satisfy the inequality $k_{a2} \geq \mathcal{U}_h/2$ in order to nullify the human input u_h to avoid collision with obstacles. The

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press



(a) $k_{a1} = 0.3$ (b) $k_{a1} = 0.5$ (c) $k_{a1} = 0.7$

(d) $k_{a1} = 0.9$ (e) $k_{a1} = 1.1$ (f) $k_{a1} = 1.3$

(g) $k_{a1} = 1.5$ (h) $k_{a1} = 1.7$

Fig. 14. Experimental results showing the robot trajectories when the human input is adversarial to the goal of obstacle avoidance. The parameter k_{a1} is varied while $\gamma = 1, \beta = 1, k_{a2} = 1$. (a) $k_{a1} = 0.3$. (b) $k_{a1} = 0.5$. (c) $k_{a1} = 0.7$. (d) $k_{a1} = 0.9$. (e) $k_{a1} = 1.1$. (f) $k_{a1} = 1.3$. (g) $k_{a1} = 1.5$. (h) $k_{a1} = 1.7$.

near-circular trajectories in Fig. 15 are the result of equal control effort in opposite directions exerted by the human and the autonomy, as discussed earlier.

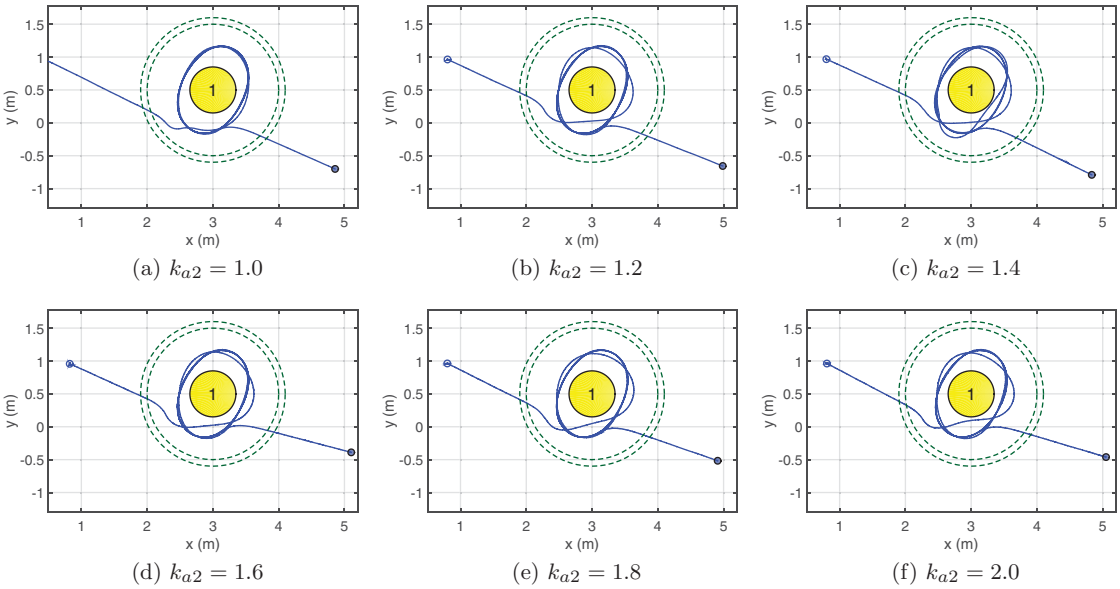
Finally, the performance of the controller is validated for the scenario shown in Fig. 5 when the obstacle geometries overlap. In particular, three cases are considered—(a) the hysteresis zones overlap $\mathcal{M}_{h1} \cap \mathcal{M}_{h2} \neq \emptyset$, (b) the hysteresis and the danger zones overlap $\mathcal{M}_{h1} \cap \mathcal{M}_{d2} \neq \emptyset$, and (c) the danger zones overlap $\mathcal{M}_{d1} \cap \mathcal{M}_{d2} \neq \emptyset$. Fig. 16 shows the robot trajectories for the three cases. It can be seen that the robot trajectories are similar to the anticipated trajectories in Fig. 5 and avoid collision with the obstacles by following a switching potential field as discussed in Section 3.4.

5. Conclusion

An execution-level shared controller is developed for a mobile robot navigating in a complex environment with obstacles. In contrast to much existing work, the paper focuses on rigorous controller development and stability analysis of the shared controller. The controller is shown to be robust to adverse human actions, which can be important for safety of many assistive robotics systems. Using Lyapunov-based stability analysis, the developed shared controller is shown to be finite-time stable. The performance of the shared control system was analyzed through indoor experiments.

One of the avenues for future research is to include human factors, such as fatigue and cognitive workload, into the adaptive control allocation function to improve robustness to erroneous human inputs. Another area of future work is to integrate online obstacle detection methods into the existing framework.

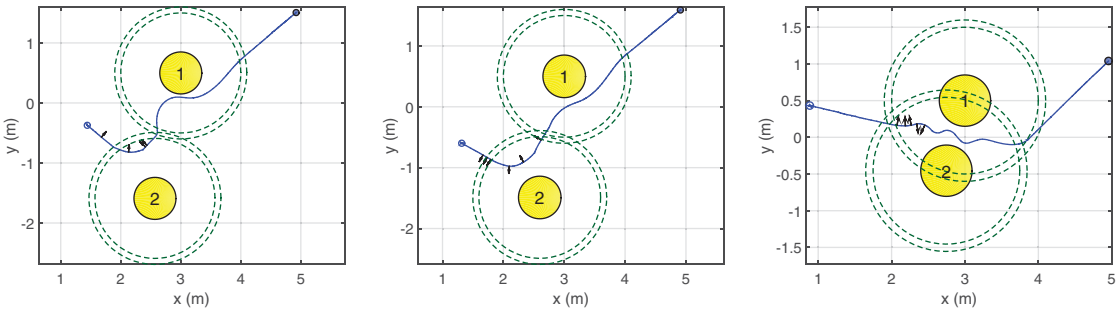
https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press



(a) $k_{a2} = 1.0$ (b) $k_{a2} = 1.2$ (c) $k_{a2} = 1.4$

(d) $k_{a2} = 1.6$ (e) $k_{a2} = 1.8$ (f) $k_{a2} = 2.0$

Fig. 15. Experimental results showing the robot trajectories when the human input is adversarial to the goal of obstacle avoidance. The parameter k_{a2} is varied while $\gamma = 1, \beta = 1, k_{a1} = 0.7$. (a) $k_{a2} = 1.0$. (b) $k_{a2} = 1.2$. (c) $k_{a2} = 1.4$. (d) $k_{a2} = 1.6$. (e) $k_{a2} = 1.8$. (f) $k_{a2} = 2.0$.



(a) $\mathcal{M}_{h1} \cup \mathcal{M}_{h2} \neq \emptyset$ (b) $\mathcal{M}_{h1} \cup \mathcal{M}_{d2} \neq \emptyset$ (c) $\mathcal{M}_{d1} \cup \mathcal{M}_{d2} \neq \emptyset$

Fig. 16. Experimental results showing the robot trajectories between two obstacles. (a) $\mathcal{M}_{h1} \cup \mathcal{M}_{h2} \neq \emptyset$. (b) $\mathcal{M}_{h1} \cup \mathcal{M}_{d2} \neq \emptyset$. (c) $\mathcal{M}_{d1} \cup \mathcal{M}_{d2} \neq \emptyset$.

Acknowledgments

The authors thank Dr Kevin Brink and other members of the AFRL sponsored Autonomous Vehicles Laboratory at UF REEF for making the facility and resources available for us to conduct experiments, and Drs Emily Doucette and Jess Willard Curtis for their continued support of our research endeavors. This research is supported in part by the contracts with the AFRL Mathematical Modeling and Optimization Institute #FA8651-08-D-0108/049-050 and a grant from the USDA NIFA AFRI National Robotics Initiative #2013-67021-21074. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring agency.

References

1. T. B. Sheridan, "Telerobotics," *Automatica* **25**(4), 487–507 (1989).
2. T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control* (MIT Press, Cambridge, MA, USA, 1992).
3. X. C. Ding, M. Powers, M. Egerstedt, S. Yih Young and T. Balch, "Executive decision support," *IEEE Robot. Autom. Mag.* **16**(2), 73–81 (2009).
4. S. Nudehi, R. Mukherjee and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Trans. Control Syst. Technol.* **13**(4), 588–592 (2005).

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press

5. J. Kim, H. Ladjal, D. Folio, A. Ferreira and J. Kim, "Evaluation of telerobotic shared control strategy for efficient single-cell manipulation," *IEEE Trans. Autom. Sci. Eng.* **9**(2), 402–406 (2012).
6. S. Lee and H. S. Lee, "Modeling, design, and evaluation of advanced teleoperator control systems with short time delay," *IEEE Trans. Robot. Autom.* **9**(5), 607–623 (1993).
7. J. Kofman, X. Wu, T. Luu and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Trans. Ind. Electron.* **52**(5), 1206–1219 (2005).
8. W. Griffin, W. Provancher and M. Cutkosky, "Feedback strategies for telemanipulation with shared control of object handling forces," *Presence* **14**(6), 720–731 (2005).
9. A. Franchi, C. Secchi, M. Ryll, H. Bulthoff and P. Giordano, "Shared control : Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robot. Autom. Mag.* **19**(3), 57–68 (2012).
10. P. Aigner and B. McCarragher, "Shared control framework applied to a robotic aid for the blind," *IEEE Control Syst.* **19**(2), 40–46 (1999).
11. P. Aigner and B. J. McCarragher, "Modeling and constraining human interactions in shared control utilizing a discrete event framework," *IEEE Trans. Man Cybern. Part A: Syst. Humans* **30**(3), 369–379 (2000).
12. R. Cooper, T. Corfman, S. Fitzgerald, M. Boninger, D. Spaeth, W. Ammer and J. Arva, "Performance assessment of a pushrim-activated power-assisted wheelchair control system," *IEEE Trans. Control Syst. Technol.* **10**(1), 121–126 (2002).
13. C. Cipriani, F. Zaccone, S. Micera and M. Carrozza, "On the shared control of an EMG-controlled prosthetic hand: Analysis of user-prosthesis interaction," *IEEE Trans. Robot.* **24**(1), 170–184 (2008).
14. H. Yu, M. Spenko and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *Auton. Robots* **15**(1), 53–66 (2003).
15. C. Urdiales, J. M. Peula, M. Fdez-Carmona, C. Barrué, E. J. Pérez, I. Sánchez-Tato, J. C. Toro, F. Galluppi, U. Cortés, R. Annichiaricco, C. Caltagirone and F. Sandoval, "A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation," *Auton. Robots* **30**(2), 179–197 (2010).
16. H. Wang and X. Liu, "Adaptive shared control for a novel mobile assistive robot," *IEEE/ASME Trans. Mechatronics* **19**(6), 1725–1736 (2014).
17. H. Kim, J. Biggs, D. W. Schloerb, J. Carmena, M. Lebedev, M. Nicolelis and M. Srinivasan, "Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces," *IEEE Trans. Biomed. Eng.* **53**(6), 1164–1173 (2006).
18. J. Borenstein and Y. Koren, "Teleautonomous guidance for mobile robots," *IEEE Trans. Syst. Man Cybern.* **20**(6), 1437–1443 (1990).
19. S. Venkataraman and S. Hayati, "Shared/traded control of telerobots under time delay," *Comput. Electr. Eng.* **19**(6), 481–494 (1993).
20. G. Hirzinger, B. Brunner, J. Dietrich and J. Heindl, "Sensor-based space robotics-rotex and its telerobotic features," *IEEE Trans. Robot. Autom.* **9**(5), 649–663 (1993).
21. T. W. Fong, C. Thorpe and C. Baur, "A Safeguarded Teleoperation Controller," *Proceedings of the IEEE International Conference on Advanced Robotics*, Budapest, Hungary (2001).
22. D. Bruemmer, D. Few, R. Boring, J. Marble, M. Walton and C. Nielsen, "Shared understanding for collaborative control," *IEEE Trans. Man Cybern. Part A: Syst. Humans* **35**(4), 494–504 (2005).
23. B. Khademian and K. Hashtrudi-Zaad, "Performance Issues in Collaborative Haptic Training," *Proceedings of the IEEE International Conference on Robotics and Automation* (2007) pp. 3257–3262.
24. B. Khademian and K. Hashtrudi-Zaad, "Shared control architectures for haptic training: Performance and coupled stability analysis," *Int. J. Robot. Res.* **30**(13), 1627–1642 (2011).
25. B. Khademian, J. Apkarian and K. Hashtrudi-Zaad, "Assessment of environmental effects on collaborative haptic guidance," *Presence* **20**(3), 191–206 (2011).
26. B. Khademian and K. Hashtrudi-Zaad, "Dual-user teleoperation systems: New multilateral shared control architecture and kinesthetic performance measures," *IEEE/ASME Trans. Mechatronics* **17**(5), 895–906 (2012).
27. D. Powell and M. OḾalley, "The task-dependent efficacy of shared control haptic guidance paradigms," *IEEE Trans. Haptics* **5**(3), 208–219 (2012).
28. K. T. Song, S. Y. Jiang and M. H. Lin. "Interactive teleoperation of a mobile manipulator using a shared-control approach," *IEEE Trans. Human-Mach. Syst.* **46**(6), 834–845 (2016).
29. A. Hansson and M. Servin, "Semi-autonomous shared control of large-scale manipulator arms," *Control Eng. Pract.* **18**(9), 1069–1076 (2010).
30. A. Poncela, C. Urdiales, E. Perez and F. Sandoval, "A new efficiency weighted strategy for continuous human/robot cooperation in navigation," *IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans* **39**(3), 486–500 (2009).
31. W. Ren and R. Beard, "Satisficing Approach to Human-in-the-Loop Safeguarded Control," *Proceedings of the American Control Conference*, Portland, OR, USA (2005) pp. 4985–4990.
32. J. Jiang and A. Astolfi, "Shared-Control for the Kinematic Model of a Mobile Robot," *Proceedings of the IEEE 53rd Annual Conference on Decision and Control* , Los Angeles, CA, USA (2014) pp. 62–67.
33. J. Jiang, P. Di Franco and A. Astolfi. "Shared control for the kinematic and dynamic models of a mobile robot," *IEEE Trans. Control Syst. Technol.* **24**(6), 2112–2124 (2016).
34. J. Jiang and A. Astolfi, "State and output-feedback shared-control for a class of linear constrained systems," *IEEE Trans. Autom. Control* **61**(10), 3209–3214 (2016).

35. S. M. Erlien, S. Fujita and J. C. Gerdes. "Shared steering control using safe envelopes for obstacle avoidance and vehicle stability," *IEEE Trans. Intell. Transp. Syst.* **17**(2), 441–451 (2016).
36. J. G. Storms and D. M. Tilbury, "Blending of Human and Obstacle Avoidance Control for a High Speed Mobile Robot," *Proceedings of the American Control Conference*, Portland, OR, USA (2014) pp. 3488–3493.
37. S. Y. Jiang, C. Y. Lin, K. T. Huang and K. T. Song, "Shared control design of a walking-assistant robot," *IEEE Trans. Control Syst. Technol.* **25**(6), 2143–2150 (2017).
38. S. Hayati and S. Venkataraman, "Design and Implementation of a Robot Control System with Traded and Shared Control Capability," *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, USA (1989) pp. 1310–1315.
39. S. Anderson, J. Walker and K. Iagnemma, "Experimental performance analysis of a homotopy-based shared autonomy framework," *IEEE Trans. Human-Mach. Syst.* **44**(2), 190–199 (2014).
40. J. Guldner and V. Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," *IEEE Trans. Robot. Autom.* **11**(2), 247–254 (1995).
41. Z. Wu, G. Hu, L. Feng, J. Wu and S. Liu, "Collision avoidance for mobile robots based on artificial potential field and obstacle envelope modelling," *Assembly Autom.* **36**(3), 318–332 (2016).
42. Z. Wang, J. Lam and K. J. Burnham, "Stability analysis and observer design for neutral delay systems," *IEEE Trans. Autom. Control* **47**(3), 478–483 (2002).
43. H. Hashimoto, F. Harashima, V. Utkin, S. Krasnova and I. Kaliko, "Sliding Mode Control and Potential Fields in Obstacle Avoidance," *Proceedings of European Control Conference*, Groningen, Netherlands (1993) pp. 859–862.

https://doi.org/10.1017/S0263574717000510 Published online by Cambridge University Press