A Shared-Autonomy Approach to Goal Detection and Navigation Control of Mobile Collaborative Robots

Soheil Gholami[1,2], Virginia Ruiz Garate[1], Elena De Momi[2], and Arash Ajoudani[1]

***Abstract*— Autonomous goal detection and navigation control of mobile robots in remote environments can help to unload human operators from simple, monotonous tasks allowing them to focus on more cognitively stimulating actions. This can result in better task performances, while creating user-interfaces that are understandable by non-experts. However, full autonomy in unpredictable and dynamically changing environments is still far from becoming a reality. Thus, teleoperated systems integrating the supervisory role and instantaneous decision-making capacity of humans are still required for fast and reliable robotic operations. This work presents a novel shared-autonomy framework for goal detection and navigation control of mobile manipulators. The controller exploits human-gaze information to estimate the desired goal. This is used together with control-pad data to predict user intention, and to activate the autonomous control for executing a target task. Using the control-pad device, a user can react to unexpected disturbances and halt the autonomous mode at any time. By releasing the control-pad device (e.g., after avoiding an instantaneous obstacle) the controller smoothly switches back to the autonomous mode and navigates the robot towards the target. Experiments for reaching a target goal in the presence of unknown obstacles are carried out to evaluate the performance of the proposed shared-autonomy framework over seven subjects. The results prove the accuracy, time-efficiency, and ease-of-use of the presented shared-autonomy control framework.**

I. INTRODUCTION

In the mid-1940s, Ray Goertz introduced a remotely controlled manipulator [1] that led to one of the most promising fields in robotics, called teleoperation. Since then, several application domains have benefited from this concept such as medical surgery [2], space exploration [3], and disaster response [4]. The stability and transparency of such teleoperation systems [5] have undergone significant progress in an attempt to ensure safety and interaction efficiency. However, the intuitiveness-of-use of the teleoperation interfaces, another equally important aspect, has received less attention [6]. This is of particular importance when users must operate multi-degrees-of-freedom follower systems that are embedded with dexterous loco-manipulation (simultaneous locomotion and manipulation) potential such as mobile manipulators or humanoids [7]. In fact, in most of the related developed frameworks (see e.g., [8]), a user, while watching a graphical user interface (GUI), constantly generates the desired motions for the follower robot to accomplish the desired task. Nevertheless, full teleoperation of such devices may be perceived as annoying during prolonged or repetitive tasks, e.g., for long-distance target reaching operations.

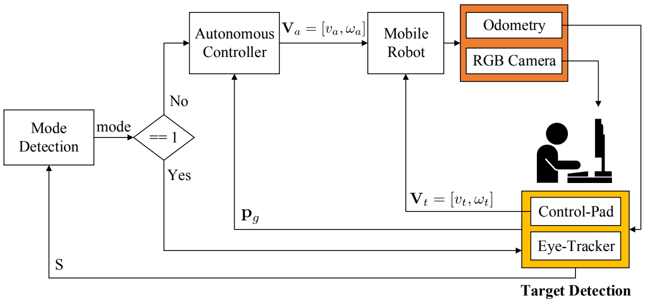

Fig. 1. Block diagram of the proposed framework, which consolidates the advantages of both motion control interface and eye-gaze tracking system. The mode detection block decides which mode (teleoperation or autonomous) should be active at each moment based on the signal S received from the target detection block. As a result, the proper twist commands are sent to the robot, allowing the user to change mode at any time.

A practical solution could be the use of fully autonomous systems that can navigate through selected indoor [9] and outdoor [10] coordinates. However, these systems lack the flexibility, reactivity, and supervisory skills of human operators, which can be crucial, e.g., when avoiding unexpected obstacles. Therefore, the combination of autonomous and corrective control inputs with human monitory commands arises as an effective solution to improve the user experience during the performance of remote robotic tasks. These combined systems can be categorized as shared-autonomy teleoperation [11], [12], [13].

A key factor in the shared-autonomy control is the choice of the appropriate human input sensory data to be used for the intention detection. In [14], the authors utilized hand-sketches on a screen to generate a desired path for a mobile robot to follow and avoid the obstacles. The user has to look at the screen and sketch the path simultaneously. This required continuous attention and conscious effort from the user. To limit this effort, other approaches propose to use more intuitive data such as gaze tracking [15].

Gaze estimation poses several challenges [16], eye-tracker inaccuracies, rapid eye movements, eye-blinking, etc. Hence, while this method is not encouraged to be used independently, it is best suited to be employed together with a motion-intention interface. In [15], the authors developed a shared-autonomy system, in which pick and place tasks were remotely performed by utilizing gaze and joystick data. Moreover, a predictive system based on the gaze information and the body motion sensors is proposed in [17] and [18], which improves the maneuverability in the leader-follower

This work was supported by the European Research Council (ERC) starting grant Ergo-Lean, Grant Agreement No. 850932

[1] Human-Robot Interfaces and physical Interaction (HRI^2) Lab, Istituto Italiano di Tecnologia, Genoa, Italy, Email: Soheil.Gholami@iit.it

[2] NearLab, Dept. of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy

systems in performing the reach and grasp tasks for a fixed-base manipulator. Although these solutions have made the remote grasping tasks more intuitive and fast, the proposed frameworks cannot be applied to mobile robot navigation due to the different nature of the motions.

Consequently, the objective of this work is to introduce a novel shared-autonomy framework for remote navigation control of mobile platforms. This study proposes a hybrid interface that merges information from a control-pad device and human eye-gaze to calculate the intended reaching goal by the mobile robot. Once a goal is detected, the robot's autonomous navigation controller brings the platform to the target. Still, during this autonomous motion, the framework keeps a human-in-control approach, in which the user is given full control over the robot's motion through the control-pad device. Our proposed method unloads users from the low-level control of the robot, allowing them to focus on the more high-level decision-making commands. For example, in a logistics environment, the user would be able to command the final location of the robot without needing to manage its motion during the full trajectory.

The performance and intuitiveness-of-use of the shared-autonomy navigation interface are evaluated on 7 users and compared to the full teleoperation mode where no autonomous goal detection and navigation are exploited.

II. METHODOLOGY

The schematic of the proposed framework is displayed in Fig. 1, which incorporates the advantages of motion control interface and eye-gaze tracking system. The mode detection block decides which mode (teleoperation or autonomous) should be active at each moment based on the signal S received from the target detection block. This block includes the gaze estimation (Section II-A) and control-pad interface.

As an overview, when a target is established by the gaze tracker system, if the user starts to navigate the robot towards this target, a message is shown in the GUI's window which indicates the release of the control-pad device. By doing so, the mode detection resets the corresponding mode flag and the autonomous controller sends the required twist commands $\mathbf{V}_a = [v_a,\, \omega_a]$ to the mobile robot. (the twist command $\mathbf{V}$ is composed of linear velocity v and angular velocity ω, see Fig. 2.) If the operator starts using the control-pad device, both at the beginning or in the middle of the autonomous control, the mode detection block triggers the mode flag. This activates the teleoperation mode, in which the mobile robot is controlled by the user through desired twist commands $\mathbf{V}_t = [v_t,\, \omega_t]$. Teleoperation is most effective within the proposed method in case of failures due to sudden and unpredictable environment changes, giving the user full control over the robot's motion. User continuously receives visual feedback from an RGB camera sensor.

The different components of the shared-autonomy framework are explained in detail hereafter.

A. Target detection

To detect the desired target for autonomous navigation, the gaze tracker information is enhanced with the inputs coming from the control-pad and the actual robot's heading angle, being the latter updated by the odometry sensors.

Starting with the gaze tracking system, this measures the gaze coordinates in pixels with respect to (w.r.t.) the top-left corner of the computer monitor, i.e., frame $\{T\}$ (Fig. 2-a), and outputs $\mathbf{p}_T = [u_T,\, v_T]^T$ where u_T and v_T are horizontal and vertical pixel coordinates of $\{T\}$, respectively. Since the origin of the GUI's coordinates is considered to be located at frame $\{S\}$ (Fig. 2-a), $\mathbf{p}_T$ needs to be first transformed to this frame by making use of a map-function, $\mathbf{f}_{TS}(\cdot) : \mathbf{R}^2 \to \mathbf{R}^2$ being the result defined as $\mathbf{p}_S = [u_S,\, v_S]^T$:

$$u_S = \frac{W_{gui}}{2} - u_T, \quad v_S = \frac{H_{gui}}{2} - v_T. \tag{1}$$

W_{gui} and H_{gui} are the pixel-based width and height of the GUI window, respectively. Next, another function $\mathbf{f}_{SC}(\cdot) : \mathbf{R}^2 \to \mathbf{R}^3$ is utilized to express $\mathbf{p}_S$ in camera coordinates $\{C\}$ (Fig. 2-b). In this work, a predefined discrete mapping is employed to convert pixel coordinates $\mathbf{p}_{S_i} = [u_i,\, v_i]^T$ to the 3D camera coordinates $\mathbf{p}_{C_i} = [x_i,\, y_i,\, z_i]^T$. Thus, a set of specific coordinates (with n points) in pixel space ($\mathbf{S}_{uv} = \{\mathbf{p}_{S_i} \in \mathbf{R}^2 \mid i \in \{1, \cdots, n\}\}$) and its representative set of points in the camera coordinates in the Cartesian space ($\mathbf{S}_{xyz} = \{\mathbf{p}_{C_i} \in \mathbf{R}^3 \mid i \in \{1, \cdots, n\}\}$) are considered to be known beforehand. $\mathbf{p}_C$ is expressed in the robot's base frame, $\{R\}$ (Fig. 2-b), by employing the constant transformation matrix between camera and robot's base frame as $\left[\mathbf{p}_R^T,\, 1\right]^T = {}^{\mathrm{R}}\mathbf{T}_{\mathrm{C}}\left[\mathbf{p}_C^T,\, 1\right]^T$.

Based on the nearest neighbor search (NNS) algorithm [19], at each iteration, the measured gaze information in $\{S\}$, $\mathbf{p}_S$, is compared with the stored data in $\mathbf{S}_{uv}$ during gaze fixation time T_{fix} (time-period in which the eye is kept aligned around a certain point located at the screen, which is dependent on the person who carries out the experiment, and it usually ranges in 2 to 4 seconds [20]). If the shortest Euclidean distance between $\mathbf{p}_S$ and a particular $\mathbf{s}_{uvi} \in \mathbf{S}_{uv}$ is lower than a selected threshold (d_{tsh}) during T_{fix}, $\mathbf{s}_{uvi}$ is chosen as the intended pixel-space goal. In the camera's coordinates, $\mathbf{s}_{uvi}$ is expressed as $\mathbf{p}_{C_i} \in \mathbf{S}_{xyz}$, which is then converted to $\mathbf{p}_{R_i}$ by employing ${}^{\mathrm{R}}\mathbf{T}_{\mathrm{C}}$. Thus, the fixed-gaze direction is obtained by: $\phi_{fix} = tan^{-1}\left(\mathbf{p}_{R_iy}, \mathbf{p}_{R_ix}\right)$. This direction is updated based on the user will, who is allowed to restart the operation to modify the estimated location.

For calculating the intended direction at instance k, first, ϕ_{fix} is updated due to the robot's movements during the time interval $t_0 - t_1$, by making use of the robot's odometry data ($x - y$ position of the robot) and the vectors addition property:

$$\phi^{\star}_{fix}(k) = tan^{-1}\left(\mathbf{p}_{R_iy} - y(k),\, \mathbf{p}_{R_ix} - x(k)\right) \tag{2}$$

Second, the average value of the heading angle, measured by the odometry sensors, over the past N samples is calculated:

$$\bar{\theta}(k) = \frac{1}{N}\sum_{i=k-N-1}^{k}\theta_i. \tag{3}$$

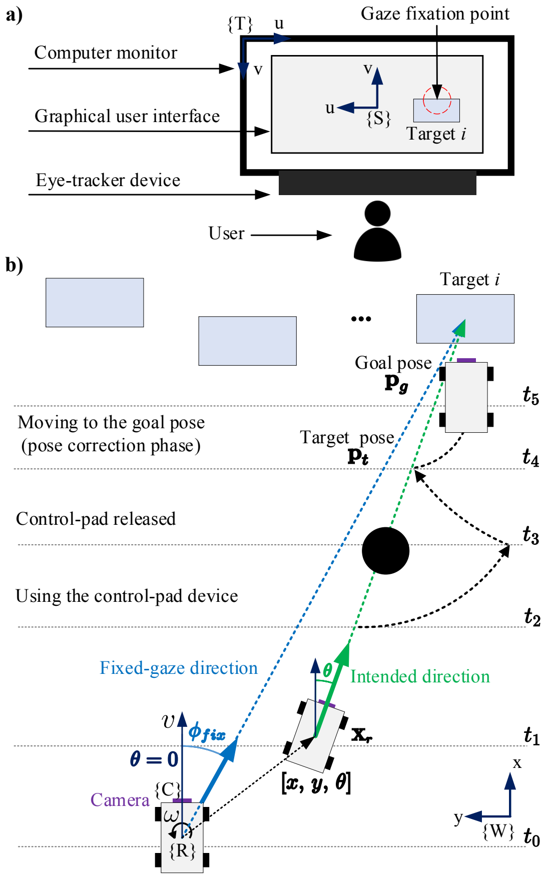

Fig. 2. (a) The user observes the environment through a screen, and controls the robot through the gaze tracker and the control-pad device. (b) Switching between teleoperation and autonomous modes in the proposed framework.

If $|\bar{\theta}(k) - \phi^{\star}_{fix}| < \Psi_{tsh}$, the user is informed through the GUI to release the control-pad to activate the autonomous controller setting $\theta(k)$ as the intended direction.

B. Autonomous controller

The autonomous controller designed based on the bicycle model for differential-drive mobile robots [21]. According to this model, the platform motion is described as:

$$\dot{x} = v\cos(\theta), \quad \dot{y} = v\sin(\theta), \quad \dot{\theta} = \omega. \tag{4}$$

v is the linear velocity and ω is the angular velocity. $\mathbf{X} = [x,\, y]^T$ and θ are the current position and orientation of the platform, respectively, being the current pose of the platform defined as $\mathbf{p} = [\mathbf{X},\, \theta]^T$.

The controller is used during two phases: path-following and pose-correction. The former is in charge of navigating the robot on the desired generated path (starts in t_1 and ends in the target pose at t_4) while the latter contributes to the realization of the robot's goal pose. This pose is considered to be located at a distance of d_{safe} before the real goal for safety reasons and avoiding collisions with it ($\mathbf{p}_g = [x_g,\, y_g,\, \theta_g]^T$).

To assign desired initial and final values for position, velocity, and acceleration, a 5-th order trajectory ($\mathbf{X}_d = [x_d,\, y_d]^T$) is generated for the path-following phase, i.e., from the instance the user releases the control-pad to the moment the robot reaches its target position (time interval $t_1 - t_4$ in Fig. 2-b). The robot's position in the release and target moments are defined as $\mathbf{X}_r = [x_r,\, y_r]^T$ and $\mathbf{X}_t = [x_t,\, y_t]^T$, respectively. As a result, $\mathbf{X}_d$ is given by:

$$\mathbf{X}_d(\tau) = \mathbf{X}_r + \Delta\mathbf{X}\left(6\tau^5 - 15\tau^4 + 10\tau^3\right). \tag{5}$$

$\Delta\mathbf{X} = \mathbf{X}_t - \mathbf{X}_r$, $\dot{\mathbf{X}}_r = \dot{\mathbf{X}}_t = 0$, $\ddot{\mathbf{X}}_r = \ddot{\mathbf{X}}_t = 0$, and $\tau = t/T$ with a total trajectory time T (time interval t_1 to t_4 in Fig. 2-b). T is derived from $T = d/v^{\star}$, where $d = \|\mathbf{X}_t - \mathbf{X}_r\|$ and $v^{\star}$ is chosen based on the maximum linear velocity v_{max} allowed for the mobile platform ($v^{\star} = \alpha\, v_{max}$, being α the percentage of the maximum velocity). After releasing the device, the robot autonomously navigates, on a straight line, towards the target pose following the next control laws:

$$v = k_p\, e + k_i \int e\, dt + k_d\, \dot{e}, \quad \omega = k_h\, (\theta^{\star} - \theta). \tag{6}$$

$k_p,\, k_i,\, k_d,\, k_h > 0$, $d^{\star}$ is a distance behind the pursuit point, and e and $\theta^{\star}$ are defined as [21]:

$$\begin{aligned} e &= \sqrt{(x_d(t) - x(t))^2 + (y_d(t) - y(t))^2} - d^{\star}, \\ \theta^{\star} &= tan^{-1}([y_d(t) - y(t)]/[x_d(t) - x(t)]). \end{aligned} \tag{7}$$

During this phase, the user-in-control approach allows the operator to take full control over the platform. There are several cases where this can be needed, e.g., when obstacles are not detectable (glass objects), change of task goal, and sensor failures. When the user takes control of the robot by using the control-pad, time t_2 in Fig. 2-b, teleoperation mode is activated triggering the mode variable, and consequently, the autonomous navigation is put on hold. After bypassing the obstacle at time t_3, the user activates the autonomous mode simply by releasing the control-pad. Consequently, the robot resumes the autonomous navigation generating a path (5) towards the previously detected target pose ($\mathbf{p}_t$). The path-following controller (6) then navigates the robot and it reaches the goal at time t_4 (Fig. 2-b).

After reaching the target pose, the move-to-pose controller brings the robot to the goal pose $\mathbf{p}_g$ at time t_5 by employing the following control rules [21]:

$$\begin{aligned} &v = k_{\rho}\, \rho, \;\; \omega = k_{\alpha}\, \alpha - k_{\beta}\, \beta, \;\; \rho = \sqrt{(x_g - x)^2 + (y_g - y)^2}, \\ &\alpha = tan^{-1}([y_g - y]/[x_g - x]), \quad \beta = -\theta - \alpha + \theta_g. \end{aligned} \tag{8}$$

$k_{\rho} > 0$, $k_{\beta} < 0$, and $k_{\alpha} > k_{\rho}$.

III. EXPERIMENTS

The performance of the proposed framework is evaluated in a navigation task in two scenarios, and compared to the full teleoperation mode. The software architecture is implemented based on the Robot Operating System (ROS) and MATLAB software. In the first set of experiments, i.e. scenario 1, the aim is to analyze the robot motion and the subjects' efforts in an obstacle-free environment in two modes: shared-autonomy and teleoperation. In scenario 2, we repeat the scenario 1 but in the presence of an obstacle in the environment and testing more subjects. It should be noted that in this case, we use a fixed positioned obstacle to allow a better comparison between users. Still, the same method is applicable to moving and unpredictable obstacles.

A. *Experimental Setup*

The experimental set-up is shown in Fig. 3. The mobile platform used in our experiments (MOCA [8]) is a recent integration of an Omni-directional base (SUMMIT XL STEEL from Robotnik), a robotic arm (Franka Emika Panda) and an under-actuated hand (Pisa/IIT soft hand [22]). As this work mainly focuses on the remote navigation control of the mobile platform towards a target goal, the arm and the hand are controlled to keep a fixed configuration.

During the experiments, the user sits in front of a computer monitor with a dual-shock SONY PS4 control-pad. Visual-data is fed-back from the platform's camera through the provided MATLAB GUI in Microsoft Windows 10 operating system (Fig. 3-a). In the shared-autonomy mode the task's current state is also shown to the user. The eye-tracker application is based on Myex, which is specifically developed for Tobii EyeX system [23] to estimate gaze location. The resolution of the computer monitor, for which the eye-tracker is calibrated (Tobii eye-tracking core software), and the GUI window are 1680×1050 pixels and 640×480 pixels, respectively. (the robot's RGB camera resolution is also 640×480 pixels)

The goal is set as one object located on a desk (Fig. 3-d). Besides, several boxes are placed to enhance the perception of the final safe pose while the robot approaches the desk (Fig. 3-c). These boxes are never used for the control of the platform, but only for the mentioned user perception. For the second scenario, an obstacle is additionally located along the way of the platform (Fig. 3-c).

The first scenario is performed by one person who is not familiar with the shared-autonomy concept (male, 27 years old, and without eye-glass). He executes five times the task in each mode. For the second scenario, 6 subjects are selected to perform the navigation task in each mode while avoiding the obstacle. Subjects are of different ages (24 to 37), gender (4 males and 2 females), and 5 with glasses and 1 without, and have little knowledge about the shared-autonomy navigation but are familiar with teleoperation.

During the teleoperation mode in both scenarios, subjects are asked to navigate the robot from the start pose (Fig. 3-b) to the goal pose (Fig. 3-d) while watching the camera data on the screen. In the shared-autonomy mode, the subjects are asked to perform the same task with the developed shared-autonomy framework. Performance of the subjects during the experiments will be evaluated in terms of task completion time, and position and orientation errors of the final platform's pose. Moreover, the subjective perception will also be considered through a Likert scale subjective questionnaire [24].

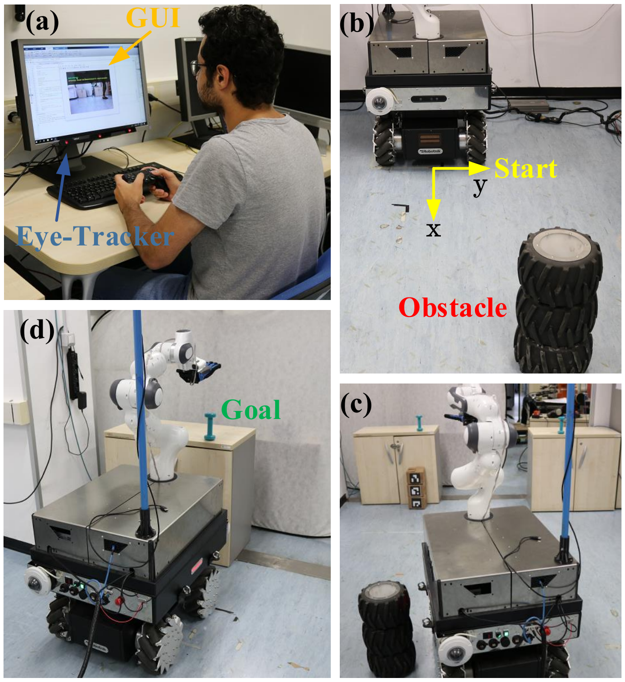

Fig. 3. The experimental setup: a) the user sits in front of a screen which is equipped with an eye-tracking device to measure gaze information. He is also provided with a control-pad device. b) The start pose of the platform and the location of the obstacle. c) The robot deviates from the original path to avoid the obstacle. d) The robot reaches its final goal pose.

The controller parameters during the shared-autonomy mode are selected as: $k_p = 1.25$, $k_i = 0.05$, $k_d = 0.001$, $k_h = 2.5$, $k_\alpha = 15$, $k_\beta = -4.5$, $k_\rho = 5$, $|v^\star| = 0.165$ m/sec, $|v_{max}| = 0.20$ m/sec, $|w_{max}| = 0.175$ rad/sec, $d_{tsh} = 50$ pixels, $d_{safe} = 0.40$ m, $T_{fix} = 3$ sec [20], $N = 30$, and $\Psi_{tsh} = 3$ deg. The maximum linear and angular velocities and controllers' gains are set to allow the subjects to react to possible obstacles while being fast enough not to render the experiments tedious. The values of k_h, k_α, k_β, and k_ρ are chosen to satisfy the conditions of Eq. 8. The threshold values are chosen empirically based on the previous experiments and analysis of the processed data. In addition, W_{GUI} and H_{GUI} are 640 and 480 pixels, respectively. Also, the transformation matrix ${}^{\mathrm{R}}\mathbf{T}_{\mathrm{C}}$ is obtained by using the tf conversion library provided by the mobile robot's ROS software package[1]. Moreover, the goal position and orientation in the camera coordinates are $\mathbf{X}_g = [3.1, -0.72]^T$ m and $\theta_g = 0$ deg, respectively, while the corresponding goal location in the pixel space is $[u_g, v_g]^T = [220, 200]^T$ pixel.

B. *Experimental Evaluation*

Quantitative and subjective analyses are carried out for both scenarios. Quantitatively, the interpretation of quantile-quantile plots [25] yields that the data does not follow a normal distribution in both modes and scenarios. Thus, the following parameters are studied: median (M) and interquartile range (IQR) of the position error $e_p = \|\mathbf{X}_g - \mathbf{X}_f\|$, with $\mathbf{X}_f$ being the final position of the platform; and the orientation error $e_\theta = |\theta_g - \theta_f|$, with θ_f being the platform's final heading angle. The final pose of the platform $\mathbf{p}_f = [\mathbf{X}_f, \theta_f]^T$ is given by the odometry sensors when the task is

[1] www.github.com/RobotnikAutomation/

TABLE I

MEDIAN (M) AND INTER-QUARTILE RANGE (IQR) OF THE POSITION AND ORIENTATION ERRORS. THE ACRONYMS PAR, TELEOP, AND SHARED STAND FOR PARAMETERS, TELEOPERATION MODE, AND SHARED-AUTONOMY MODE, RESPECTIVELY.

		Scenario 1		*Scenario 2*	
variable	**par**	**teleop**	**shared**	**teleop**	**shared**
position	M	0.1568	0.0040	0.2860	0.0275
error [m]	IQR	0.0372	0.0045	0.3484	0.0159
orientation	M	5.6701	3.8221	10.5721	3.8988
error [deg]	IQR	5.6548	0.1477	8.0674	0.0381

finished. These quantitative results are presented in Table I. Also, the task completion time is recorded in each trial and the average values, related to each scenario and mode, are then calculated. It should be noted that the gaze fixation time is also included in the shared-autonomy task achievement.

Additionally, statistical tests are invoked to compare both modes in the sense of position and orientation errors. Due to the small amount of subjects available during the experiments, it is not possible to prove a particular distribution type of the data. Therefore, non-parametric tests should be considered. Also, in each scenario, the experimental conditions are kept fixed in both modes, and the same person is compared when executing the trials in mode 1 and mode 2, being the experiments defined as paired. Based on these experimental characteristics, the Wilcoxon signed-rank test [26] is employed for the statistical evaluation with the significance level α set to 0.05.

The subjective analysis is done through a Likert scale subjective questionnaire [24] during scenario 2. The answers are mapped to numerical grades between -1 (strongly disagree) to +1 (strongly agree). The following statements are asked in the survey: (Q1) It was physically tiresome to accomplish the task, (Q2) It was psychologically tiresome to accomplish the task, (Q3) The GUI accurately detected the intended location in around 3 seconds, (Q4) I had good control over the task performance (controlling the robot and obstacle avoidance), (Q5) It was not intuitive to activate/deactivate the autonomous motion controllers, (Q6) It was not intuitive to understand the current action to perform (GUI), (Q7) It was easy to keep the focus on the task execution (robot motions), (Q8) I think by using the proposed framework I can repeat the same task for a longer time and in a better way (more precisely and faster) than in pure teleoperation, (Q9) I felt satisfied with the current task. Statements [3-5-6-8] are just applicable to the shared-autonomy control structure.

C. Experimental Results and Discussion

Table I shows the results of the median and IQR values of the position and orientation errors, being these greater in teleoperation than in the shared-autonomy mode in both scenarios. Moreover, the presence of an obstacle in the second scenario increases the final position error in both modes. This is more notable in the teleoperation mode, which may be due to the loss of perception of the users of the relative orientation between the platform and the goal after the obstacle is passed.

As illustrated in Fig. 4, during scenario 1 in teleoperation mode, the robot deviates from the optimal path in all trials. Indeed, the robot moves more accurately towards the goal in the shared-autonomy mode (less position and orientation error, Table I). Moreover, the average time that the user commands the robot with the control-pad is, as expected, higher for the teleoperation (28.80 sec) than for shared-autonomy (27.80 sec) mode.

Fig. 5 shows the robot's odometry data after adding an obstacle to the environment in scenario 2 with the 6 subjects in both modes. In the presence of an obstacle, the subjects tend to start deviating from the optimal path sooner, which may be related to the fear of collision. Additionally, in the teleoperation mode, the participants perform several forward-backward motions with the robot, which does not happen in the shared-autonomy mode. As shown in Table I, again this reflects in bigger final pose errors. During these experiments though, the average execution time with the shared-autonomy mode (47.67 sec) is greater than the teleoperation one (44.65 sec), which can be due to the need of change between modes during the obstacle avoidance. Moreover, it must be noted it takes about 3 ± 0.5 sec for goal detection through gaze and around 1.5 ± 1 sec to estimate the desired path by comparing the $\phi^{\star}_{fix}$ and $\bar{\theta}$. Moreover, gaze data during the goal detection, is shown in Fig. 6. It can be seen that overall, the data is concentrated around the goal. Present variations in the data are due to eye-blink, head motions, etc.

Finally, the p-values calculated by the Wilcoxon signed-rank test confirm that the shared-autonomy mode has statistically fewer position errors than the teleoperation mode both in scenario 1 (p = 0.0313) and scenario 2 (p = 0.0156). However, this test yields $p > 0.05$ for the orientation errors. The p-values for the orientation error tests of scenario 1 and scenario 2 are p = 0.0625 and p = 0.0781, respectively. A possible reason of the p-values being greater than 0.05 can be the small sample size of the data, as the obtained p-values are actually close to the threshold. Specifically, from the individual data, we could observe how one of the subjects performed very differently from the rest in terms of orientation error in the second scenario. Due to the small size of our data set, this could have a big impact on the overall evaluation, yielding a $p > 0.05$. Future works envisage the collection larger data sets of subjects to minimize the influence of possible individual deviations. The use of more subjects will also have an impact on the average execution time, allowing a better comparison of both modes in completion time.

Regarding the subjective questionnaire, results are illustrated in Fig. 7. The participants show less physical and psychological effort for the navigation execution with the activated shared-autonomy algorithm than with the teleoperation mode (Q1-Q2). Instead, for the motion control and obstacle avoidance (Q4-Q7), differences are quite low and the satisfaction level of both modes is almost the same (Q9). Regarding the developed framework, the participants are generally satisfied with the GUI; stating in Q3 that it detects the intended target accurately and in about 3 sec. Plus, subjects have indicated no problems for understanding the

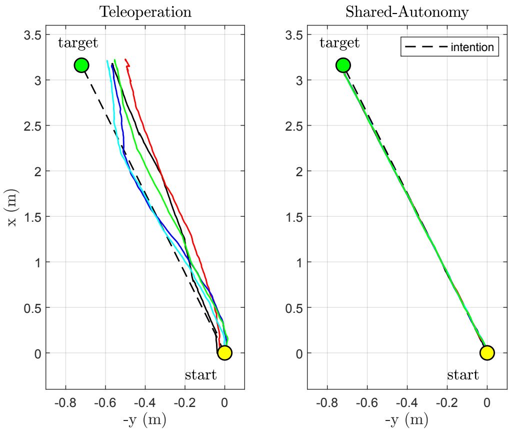

Fig. 4. Robot's position in teleoperation and shared-autonomy modes (scenario 1). Each color is assigned to one trial.

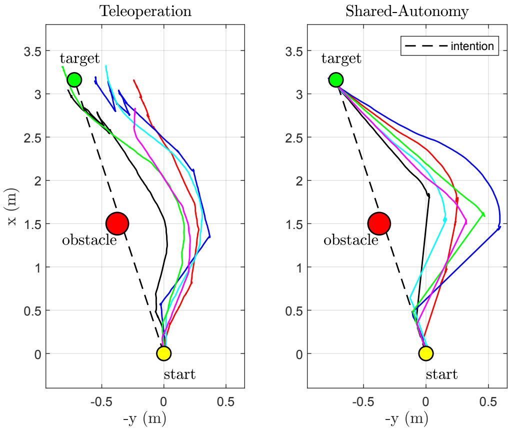

Fig. 5. Robot's position in teleoperation and shared-autonomy modes in the presence of an obstacle (scenario 2). Each color is assigned to one subject.

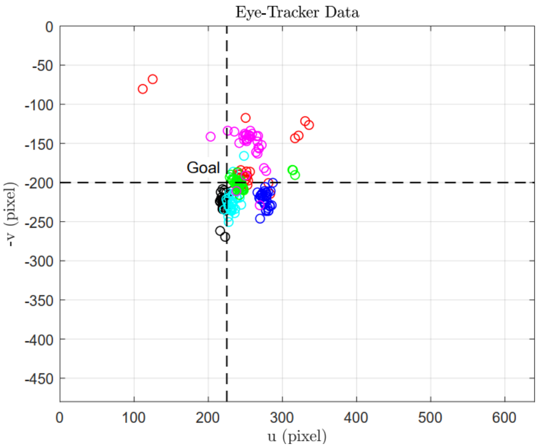

Fig. 6. Gaze coordinates (scenario 2). Gaze information is key for the detection of the goal position (the intersection point of both dashed lines).

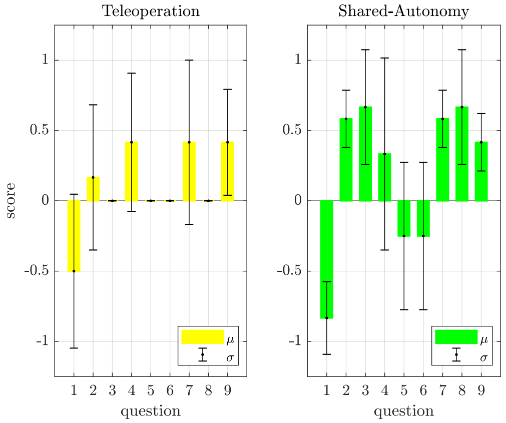

Fig. 7. Likert scale-based questionnaire scores for teleoperation and shared-autonomy tasks. The scores are: -1 (strongly disagree), -0.5 (disagree), 0 (neither agree nor disagree), 0.5 (agree), 1 (strongly agree).

current action to perform (Q6) and for activating/deactivating the autonomous motion controller (Q5). Finally, subjects show an agreement on the fact that they can repeat the same task for a longer time and in a more precise and faster way with the developed interface (Q8).

IV. CONCLUSIONS

In this work, we proposed a shared-autonomy framework for the navigation control of a mobile robot in remote environments. The developed control interface processes the data from an eye-tracking system and a control-pad device. From these data, the target goal is quickly identified activating the autonomous navigation controller. The users can go back to control robot movements at any time through the control-pad, e.g., when an obstacle is observed through the monitor. After releasing the control-pad device, the control framework smoothly switches back to the autonomous mode and navigated the mobile platform towards the previously identified goal.

Experiments with 7 subjects in pure teleoperation and shared-autonomy control modes revealed the accuracy, time-efficiency, and ease-of-use of the proposed shared-autonomy interface. The latter was evidenced by a reduction in the physical and cognitive loads thanks to the intelligent goal detection and robot navigation algorithms developed in this work.

Future works envisage the addition of an obstacle-avoidance protocol that will handle easy to detect obstacles, being the operator's cognitive skills only required in case of unexpected events or trajectory corrections.

REFERENCES

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

[2] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, "Medical robotics and computer-integrated surgery," in *Springer handbook of robotics*. Springer, 2016, pp. 1657–1684.

[3] D. Reintsema, K. Landzettel, and G. Hirzinger, "DLR's advanced telerobotic concepts and experiments for on-orbit servicing," in *Advances in telerobotics*. Springer, 2007, pp. 323–345.

[4] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V.-G. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino *et al.*, "Walk-man: A high-performance humanoid platform for realistic environments," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.

[5] T. B. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report," *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995.

[6] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," in *2018 IEEE-RAS 18th Int. Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[7] J. Borras and T. Asfour, "A whole-body pose taxonomy for loco-manipulation tasks," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1578–1585.

[8] Y. Wu, P. Balatti, M. Lorenzini, F. Zhao, W. Kim, and A. Ajoudani, "A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3593–3600, Oct 2019.

[9] H. Cheng, H. Chen, and Y. Liu, "Topological indoor localization navigation for autonomous industrial mobile manipulator," in *2012 11th International Conference on Machine Learning and Applications*, vol. 1, Dec 2012, pp. 238–243.

[10] M. J. Procopio, J. Mulligan, and G. Grudic, "Coping with imbalanced training data for improved terrain prediction in autonomous outdoor robot navigation," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 518–525.

[11] B. Brunner, G. Hirzinger, K. Landzettel, and J. Heindl, "Multisensory shared autonomy and tele-sensor-programming-key issues in the space robot technology experiment rotex," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 3. IEEE, 1993, pp. 2123–2139.

[12] S. Reddy, S. Levine, and A. D. Dragan, "Shared autonomy via deep reinforcement learning," *ArXiv*, vol. abs/1802.01744, 2018.

[13] A. D. Dragan, S. S. Srinivasa, and K. C. T. Lee, "Teleoperation with intelligent and customizable interfaces," *J. Hum.-Robot Interact.*, vol. 2, no. 2, pp. 33–57, Jun. 2013. [Online]. Available: https://doi.org/10.5898/JHRI.2.2.Dragan

[14] K. Lee, U. Mehmood, and J. Ryu, "Development of the human interactive autonomy for the shared teleoperation of mobile robots," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1524–1529.

[15] H. Admoni and S. S. Srinivasa, "Predicting user intent through eye gaze for shared autonomy," in *AAAI Fall Symposia*, 2016.

[16] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16 495–16 519, 2017.

[17] A. Namiki, Y. Matsumoto, T. Maruyama, and Y. Liu, "Vision-based predictive assist control on master-slave systems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5357–5362.

[18] A. Matsuzaka, L. Yang, C. Guo, T. Shirato, and A. Namiki, "Assistance for master-slave system for objects of various shapes by eye gaze tracking and motion prediction," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2018, pp. 1953–1958.

[19] I. Kalantari and G. McDonald, "A data structure and an algorithm for the nearest point problem," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 5, pp. 631–634, Sep. 1983.

[20] L. Fortini, P. Balatti, E. Lamon, E. De Momi, and A. Ajoudani, "A collaborative robotic approach to gaze-based upper-limb assisted reaching," in *IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, 07 2019.

[21] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms In MATLAB, Second Edition*, 2nd ed. Springer Publishing Company, Incorporated, 2017.

[22] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, M. Bonilla, M. Garabini, C. Piazza, M. Gabiccini, and A. Bicchi, "From soft to adaptive synergies: The pisa/iit softhand," in *Human and Robot Hands*. Springer, 2016, pp. 101–125.

[23] A. Gibaldi, M. L. Vanegas, P. J. Bex, and G. Maiello, "Evaluation of the tobii eyex eye tracking controller and matlab toolkit for research," in *Behavior research methods*, 2017.

[24] R. Likert, "A technique for the measurement of attitudes." *Archives of psychology*, 1932.

[25] Y. Dodge, *The concise encyclopedia of statistics*. Springer Science & Business Media, 2008.

[26] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945. [Online]. Available: http://www.jstor.org/stable/3001968