

Toward Human-Robot-Teaming for Robot Navigation Using Remote Control, Digital Twin, and Self-Supervised Traversability Prediction

Trung Kien La¹ and Eric Guiffo Kaigom¹

Abstract—Collision prediction and avoidance are essential for the autonomous navigation of mobile robots. The prediction of a traversability map is a way to achieve this goal. However, this approach might fail if the prediction model is exposed to novel semantic classes unseen during self-supervised training or the environment is subject to highly dynamic motions of living and artificial entities. Included are pedestrians, animals, and vehicles that the robot can hardly handle. In this paper, we embrace this challenge by describing the development of a flexible human-robot teaming that leverages the shared control of the robot to accommodate critical situations and allow the human operator to be otherwise engaged. The operator can remotely perceive the surrounding and actively guide the robot as well as participatively share or leave full control to the robot that autonomously moves toward a desired common goal. A situation-aware control signal balances the input of the motion planner of the autonomous robot and the cognitive input of the remote operator issued by using affordable interfaces. The human-in-the-loop control is realized through a bidirectional wireless communication between the physical robot and its digital twin. The human-robot teaming enhances the likeliness of a safe and successful navigation under dynamic obstacles and enables a navigation without pre-defined goals. The conceptual architecture is introduced and preliminary results are shared.

I. INTRODUCTION

The navigation of mobile robot in unstructured environments requires the detection, interpretation, and avoidance of obstacles. Self-supervised learning can be leveraged to train models that predict the traversability map from RGB and depth images data forwarded as inputs to a motion planner to generate collision-free motions to a desired location [1]–[4]. Even though traversability values can incorporate the semantics of obstacles beyond their standard geometric detection, their reliability can drop in dynamic and previously unseen environments [5], [6] where people or other entities exhibit a high motion dynamics, too. Furthermore, pivotal parameters are often overlooked by these models. Included are the battery state of charge (SoC) and engine temperatures of the robot that are equally important to predict a feasible and safe navigation. It is therefore common for robots to be teleoperated by a human operator [7], [8]. However, teleoperation might require a constant attention of the operator, which prevents the operator from accomplishing other (e.g., rescue) tasks or being engaged otherwise at the same time.

In this paper, we strive to relax this constraint. We focus on a loosely-coupled collaborative interaction between a robot and its remote operator with a customizable level of



Fig. 1. Human-robot-teaming for mobile robot navigation.

human intervention and robot autonomy during safe robot navigation. On the one extremity, an experienced situation understanding and supervision skills of humans help remotely guide and assist the robot in challenging situations. Uncertainties from unseen classes are expected to be accommodated and transformed into advantage. On the other extremity, autonomous navigation abilities of the robot allow its human operator to carry out additional tasks while the robot moves toward a desired location. Between full guidance and entire autonomy, cognitive skills of the human operator (e.g., perception, empathy, anticipation, etc.) are being superimposed on top of obstacle-aware commands of the motion planner of the robot to adjust the behavior of the robot and further upskill its navigation. This collaboration is likely to raise the success rate of navigation by combining the skillsets of humans and robots in a seamless and dynamic way.

Toward this end, we provide the robot operator with a web-based graphical unit accessible from most devices. The unit interfaces the robot through its digital twin, reflects the perception of the internal state of the robot and its surroundings as live video streams, visualizes its critical data, and offers multiple control options at e.g. velocity level. We strive to sum velocity signals from the motion planner of the mobile robot and the graphic unit used by the operator to realize a shared control [9] of the robot that fosters human-robot-teaming. In this case, the autonomous behavior of the

¹The authors are with the Department of Computer Science & Engineering, Frankfurt University of Applied Sciences, 60318 Frankfurt am Main, Germany, e-mails: trung.la@stud.fra-uas.de; kaigom@fb2.fra-uas.de

robot and amount of human interventions are balanced by an arbitration factor that depends upon the traversability map of the robot in real-time. Security aspects such as login and persistent data storage via a database are taken into account.

II. RELATED WORK

Virtual-reality (VR) based interactions have been recently proposed in [7] for the intuitive oversight of robot teams. The VR interface supports the reconstruction of pre-recorded 3D models and real-time capture of scenes, similarly to our work. However, traversability was not considered to semantically handle obstacles. There are several web-based apps that have been developed for ROS robots such as the Husky Unmanned Ground Vehicle (UGV) from Clearpath (see Fig.1) [8]. These applications are often simple, intuitive and inexpensive as they control virtual robots and are suitable for ROS novices. They are based on the Rosbridge protocol, which provides an interface between ROS and web apps via a WebSocket connection [10]–[13]. The protection of the app or the visualization and storing of parameters such as the real battery SoC play an increasingly critical role for autonomous robots. Hence, among other things, we implement a database and visualize ROS data that the Husky provides via its topics. Further implementations are described in III. Operators have been provided in [14] with egocentric and exocentric 3D perspectives on robots. However, shared control to balance control modes between a planner and operator was omitted.

III. WEB-BASED INTERFACE FOR REMOTE OPERATOR

The web framework is based on Flask [15] and is used to control and monitor the Husky, which runs on ROS2 Humble and is platform-independent. A bidirectional communication between human and robot is provided by the `rosbridge_suite`

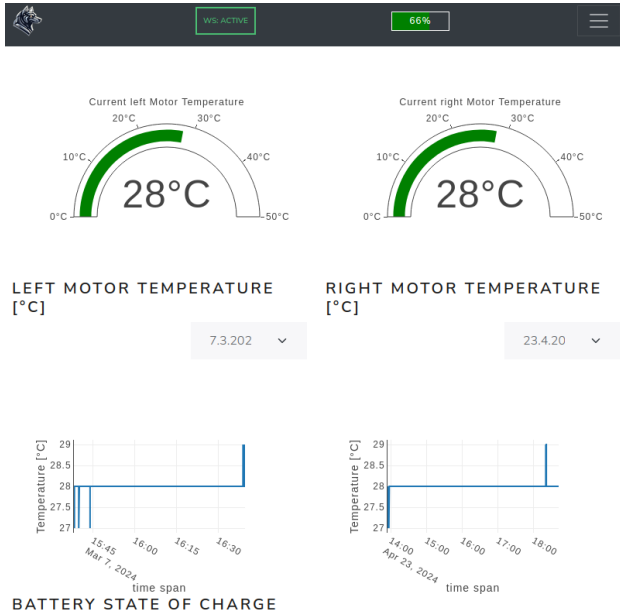


Fig. 2. Data panel as component of the digital twin (monitoring service/interface layer in fig. 4). Sensed left and right motor temperatures are presented live as gauges. Historic data managed via a SQLite database.

package [13]. Due to the web-based approach, all devices with a browser can access the app, provided they are in the same local network as the robot. Fig. 5 shows the backend architecture, ROS packages, and other relevant information which were leveraged for developing the framework. A twist message, which consists of angular and linear velocity, can be sent from the webpage (frontend) via e.g. an intuitive virtual joystick to the `/cmd_vel` topic of the robot to teleoperate it. Visualized otherwise invisible ROS data about the robot (see Fig. 2 and 5) offer a digital twin for decision support.

A. Control panel

The control panel in Fig. 3 shows a live image stream of a Zed2i camera attached on the Husky. Above the camera image, a navigation bar is located. Below the camera image there is a speed slider and two control elements such as a virtual joystick and virtual keyboard keys for controlling the Husky. The arrangement of the control elements was chosen to reflect the natural hand position when operating the displays. In addition, the Husky can be controlled with a physical keyboard when the operator has access to the control panel, which is protected by a login system. Keyboard control can be helpful in situations where finer precision is required compared to the joystick. Other elements such as the battery SoC and the WebSocket connection are also shown embedded in the navigation bar to ensure that the operator is always informed about the energy and connection status of the robot to assess the feasibility of a planned navigation.

B. Data panel

The data panel in Fig. 2 shows critical live robot data such as the left and right motor temperatures as gauges. Furthermore, the temperatures and the battery SoC are shown in diagrams with time progression. The data for the visualization is provided by an SQLite database, which in turn reads the data from the respective ROS topics of the Husky. It is possible to select and visualize historical logs of single days in the diagrams and snapshots can be taken if required.

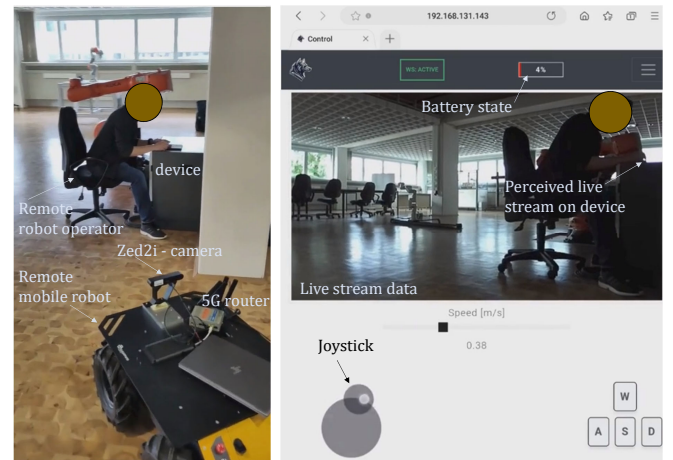


Fig. 3. Remote control (top left) of the Husky robot (bottom left) in real-time via the web interface with a virtual joystick, a speed slider and live camera images (top right). The battery state helps predict motion feasibility.

C. Additional features of the operator interface

A user login system is implemented for privacy preservation and security purposes. A six-layer digital twin architecture (see fig. 4) supports the monitoring, 3D visualization, and control of the robot. Robot perception and supervision are thereby facilitated. A specially developed OPC UA server keeps the framework open, semantically interoperable, scalable, and industry-relevant. The server stores and exposes e.g. current states of the mobile robot and manipulators mounted on it in real-time. The web interface visualizes these states by using the digital twin and acts as a multi-modal immersion interface. Bootstrap, a frontend framework, was leveraged to facilitate responsive web design [16]. This allows control elements and data visualizations to be optimally displayed on mobile or desktop devices and ensures that the operator experiences a pervasive, contextualized, and personalized overview and collaboration with the robot.

IV. MULTI-MODAL REMOTE CONTROL OF THE ROBOT

We strive to offer three modalities (see fig. 6) for the remote control of the mobile robot in our ongoing work. During full control, the remote operator actively guides the robot. Interface details and results on this control mode are provided in section III. Its core advantage is the direct usage of experienced human skills and knowledge to yield a thoughtful robot behavior. Since the operator might be occupied with the completion of tasks with a higher priority, the second autonomous control mode differs in the sense that the path planner of the robot aims at achieving safe motions and the operator does not intervene. To this end, the path planner is provided with traversability values from a prediction model trained by using self-supervised learning [4]. A semantic understanding of the terrain is incorporated in these values. Unlike geometrical approaches, semantic obstacle awareness facilitates an intelligent estimation of traversability in the presence of different properties of objects. While a stiff wall and tall grass can be geometrically identical, the latter is traversable and the former is not. As human intervention becomes critical, the third shared control

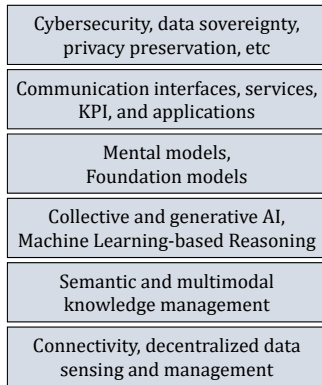


Fig. 4. Six-layer architecture of our digital twin. Bottom = 1. layer. Top = 6. layer. 5G connectivity is in the 1. layer. Monitoring, simulation/extended reality/Metaverse, and visualization services as well as the computation of Key Performance Indexes (KPI) are carried out in the 5. layer.

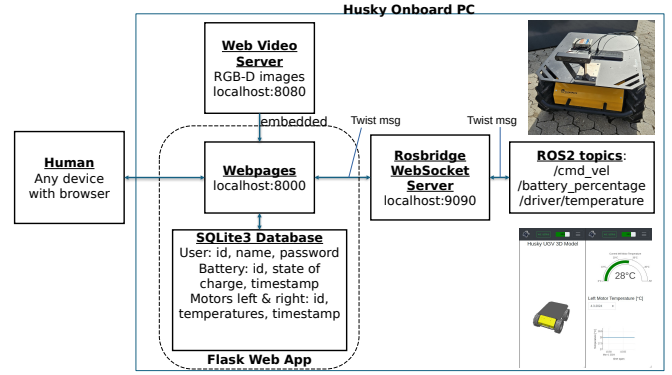


Fig. 5. Backend architecture of the web app and tools used. The digital twin (bottom right) reflects hidden and hardly accessible motor and battery states to assess and predict the feasibility of planned motions.

mode balances previous schemes by combining human and machine intelligence. Flexibility from autonomy and acumen from supervision are likely to enlarge the range of supported applications and enhance the operator experience in practice.

A traversability map contains pairs of linear traversability value $\mu \in [0, 1]$ and angular traversability value $v \in [0, 1]$ for each position on the terrain. A traversability value close to one indicates that a navigation is possible whereas a value close to zero means that the robot gets stuck [1]. Instead of handling two scalars, the linear and angular traversability values can be combined to yield

$$\alpha = \frac{\lambda_1 \mu + \lambda_2 v}{2} \quad (1)$$

with the weighting coefficients

$$\lambda_1 \in [0, 1] \quad (2)$$

and

$$\lambda_2 \in [0, 1]. \quad (3)$$

Observe that $\alpha \in [0, 1]$, too. $\mu = 0$ and $v = 0$ together lead to $\alpha = 0$. Hence, the scalar α can be used to characterize the traversability, as employed in fig. 7 for control purposes.

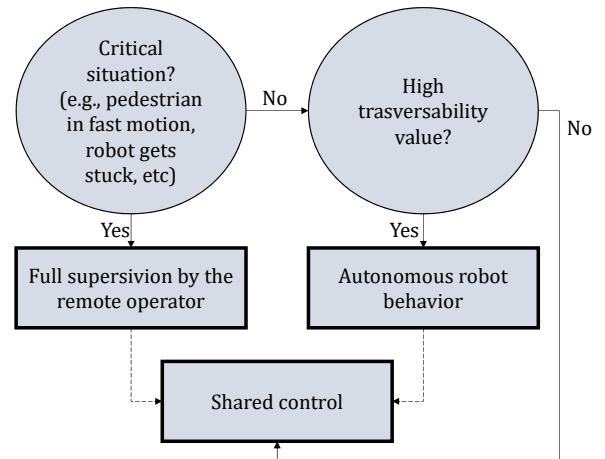


Fig. 6. Simplified overview of the multiple modes for remote control

