Toward Human-Robot-Teaming for Mobile Robot Navigation Using Sensor Fusion, Remote control, Digital Twin, and Traversability Map

Trung Kien La[1] and Eric Guiffo Kaigom[1]

***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 or artificial entities. Included are pedestrians, animals, and vehicles that the robot can hardly handle. In this paper, we embrace this challenge by describing our ongoing 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 successful navigation under dynamic obstacles and enables a navigation without pre-defined goals. The paper introduces the conceptual architecture and shares preliminary results.**

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

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 a model that predicts the traversability map from RGB and depth images data forwarded as inputs to a motion planner to generate collision-free motion to a desired location [1]–[4]. Even thought traversability values can incorporate the semantics of obstacles beyond their standard geometric detection, they 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) or 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 human intervention and robot autonomy during the 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 thereby 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. In-between full guidance and entire autonomy, cognitive skills of the human operator (e.g., perception, empathy, anticipation, etc.) are being superimposed on top of semantics-aware commands of the motion planner of the robot to refine or adjust the behavior of the robot and 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 use velocity signals from the motion planer 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

[1]The authors are with the Deptartment 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 value 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 Rosbrigde 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. Bidirectional communication between human and the robot is provided by the rosbridge_suite 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. a virtual joystick to the /cmd_vel topic of the robot to teleoperate it. The ROS data published by the robot is in turn displayed on the app (see Fig. 2).

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.

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 individual days in the diagrams and snapshots can be taken if required.

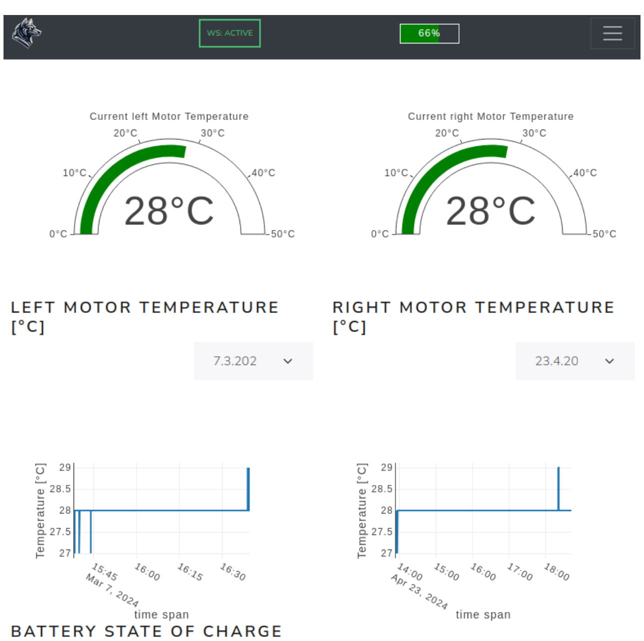

Fig. 2. Data panel with left and right motor temperatures which are ae presented live as gauges and historic data is presented via the SQLite database.

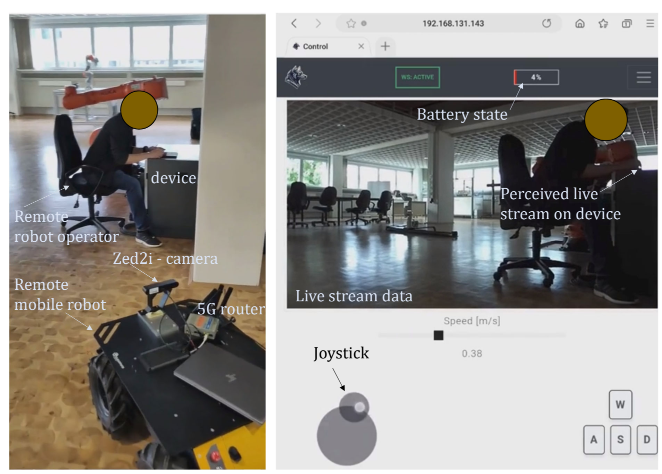

Fig. 3. Remote control of the Husky robot in real-time via the web interface with the virtual joystick, a speed slider and the live camera image.

C. Additional features and Considerations

Additional features of the framework include a user login system for security purposes and a six-layer digital twin architecture (see fig. 4) for monitoring, visualization, and control of the Husky 3D model to further increase the human perception and supervision of the robot. The database not only stores ROS and user data, but is also connected to a specially developed OPC UA server. This server provides data from manipulators, e.g., which can be mounted on the Husky. The web interface visualizes this data. It can therefore be regarded as a holistic visual data source. Bootstrap, a frontend framework, was leveraged to facillitate responsive web design [16]. This allows control elements and data visualizations to be optimally displayed on mobile or desktop devices of the robot operator and ensures that the robot operator retains an overview and control of the robot.

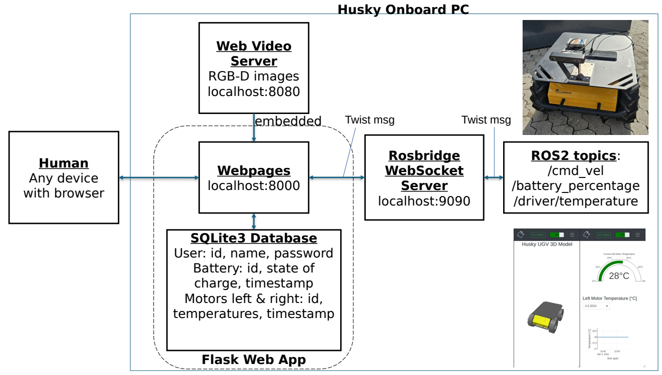

Fig. 5. Backend architecture of the web app and tools used. The digital twin (bottom right) reflects hidden and hardly accessible motor and battery states to assess and predict the feasibility of planned motions.

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. Details and result on this mode have been described 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 approach, semantic situational 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 very identical, the later is traversable and the former is not. As human intervention becomes critical, the third shared control 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 $\nu \in [0,1]$ for each position of 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

$$\alpha = \frac{\lambda_1 \mu + \lambda_2 \nu}{2} \tag{1}$$

with the weighting coefficients

$$\lambda_1 \in [0,1] \tag{2}$$

and

$$\lambda_2 \in [0,1]. \tag{3}$$

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

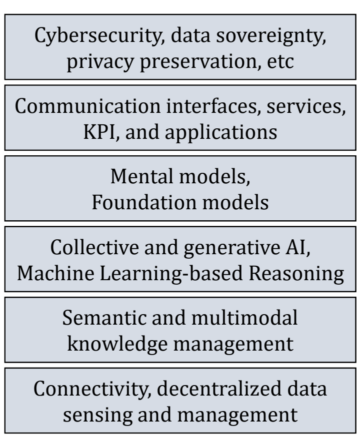

Fig. 4. Six-layer digital twin architecture.

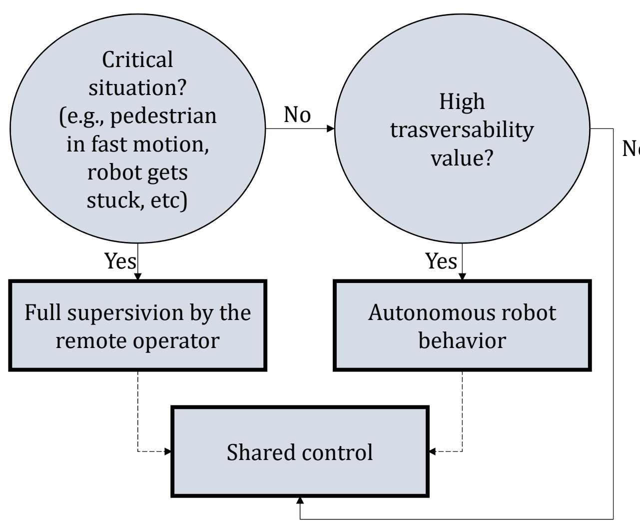

Fig. 6. Simplified overview of the multiple modes for remote control

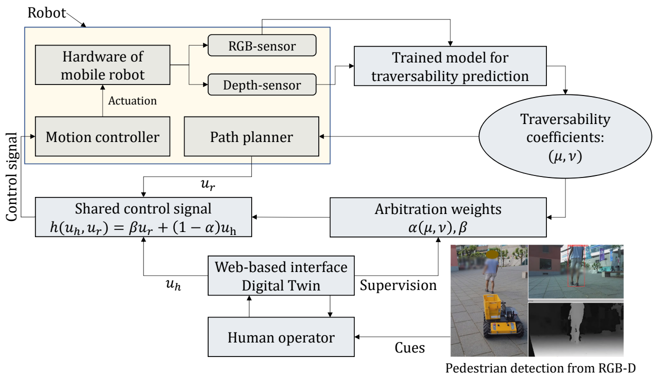

Fig. 7. Shared control scheme

A. Autonomous robot behavior

Conditions for a full robot autonomy to happen are that the arbitration variable $\beta = 1$ and $\alpha \rightarrow 1$ in figs. 6 and 7. As a result, the control velocity h is essentially the output of the motion planner of the robot u_r (see fig. 7). It is worth to note that the path planner takes advantage of the traversability values to plan a collision free path that safely guides the robot to the desired goal without human intervention. Such a path planner can be the Dynamic Window Approach or Rapidly-Exploring Random Tree scheme [2] to cite a few.

B. Full robot supervision

Trained classification models employ images to detect the presence of critical obstacles, such as pedestrians (see fig. 6), interpret their behavior, and inform the operator using cues, as depicted on the right-hand-side of fig. 7). Methods for behavior identification based upon graph neural networks are suitable to this end [17]. In fig. 7, RGB and depth data from a state-of-the-art Stereolabs ZED 2i stereo camera sensor (see fig. 5) are forwarded to a pre-trained YOLOv8 model for pedestrian detection (position and orientation). The operator is also aware of potential obstacles around the robot through visual feedback. To manage the situation, the operator takes over the full control of the robot ($\beta = 0$ and $\alpha = 0$), in which case the operator input (i.e., $h = u_h$, see fig. 7) is directly forwarded to the motion controller as commanded velocity.

C. Shared control between operator and robot

In this control mode, the velocity issued by the human operator (u_h) is superimposed on top of the velocity u_r generated by the motion planner (note that $\beta = 1$) to yield the total commanded velocity. The lower the traversability value, the higher weighting of the contribution of the velocity from the operator to the total velocity. If the operator is otherwise engaged ($u_h = 0$), the robot evolves autonomously ($h = u_r$). Once the operator is informed or sees that the robot gets stuck ($\alpha \rightarrow 0$), the operator can intervene ($h \approxeq u_r + u_h$) and skillfully sustain the navigation of the mobile robot.

V. CONCLUSIONS

This paper has introduced different modes for the multi-modal remote control of a mobile robot for collaborative navigation purposes. The proposed shared control foster human-robot-teaming and supports the continuum between the full supervision of the robot motion using human intelligence and the autonomous robot behavior based upon a semantic-aware machine intelligence (e.g., traversability values). The approach has the potential to enable a wide range of applications and provides many benefits to the human operator. These include workflow flexibility and enhanced navigation experience. The approach is suitable for not only wheeled but also e.g. legged robot mobile robots.

REFERENCES

[1] M. V. Gasparino et al., ”WayFAST: Navigation With Predictive Traversability in the Field,” in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10651-10658, Oct. 2022, doi: 10.1109/LRA.2022.3193464.

[2] Leung, T., Ignatyev, D. & Zolotas, A. Hybrid terrain traversability analysis in off-road environments. *2022 8th International Conference On Automation, Robotics And Applications (ICARA)*. pp. 50-56 (2022)

[3] Endo, M., Honda, K. & Ishigami, G. BenchNav: Simulation Platform for Benchmarking Off-road Navigation Algorithms with Probabilistic Traversability. *IEEE ICRA 2024 Workshop on Resilient Off-road Autonomy*. (2024)

[4] M. V. Gasparino et al., ”WayFASTER: a Self-Supervised Traversability Prediction for Increased Navigation Awareness,” Feb. 2024 arXiv preprint arXiv:2402.00683.

[5] Frey, J., Khattak, S., Patel, M., Atha, D., Nubert, J., Padgett, C., Hutter, M. & Spieler, P. RoadRunner–Learning Traversability Estimation for Autonomous Off-road Driving. *ArXiv Preprint ArXiv:2402.19341*. (2024)

[6] Muhamad, F., Kim, J. & Park, J. Robust Traversability Prediction Using Multiple Costs for Quadruped Robot in Random Terrains. *IEEE Access*. (2024)

[7] Huang, C., Chou, S., Liou, L., Moy, N., Wang, C., Wang, H., Ahn, C., Huang, C. & Yu, L. An Evaluation Framework of Human-Robot Teaming for Navigation Among Movable Obstacles via Virtual Reality-Based Interactions. *IEEE Robotics And Automation Letters*. (2024)

[8] ”Husky Unmanned Ground Vehicle,” Clearpath, https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/ (accessed Aug. 9, 2024).

[9] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi and B. Siciliano, ”Autonomy in Physical Human-Robot Interaction: A Brief Survey,” in IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 7989-7996, Oct. 2021, doi: 10.1109/LRA.2021.3100603.

[10] Zinaid Kapić, Aladin Crnkić, Edin Mujčić and Jasna Hamzabegović, ”A web application for remote control of ROS robot based on WebSocket protocol and Django development environment”, in IOP Conf. Ser.: Mater. Sci. Eng. 1208 012035, 2021, doi: 10.1088/1757-899X/1208/1/012035.

[11] D. D. Rajapaksha et al., ”Web Based User-Friendly Graphical Interface to Control Robots with ROS Environment,” 2021 6th International Conference on Information Technology Research (ICITR), Moratuwa, Sri Lanka, 2021, pp. 1-6, doi: 10.1109/ICITR54349.2021.9657337.

[12] Crick, C., Jay, G., Osentoski, S., Pitzer, B., and Jenkins, O. C. ”Rosbridge: Ros for non-ros users” in Robotics Research, Springer, 2017.

[13] ”rosbridge_suite,” rosbridge suite - ROS Wiki, http://wiki.ros.org/rosbridge_suite (accessed Aug. 9, 2024).

[14] Walker, M., Gramopadhye, M., Ikeda, B., Burns, J. & Szafir, D. The Cyber-Physical Control Room: A Mixed Reality Interface for Mobile Robot Teleoperation and Human-Robot Teaming. *Proceedings Of The 2024 ACM/IEEE International Conference On Human-Robot Interaction*. pp. 762-771 (2024)

[15] ”Flask,” User’s Guide, https://flask.palletsprojects.com/en/3.0.x/

[16] ”Introduction,” Bootstrap, https://getbootstrap.com/docs/5.3/getting-started/introduction/ (accessed Aug. 9, 2024).

[17] Jang, S., Lee, H., Kim, W., Lee, J., Woo, S. & Lee, S. Multi-scale Structural Graph Convolutional Network for Skeleton-based Action Recognition. *IEEE Transactions On Circuits And Systems For Video Technology*. (2024)