Shared Autonomy: A PLACEHOLDER TITLE

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

***Abstract*— ... and highlight the main aspects of the Web Framework.**

I. INTRODUCTION

Geometric navigation approaches that are for example LiDAR based, are especially limited outdoors [1]. Self-supervised autonomous navigation systems such as BADGR [2], WayFAST [1] or the successor model WayFASTER [3], are therefore used to estimate the traversability with regard to bumps and navigable objects such as tall grass. Even though these camera-image based neural networks excel in a static environment, they are constrained in a dynamic and busy environment where people, animals or other objects such as cars are moving. Furthermore, not all parameters can be taken into account in these models. Parameters such as battery state of charge (SoC) or engine temperatures of the robot are equally important to ensure successful and safe navigation. It is therefore common for robots to be additionally monitored and controlled by a human operator. For these reasons, it is advantageous for a person with better situational awareness to take back control for a certain action or period of time. In this respect, shared autonomy describes a robotic system that independently adapts its autonomy level to the given environmental factors [4]. To support this physical robot-human interaction (pHRI), intuitive control and monitoring of critical data is required. We therefore provide the robot operator with a web-based interface that reflects the perception of the robot, visualizes its critical data and offers multiple control options. Security aspects such as login and persistent data storage via a database are also integrated. In addition, a concept for shared autonomy with the Husky UGV mobile robotic platform from Clearpath [5] and WayFASTER are presented.

II. WEB FRAMEWORK

The web framework is based on flask 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 [source]. 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. 1 shows the backend architecture, ROS packages and other relevant information which were leveraged for developing the framework.

[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

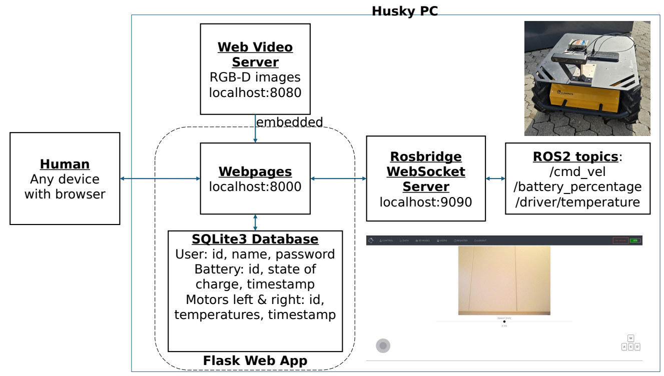

Fig. 1. Backend architecture of the web app and tools used.

A. Control panel

The control panel in Fig. 2 shows a live image stream (in the center) 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. In addition, the Husky can be controlled with a physical keyboard when the operator has access to the control panel. Keyboard control can be helpful in situations where 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.

B. Data panel

The data panel in Fig. 3 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 visual-

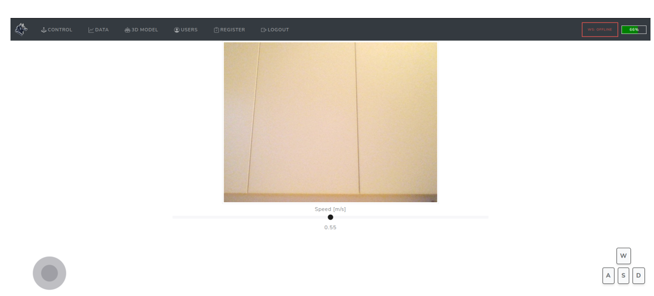

Fig. 2. Control panel of the Web App with a navigation bar on top, camera stream in the center, control elements left and right and a speed slider to adjust the robot speed.

ization 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.

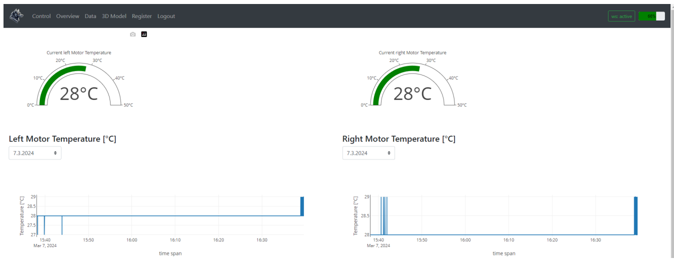

Fig. 3. Data panel with left and right motor temperatures which are ae presented live as gauges and historic data is presented via the SQLite database.

C. *Additional features and considerations*

Additional features of the framework include:

- User login system for security purposes
- Visualization of the Husky 3D model
- Database connection with a specially developed OPC-UA server, to store other robot data apart from the Husky

Bootstrap, a frontend framework, was leveraged to facillitate responsive web design [6]. 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.

III. RELATED WORK

- Shared Autonomy - Presentation of BADGR and WayFASTER - Web based UI Apps?

IV. METHODOLOGY AND CONCEPT

We propose a shared autonomy approach of the guidance of a robot such as the Husky UGV from Clearpath. The idea is to let the robot take over initial control through a self-supervised navigation system such as WayFASTER. The kino-dynamic model of a skid-steered Robot, such as the Husky, is given below: [3]

$$x_{k+1} = \begin{bmatrix} p_{x_k} \\ p_{y_k} \\ \theta_k \end{bmatrix} + \begin{bmatrix} \mu(p_{x_k}, p_{y_k}) \cdot cos(\theta_k) & 0 \\ \mu(p_{x_k}, p_{y_k}) \cdot sin(\theta_k) & 0 \\ 0 & \nu(p_{x_k}, p_{y_k}) \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}$$

where x_k is the state vector, p_x and p_y are the coordinates and θ_k is the orientation (heading angle) of the robot. The $cos(\theta_k)$ and $sin(\theta_k)$ functions describe the movement in x- and y-Axis depending on the orientation. $\mu(p_x, p_y)$ and $\nu(p_x, p_y)$ are outputs (traversability coefficients) of the WayFASTER model TravNet and have a value between 1 and 0 respectively. They indicate how well the robot can move across the terrain and are associated with the control inputs v_k and ω_k, which are the linear and angular velocities. In ROS terms are these are what a Twist Message contains.

We plan to integrate the transversability coefficients into our web framework. Since these coefficients have a value of 1 for good transversability and 0 for poor transversability, a threshold can be specified. A value of 0 (for both coefficients) would mean that the robot stops. To ensure a more seamless transition from autonomous control to human control, falling below the defined threshold will result in a warning in the web app. This is intended to indicate the human operator to take control within the app. The app is programmed in such a way that human input always has top priority. Specifically, this can be achieved with Husky by sending control commands (geometry_msgs/Twist_Message) in different topics with different priorities [source].

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 4. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

V. CONCLUSIONS

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] G. Kahn, P. Abbeel and S. Levine, "BADGR: An Autonomous Self-Supervised Learning-Based Navigation System," in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1312-1319, April 2021, doi: 10.1109/LRA.2021.3057023.

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

[4] 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.

[5] https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

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

[7] "rosbridge_suite," rosbridge suite - ROS Wiki, http://wiki.ros.org/rosbridge_suite (accessed March 14, 2024).