RoboCup 2024
Standard Platform League (SPL)
Technical Challenge
— SHARED AUTONOMY —

Adrian Müller
Desmond Krämer
Samuel Njike Megaptche
Thomas Jäger
Wilhelm Simus

last updated: July 16, 2024

1   The Challenge 2024: Shared Autonomy

In the 2024 technical challenge, teams must create a mixed team comprising one human-operated NAO robot and one fully autonomous NAO robot. Matches will be played in a two vs. two format on the standard SPL field.

1.1   The Goal

The challenge aims to progress towards enabling robots to operate alongside agents with human-level intelligence. To ensure a level playing field in terms of physical capabilities, all participants will use NAO robots. Each team will have one robot remotely operated by a human, providing human-level intelligence, while the other robot will function autonomously, adhering to the main SPL competition rules.
For further references and exact rules, see the official challenge rules provided by the technical committee.

2   Our Approach

Our approach involves developing a web-based interface to control the robots and facilitate communication. The backend system communicates with the robots via TCP, sending byte commands that are interpreted and executed by the robots. In order to act sensibly as a human operator, the robots must be able to communicate the most important sensor readings (camera, behavior status, . . .). Since we did not yet include these inputs into our web-interface, our human operator uses the camera view and information views from the B-Human simulator SimRobot, which is connected to a live bot.
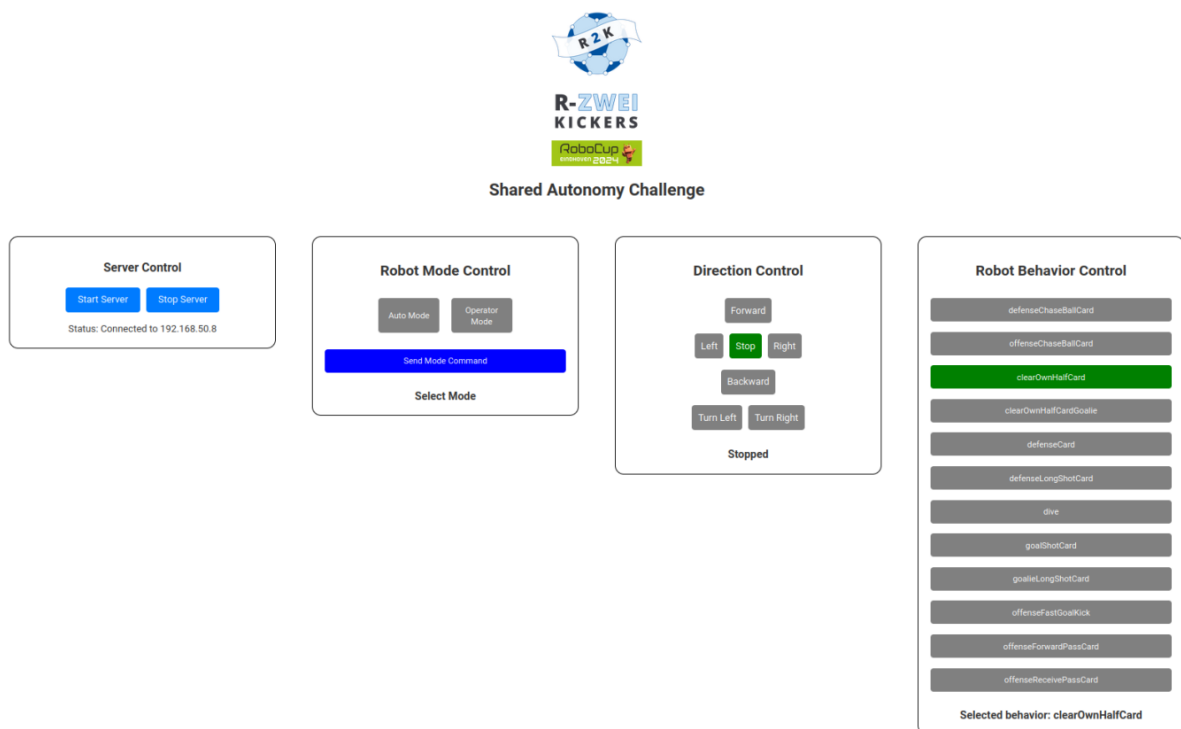
Figure 1: The Interface for the shared autonomy challenge consists of four distinct control modules.

2.1 Tactical Overview

The interface is divided into several control modules (see figure 1):

- **Server Control**: This module starts and stops a server, which opens a listener port, that the controlled robot communicates with.

- **Robot Mode Control**: This module allows switching between autonomous operation and full human control. This can be done piecewise, s.t. autonomous actions and fully controlled actions are possible, resulting in semi-autonomous behavior.

- **Direction Control**: Provides an interface for complete human control over the robot's movements.

- **Behavior Control**: Enables manual selection of behaviors from a prepared card stack using the B-Human skills and cards framework, which we use since branching off from B-Human's 2021 code base.

2.2 Technical Overview

Our system is based on a minimalist web-app using Svelte for the frontend and a Python Flask server for the backend. The server acts as a REST interface, converting commands from the frontend to byte messages sent to the robot. The robot-side module interprets these messages, integrating the commands into a custom representation for use by other modules and parameterizing skills or behavior cards.

3 Future Enhancements

To further enhance our system, we plan to improve inter-robot communication by improving message exchanges between robots. While the robots currently use our implemented inter-robot communication, we'd like to include a few strategic options optimizing gameplay (Passing the ball and communicating a pass target etc). Additionally, we aim to refine our tactical capabilities, offering more sophisticated and nuanced control options. Furthermore, we want to include the controlled robots sensor readings (cameras, microphones, ...), s.t. a future implementation does not need to run in parallel with the views from simulation.

3.1 Roadmap

1. **More Sensor Readings in Interface:** The views currently handled by the simulator should be moved into our control web-app.

2. **Inter-Robot Communication**: Implement triggers for message exchanges between robots to coordinate actions and strategies.

3. **Enhanced Tactics**: Develop advanced tactical views and strategies to optimize robot performance in various scenarios.

4. **User Interface Improvements**: Enhance the web interface to provide more intuitive and user-friendly controls.