

## Graphical Interface for Simultaneous Interaction with Multiple Robots

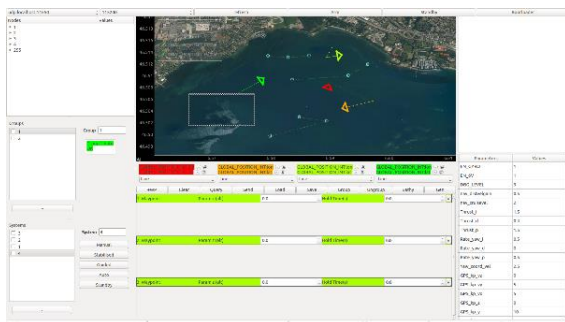
Ahmed Saadallah

Professor : Alcherio Martinoli  
Assistant(s) : Anwar Quraishi, Felix Schill

The purpose of this project is to enhance the user interface of a ground station software used to control autonomous surface vehicles and also autonomous underwater vehicles.

The focus is to support interaction with multiple robots simultaneously for swarm operations and make it easy for the user to control them while outdoors. Therefore, while implementing new features, I considered different approaches and chose those that are the most intuitive and suitable to the work environment.

Also message failure handling has been improved by dealing with or loss of messages in a smarter manner depending on the criticality of the message.



*Main interface of mavue*

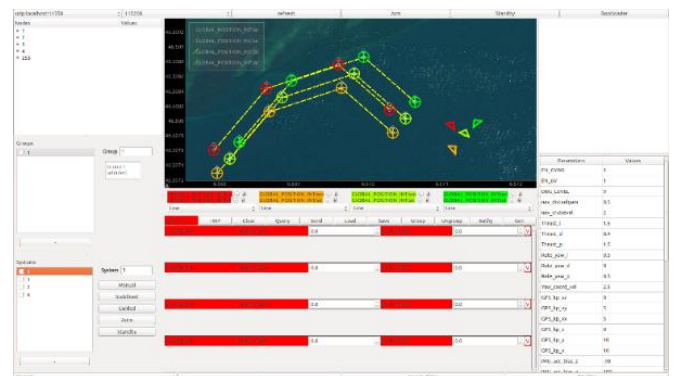
Previously multiple systems could be added but the user could control only one. I extended that concept to multiple systems. When a new system is added, new markers and list of waypoints are assigned to it. Markers are shown automatically on the map interface to highlight the current position systems and their orientation. Each system has a distinct color translated in the marker, the map legend and the list of waypoints.

To be able to specify which set of vehicles are subject to instruction, a selection tool is implemented. The selection box is rectangular and is enabled by right clicking on the map and can be resized with the motion of the mouse.

Through selection of a set, a repeatable task for each vehicle can be simplified to be executed only one time for all selected vehicles in order to save time and minimize the number of actions needed for a certain instruction. In this sense, a table containing a list of on-board parameters common to all vehicles is added. These parameters can be tuned and sent to

all designated vehicles to change the velocity or the thrust for example.

Upon selecting a subset of vehicles, the user can define a new group with its ID, members and list of waypoints. Groups show up in a group selector widget and a status text box is added to visualize which is selected and its mode status. When a group is selected and a list of waypoints is defined on the map, all members of that group get their respective list of waypoints updated following a constructed pattern. The pattern is built by setting the same offset position that separates the initial member systems of the defined waypoints to create new waypoints for each system respectively. The idea behind this is to maintain the initial formation of the group.



*Pattern waypoints for 4 vehicles*

For message failure handling, after evaluating which messages are crucial for the user, a mechanism of checking whether a message is well received is implemented. In case of important messages like vehicle state changes, the message would be sent infinitely until success is confirmed. The status of the sending procedure is displayed through a textbox and the text on the state buttons. It goes the same for switching different controlling modes like auto, guided and manual.