

CAVS Remote Control Dashboard User Guide

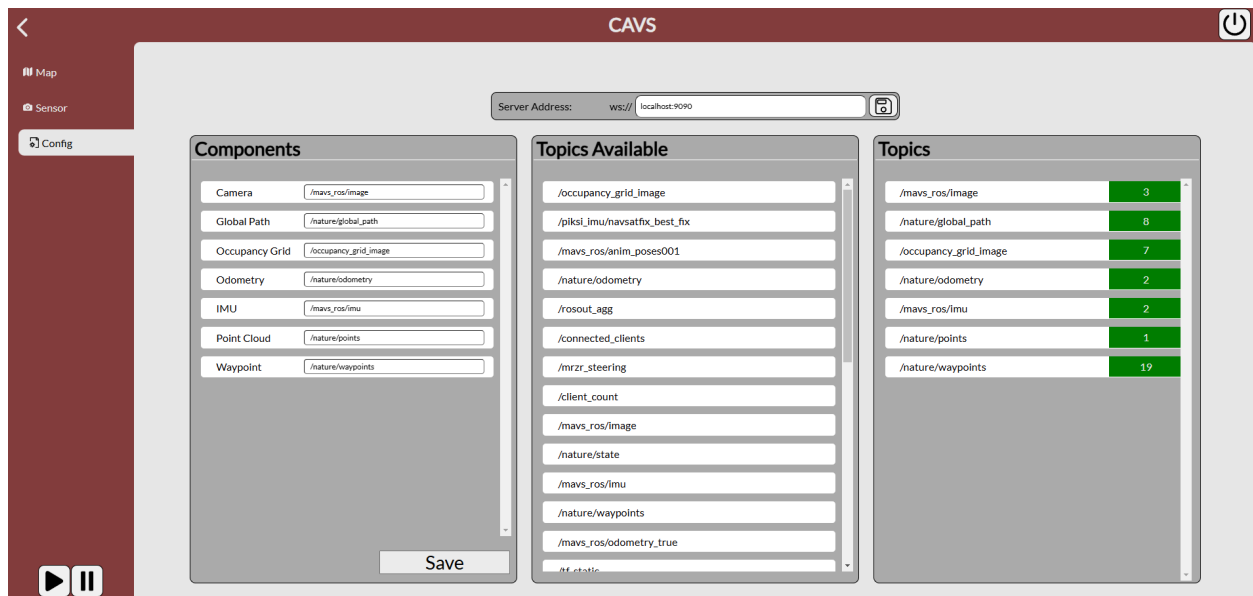
1. Introduction

This dashboard allows for remote monitoring of autonomous vehicle behavior during testing missions. It provides quick and intuitive access to mission-critical information including: camera feed, global path, local path, occupancy grid, odometry, IMU, point cloud, and mission waypoints.

The dashboard is split into three tabs, each with a distinct purpose: map, sensor, and config. Details on the functionality of these tabs and how to use them are given below. Additionally, each tab contains start, pause, and shutdown buttons. The start/pause buttons are located on the bottom left of the screen, and the shutdown button is on the top right.

2. Config Tab

Before using the dashboard, you must connect to the vehicle using the config tab and bind components to their respective topics. A screenshot of the config tab is displayed below.



In the top middle of the tab is the "Server Address" box. This is where you will enter the address of the rosbridge websocket that provides vehicle data. Once you enter the address, click the save icon on the right to update the application. If the dashboard is receiving data, the

“Topics Available” card will fill with topics. The “Topics Available” will list all advertised topics from the ros websocket. If the dashboard is NOT receiving data (likely because there is not a websocket at the specified server address), this card will remain empty, as seen in the screenshot below.



On the left-hand side of the config tab is the “Components” card. The component options displayed in this card are static, as they correspond to elements displayed on other pages. To activate these components, enter the desired topic path in the component’s text box and click “Save”. This allows you to unsubscribe from the old topic and subscribe to the new topic. If a card ever says “No <Component Name>” you likely have an invalid topic bound here. Camera and Occupancy Grid expect a **sensor_msgs/Image** message. Global Path and Waypoint expect a **nav_msgs/Path_Message** message. Odometry expects a **nav_msgs/Odometry** message. IMU expects a **sensor_msgs/Imu** message.

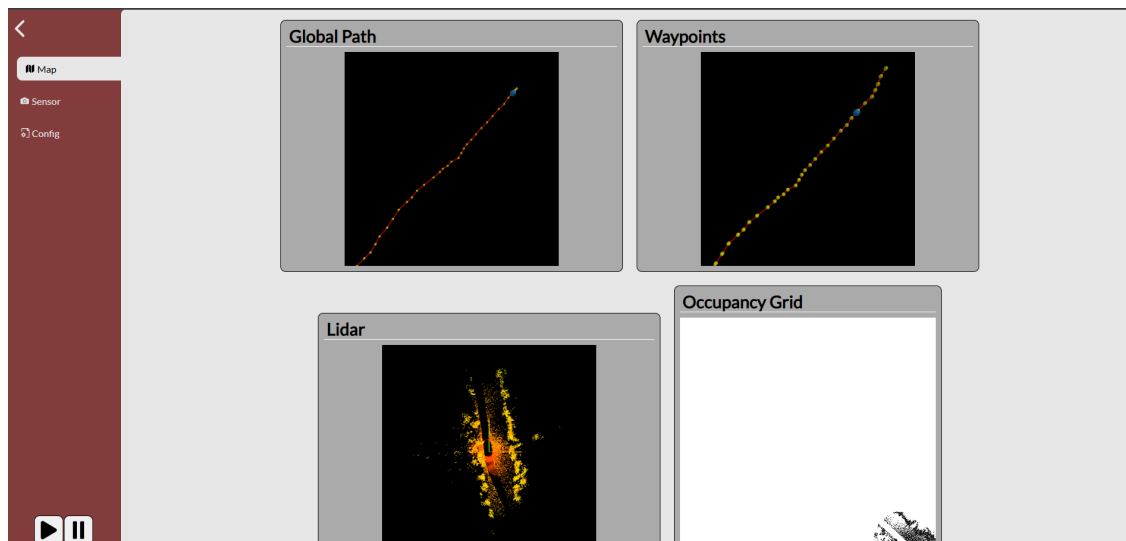
On the right-hand side of the config tab is the “Topics” card. This card displays the topics bound to a component from the “Components” card. The numerical value displayed with each component is the refresh rate in hertz. This refresh rate is based on the time between the last ten messages received from that topic. If the topic displays green, it is receiving data from the

topic (note that all refresh rates are raised to the next integer value). If it is red, no data from that topic is being received by the application. The elements in this card are draggable, allowing you to change the order in which the component refresh rates are listed.

The config tab will set a cookie to remember your server address and component bindings between sessions. When this cookie is not present, the application will load from `/src/DefaultValues/ConfigData.jsx` for a default server URL and component bindings. This file can be changed before launching the application, but please note that if the cookie remains in a browser, the ConfigData file will not be read.

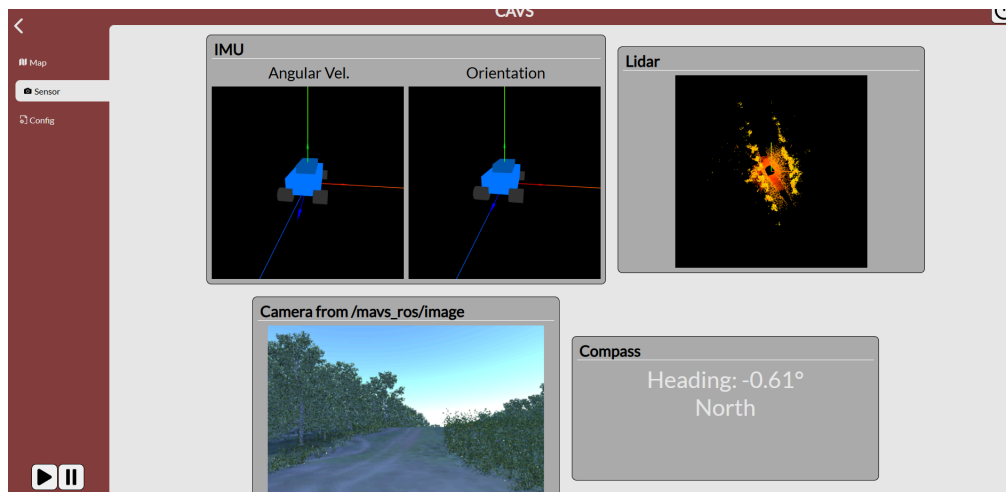
3. Map Tab

The map tab contains information relating to the vehicle's position. This includes the global path, the vehicle's current position on the path defined by the mission waypoints, the occupancy grid, and LIDAR. The Global Path and Waypoints require their respective component to be receiving data to appear. They use data from the Odometry component to place the vehicle at the correct location on the map, but this data is not required for the component to render the path. Lidar requires data from the Point Cloud component and from Odometry. If either of these are missing, the application will abort the render. The Occupancy Grid displays the occupancy grid image. A screenshot of the map tab is displayed below.



4. Sensor Tab

The sensor tab displays the camera feed, IMU, LIDAR, and compass headings. The IMU card visually represents the Angular Velocity obtained from the IMU component and the orientation data obtained from the Odometry component. The Lidar card on this tab functions the same as on the Map tab. The Camera card processes the image obtained from the Camera component. The Compass card relies on the twist obtained from the Odometry component to display the compass direction the vehicle is facing. A screenshot of the sensor tab is displayed below.



5. Missing connections

Visualization cards have three error states. The first is “websocketless” which will cause all cards to display “No websocket connection.” To fix this, go to the config tab and change the server address to a valid websocket. The next error state occurs when a card is missing critical data for the visualization. The card title will be red, and a message will indicate which component is missing. The final error state occurs when a card is missing non-critical data, this will change the card title to orange and inform of missing data from a component. Both of these missing data warnings can be seen in the image below. To resolve these issues, ensure that the named component is bound to a ros topic that provides the correct data type, as specified in section 2.



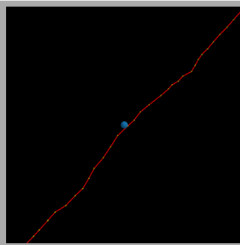
Map

Sensor

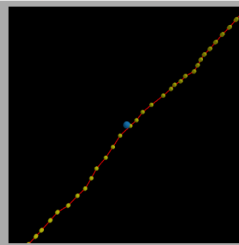
Config



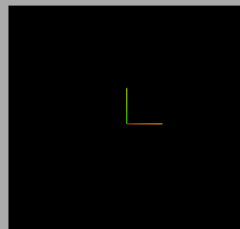
Global Path | No Odometry



Waypoints | No Odometry



Lidar | No Odometry



Occupancy Grid

