

# Introspection and Debugging with `pal_statistics` in ROS 2 Control

Maximilien Naveau (PAL Robotics)

## Duration

Standard talk ( $\approx 20$  minutes), 10 to 15 minutes of presentation + 5 to 10 minutes of Q&A.

## Abstract ( $\leq 100$ words)

This talk presents the integration of `pal_statistics` within ROS 2 Control to provide real-time introspection of controller inputs and outputs. This feature enables developers to monitor internal signals without user intervention, significantly improving debugging and tuning workflows. Beyond ROS 2 Control, `pal_statistics` can be used independently to expose metrics that tools such as PlotJuggler can parse seamlessly. We showcase a demo—implemented in `ros2_control_demos`—where we run the `ros2_control_example_1` and show case that every controller exposes its inputs/outputs. An additional node is ran to present the independent functionality of `pal_statistics` and its use with PlotJuggler.

## Detailed Description ( $\leq 1000$ words)

Debugging and introspection are critical for developing reliable real time robotic systems. The real time aspect is important as it means that the tools and methods used should not interfere with the real time loop of the robot. Moreover, controllers in ROS 2 Control typically expose only their command and state interfaces, but internal intermediate computations are often hidden from developers. This can make it difficult to identify bottlenecks, numerical issues, or misconfigurations during development.

To address this, PAL Robotics developed `pal_statistics`, a lightweight library designed to publish arbitrary metrics and statistics efficiently. The ROS 2 Control framework has integrated `pal_statistics` directly into its controller interface layer, allowing every controller to expose its inputs, outputs, and internal states immediately, without requiring additional user code.

This integration provides several advantages:

- **Automatic Introspection:** Every controller can publish its internal metrics, making debugging faster and more transparent.
- **Seamless Debugging Tools:** Tools like PlotJuggler can automatically parse and visualize these metrics thanks to recent updates (3.10.11).
- **Independent Usage:** `pal_statistics` can also be used outside ROS 2 Control in any ROS 2 node, enabling a consistent introspection workflow across different robotic subsystems.

## Example Use Case

For the live demo, we will use `ros2_control_example_1` to run the joint trajectory controller on the `rrbot`. With `PlotJuggler`, we will demonstrate how input/outputs state signals are visualized for each controller in real time. To further illustrate the flexibility of `pal_statistics`, we will add a simple Python node that uses `pal_statistics` independently of ROS 2 Control, interacting with the `example_1` setup. This will show how metrics can be exposed and visualized outside the controller framework, providing a consistent introspection workflow.

## Audience Takeaways

- Learn how `pal_statistics` integrates into ROS 2 Control for automatic introspection.
- Discover how to use `pal_statistics` independently in custom ROS 2 nodes.
- Understand how to visualize and debug controller internals using PlotJuggler.

## Resources

- pal\_statistics: [https://github.com/pal-robotics/pal\\_statistics](https://github.com/pal-robotics/pal_statistics)
- ROS 2 Control: [https://github.com/ros-controls/ros2\\_control](https://github.com/ros-controls/ros2_control)
- PlotJuggler Changelog 3.10.11: <https://github.com/facontidavide/PlotJuggler/blob/main/CHANGELOG>
- Demo repository: [https://github.com/ros-controls/ros2\\_control\\_demos](https://github.com/ros-controls/ros2_control_demos)  
[https://github.com/ros-controls/ros2\\_control/blob/master/hardware\\_interface/include/hardware\\_interface/L180C38](https://github.com/ros-controls/ros2_control/blob/master/hardware_interface/include/hardware_interface/L180C38)
- ROS 2 Control integration (controller\_interface\_base.cpp): [https://github.com/ros-controls/ros2\\_control/blob/master/controller\\_interface\\_base.cpp](https://github.com/ros-controls/ros2_control/blob/master/controller_interface_base.cpp)

## Illustration



*(Placeholder for a PlotJuggler screenshot)*