Creating a ROS2 Framework for a 3 Finger Hand Robot

Jihwan Ben Shin
jihwan.shin@sjc.ox.ac.uk

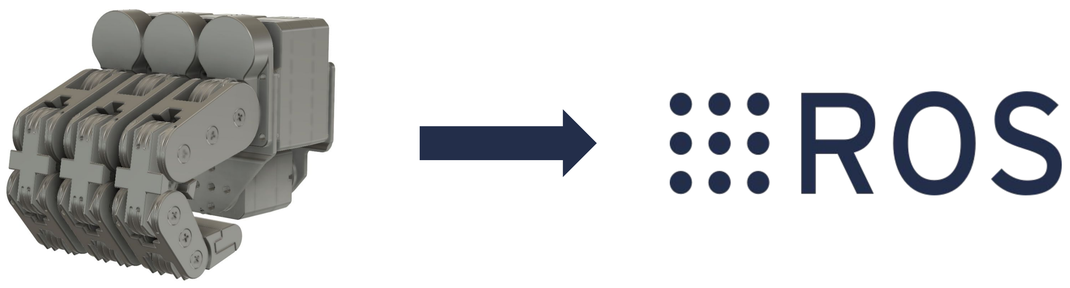
UNIVERSITY OF OXFORD
OXFORD ROBOTICS INSTITUTE
DYNAMIC ROBOT SYSTEMS GROUP

Background & Aim

The robot was created by a previous student intern based on IRIM Lab's original design of a 4-DoF 3 Finger Hand.

It has **4 servo motors**, one controlling the rotation of the thumb (*Palm*) and the rest controlling the tendons of the three fingers (*Thumb, Index, Ring*). It is a **low cost, compliant** alternative to dexterous hand robots.

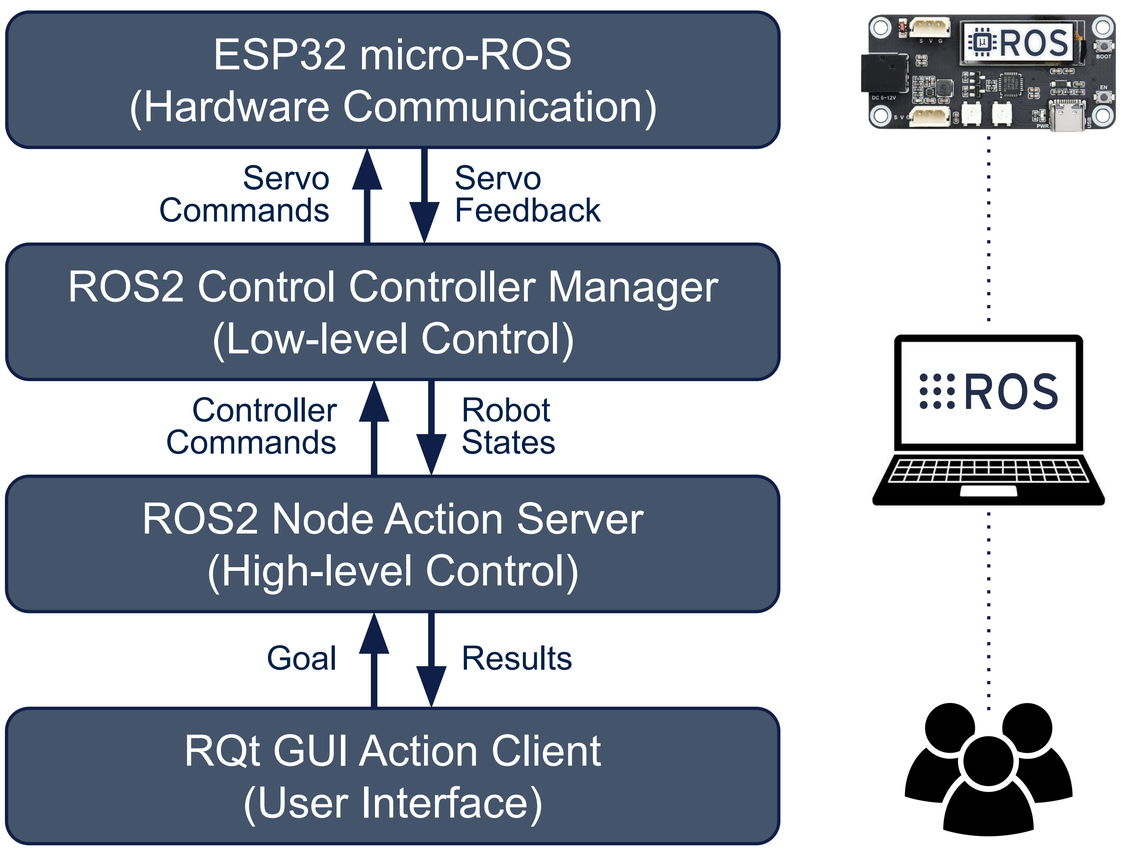
The aim of this project is to create a ROS2 framework which allows easy integration to different robotic systems and to test the functionality of the hand in high level control applications.



ROS2 Framework

The project was built using **ROS2 Humble** on Ubuntu 22.04, which is an **open source** framework supporting **node-based** communication between components in a robotic system. This is connected to **micro-ROS** (ROS2 for microcontrollers) flashed on an ESP32 servo driver board, which sends and receives hardware communications with the servos.

- **Modular**: makes it easier to fix and combine with other robotic systems (e.g. robotic arm)
- **Expandable**: flexible to integrate with different controllers and libraries for new applications
- **Distributed**: computing and communication are done in both the main computer and the ESP32 microcontroller



Controllers & GUI

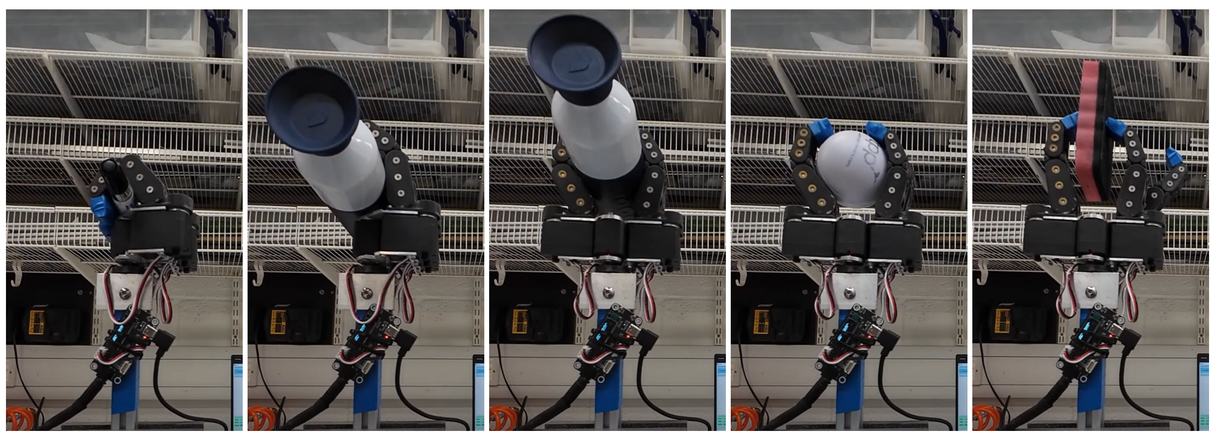
The robot's **low-level controls** are done by **ROS2 Control controller manager**, which **abstracts the hardware components** to work with forward position and velocity controllers.

High-level controls are done by connecting the controller manager to a **ROS2 node** running **action servers**.

- **Palm Action Server**: Receives **position** goals for the palm servo which rotates the thumb until the goal is met.
- **Finger Action Server**: Receives **velocity** and **torque** goals for the finger tendons. The fingers will close, open, or maintain at the specified velocity until a torque limit is met (indicating a firm grip on an object).

The **desired configurations** are sent from their **action clients**, which are implemented in **RQt** (ROS2-Qt) **GUI** (Graphical User Interface) for an easier user interaction.

Demo



The hand has been tested for **power grasp**, **parallel grasp** and **pinch grip** with the implemented controllers and user interface.

By expanding the high-level controller and user interface, the hand has the possibility to perform a more diverse set of actions. Potential ideas include a thumbs up, clicking a pen, spherical grasps, etc.

Scan to watch the full demo videos online

Limitations

- **Limited High-level Control**: Due to it being an under actuated system with elastic connections, the pose of the hand cannot be estimated by position. This makes it difficult to achieve precise high-level controls that make use of its dexterous design.
- **Difficulty in Simulation**: For the same reason, it is difficult to simulate the hand in a virtual environment. This limits the possibility to teach more complex behaviours connected to a robotic arm or a mobile system. To achieve this, additional actuators or couplings will be required.

Acknowledgements
I would like to thank Prof. Ioannis Havoutis and Pratik Somaiya for providing support with this project.

EPSRC Vacation Internship
07 November 2024