1-2 Robot – Status report 1

Kevin GAITAN FIGUEROA
Ernesto RINCON MARQUEZ

Progress

Software/electronics

- Raspberry Pi 3B+ (Rpi)
  - Backed up the code from the Rpi
  - Added Github repo to pull code for testing as we develop from our machines
  - Although no modification was made to the OS, not plugging a display and logging into the RPi via SSH reduces considerably RAM usage. (~65% → ~40% at startup)
  - Setup local lab network for SSH access to RPi
- ROS
  - Started reorganizing the Dynamixel packages to get rid of unnecessary dependencies and centralize the code (currently most of the code lives inside the example libraries of Dynamixel sdk)

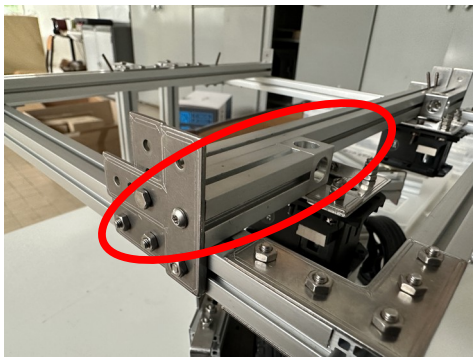
Hardware

Several improvements were made to the chassis to improve rigidity:

- Added additional mounting bars to keep the orientation motors from sliding across the axis
- Secured electronics platform to chassis using self locking nuts
- Redesigned link between orientation and spin motors using 3D printed adapters to ensure that the rotation axes of the orientation actuators line up with the Z axis of the wheels. This also produces a more mechanically sound assembly

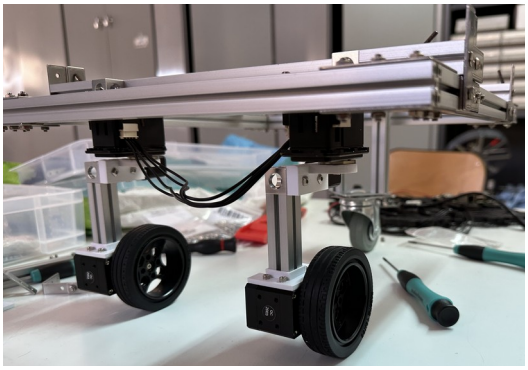
Evidence

New bar for orientation motors:



Old steerable wheel assembly

New steerable wheel assembly



Next steps

- Review ROS2 documentation for main differences with ROS1
- Create node for low level motor control (using ROS2)
  - It’s necessary to group several snippets of code into a single file, as currently everything is spread out in several of the Dynamixel examples
  - This node will be responsible for setting the motor modes and sending commands to the motors
- Create desired trajectory generator
- Finish building kinematic model and create its container node which will calculate joint velocities
- Create skeleton controller node for testing the whole system
- Create node for obtaining current motor status (position, speed, etc) and publishing to a topic
- Adapt GUI to be able to send commands to low level control node
- Adapt simulation node