2020 Field Testing Manual

Rover Setup, Connection, and ROS Configuration

Authors: *Braden S. & Pierre-Lucas A.F.*
Date: *August 17, 2020*

1. Introduction

This document is designed to be a self-contained directive on rover operation for future researchers and students at the Aerospace Robotics Laboratory (ARL). Some content is reiterated from the document *ConnectingControllingHuskyAndArgo* by J.S Fiset (see resources at `https://github.com/brahste/data_collection`). However, substantial changes to the field testing procedures have been implemented, warranting a new manual to explain the setup, both experimental and configurational.

2. Sensors & Hardware

An overview of the sensor and hardware suite is shown in Fig. 1. Three primary sensors are equipped on the rover setup:

1. a ZED stereo-camera (ZED cam). This dual-camera device acts as the eyes of the experimental setup and collects visual data in a variety of forms. For more information on this device see Appendix B.
2. a VectorNav-100 Inertial Measurement Unit (IMU). This device gathers information about linear accelerations and angular velocities induced in the rover chassis. It also provides integrals of these dynamic values (linear velocities, linear positions, angular positions). See Appendix B for more information.
3. Motor current sensors (MC sensors). These sensors detect the current being drawn by the front right and left wheels. See Appendix B for more information.

These sensors are connected as shown in Fig. 1 to two different computers, a third computer is used to remotely access and command the experiments; information on these computers follows:

1. *arl-thinkpad*: a Lenovo P53 laptop. This laptop is bound to the top of the Husky and is equipped with a GTX 1080 Ti Graphical Processing Unit (GPU) which allows it to operate the ZED cam. The *arl-thinkpad* is responsible for hosting the data collection experiments, for initializing the sensor suite, and for commanding data logging procedures and rover motions.
2. *cpr-conu1*: a rover dedicated computer. This computer is located in the bed of the Husky and serves as the ROS interface to Husky's drivetrain, as well as to the IMU and MC sensors. Although this computer plays an essential role in the field testing sessions, it is never directly accessed, instead commands to this computer are implicitly defined within the ROS configuration on *arl-thinkpad*.

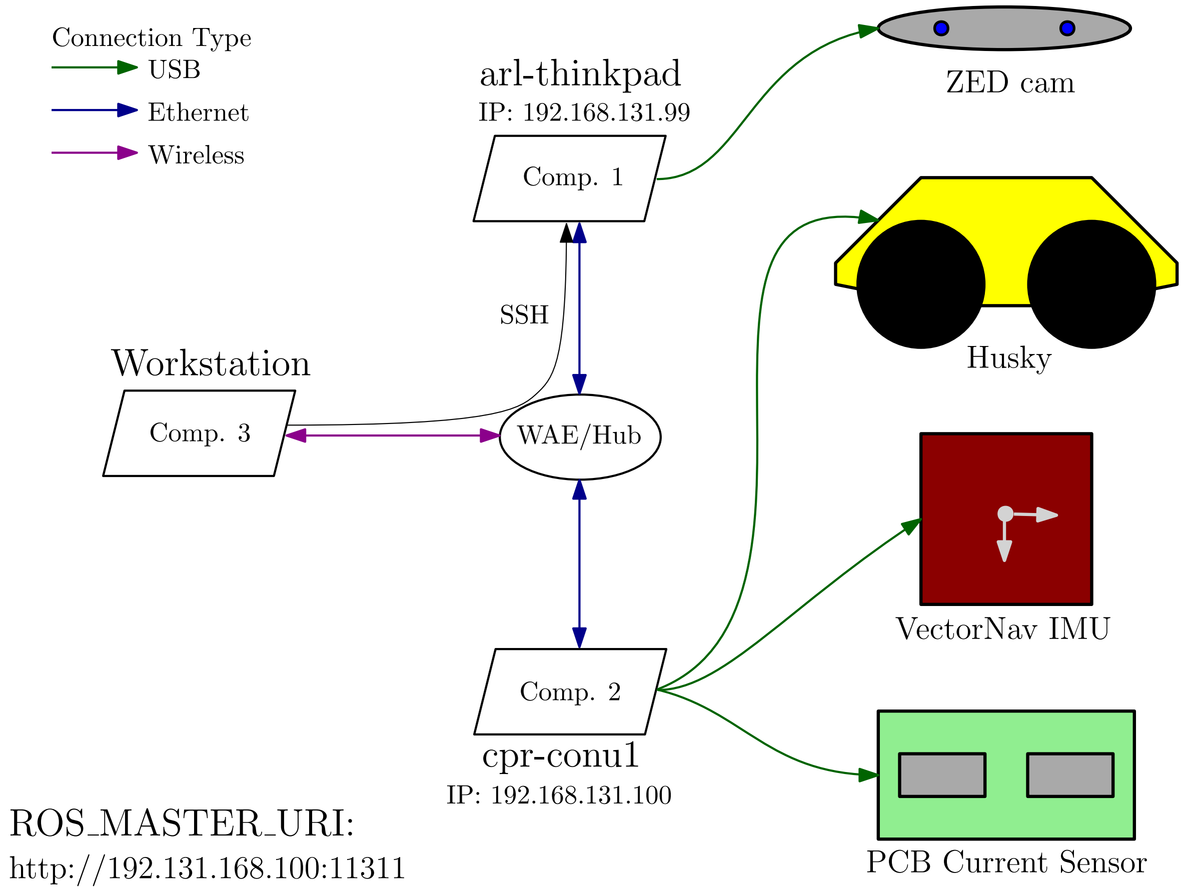

Figure 1: Overview of hardware Setup for 2020 field testing session. The workstation uses SSH to access *arl-thinkpad*, where the `data_collection` package is located. Sensor nodes and motion commands are launched from *arl-thinkpad*, sending information over the network to *cpr-conu1* when needed. Notice that although commands are sent from *arl-thinkpad*, the Master URI is configured to be on *cpr-conu1*; the reason for this is related to launch-on-boot features of the Husky and is discussed further in Section 4.

3. Workstation: any other computer suitable for the needs of the experiment. This computer is used to SSH into *arl-thinkpad*, from which the commands are launched. Ideally, this computer is a small, portable laptop with a strong battery life.

To incorporate the ZED cam into the sensor suite a laptop with a CUDA capable GPU is required. For this purpose, a Lenovo P53 with a GTX 1080 Ti is used.

3. Network Configuration

To command the rover from the Main Computer, ensure that it is connected through a hub. For our experiments a Cisco Wireless Access Point/Hub was used.

3.1. Usage.

4. Software Configuration

4.1. Overview. The field testing experiments are implemented using an array of different ROS packages, shell scripts, and devices; at the highest level these verious software items are localized and governed by the ROS package `data_collection`. Code for the package is hosted at https://github.com/brahste/data_collection. The main access point to the hardware suite is through the file `husky_sensors.launch`. Since Husky launches its nodes on boot-up (from the files in `/etc/ros/kinetic/`), `husky_sensors.launch` is responsible for initializing the nodes for peripheral sensors only. A secondary consequence of the Husky nodes intializing on boot is that it is simplest to use *cpr-conu1* as the Master URI (as shown in Fig. 1).

Although *arl-thinkpad* hosts the main data collection package and the ZED node, packages for the IMU and motor current sensors had been configured for past field tests on *cpr-conu1*; we have left this configuration intact. Since the Husky node initializes on boot, for this field testing session it was chosen to use *cpr-conu1* as the Master URI. Future researchers may find it desirable to connect and configure all sensors to *arl-thinkpad* and disable the launch-on-boot feature of Husky, this way the Master URI could be assigned to *arl-thinkpad* and complications regarding network latency would be less pronounced.

Note that `husky_sensors.launch` initializes nodes that exist on both *arl-thinkpad* and *cpr-conu1*. To enable this feature an environment configuration file `husky_env.sh`, located in the main catkin workspace of *cpr-conu1*, must be called when launching nodes over the network.

4.2. Usage.

1. Before moving forward with this section, ensure the steps from Section 3.1 are complete. Most importantly the `ROS_MASTER_URI`, hostnames, and IPs must be set correctly. At this point the Husky node should be running, this can be confirmed by calling `rosnode list`. If the nodes [...enter the names of a few nodes...] are up, then indeed the Husky node is running.

2. Initialize the peripheral sensors with:
   `roslaunch data_collection husky_sensors.launch`

   Running `rostopic list` you should now be able to see that `imu/rpy`, `imu/imu`, and `zed/*` are being published. As a sanity check (or for debugging) you can run /coderostopic echo <topic_name> and see the data streams from each sensor in your terminal.

Upon `roslaunch`-ing, the master node is initialized, along with the following nodes:

5. Running the Experiment

you can use a box as such

A. Operating System Configuration

B. Node Information