Cavecrawler 2018 Documentation

1. Description

The cavecrawler is a wheeled mobile robot built in 2004 in the Robotics Institute. It uses two 12v car batteries and a bank of power inverters to supply the AC power for the forward driving motors, which are controlled by a Kollmorgen drive (ServoStar CD, for the complete manual see https://www.kollmorgen.com/zu-za/products/drives/discontinued/cd-series-5/servostar_cd_series_hardware_and_software_reference_manual/). The robot's steering is commanded using a servo-controller (B80001 Brushless Servo Drive, for the complete manual see https://www.cnczone.com/forums/attachments/1/4/1/0/6/3/115098.attach) to move linear actuators that change the orientation of the wheels.

Due to the robot's old operative system and electronics, as well as the lack of software documentation, the only way to control it is the joystick input through the gamepad or the gamepad-like ROS interface. The last requires two Arduino boards with a Bluetooth hardware. The first Arduino is connected to the computer with the BT configured as master. The other Arduino is connected to the robot, and the BT configured as slave. The instructions to move the robot with both methods are listed below.

1.1 Moving the robot with a joystick

- Connect the joystick receiver in one of the USB hubs located on the sides of the electronics board.
- Turn on the inverters (black box next to the electronics).
- Turn on the robot using the key.
- Wait a couple of minutes until you hear a beep. After that the motors are enabled, so you should see a "1." in the forward motors drives display and the green led blinking in the steering servo controllers.
- Now press z button on the joystick and the wheels should enter in normal driving position. A detailed list of button functionalities is provided in Figure 1.
- Use the stick to move the robot.

1.2 Moving the robot with the ROS interface

- Connect the slave Arduino in one of the USB hubs located in the sides of the electronics board.
- Turn on the inverters (black box next to the electronics).
- Turn on the robot using the key.
- Connect the Arduino with the master BT interface and the joystick receiver to the computer.
- Run the nodes required for the driving interface.
- Wait a couple of minutes until you hear a beep. After that the motors are enabled, so you should see a "1." in the forward motors drive and the green led blinking in the steering servo controllers.

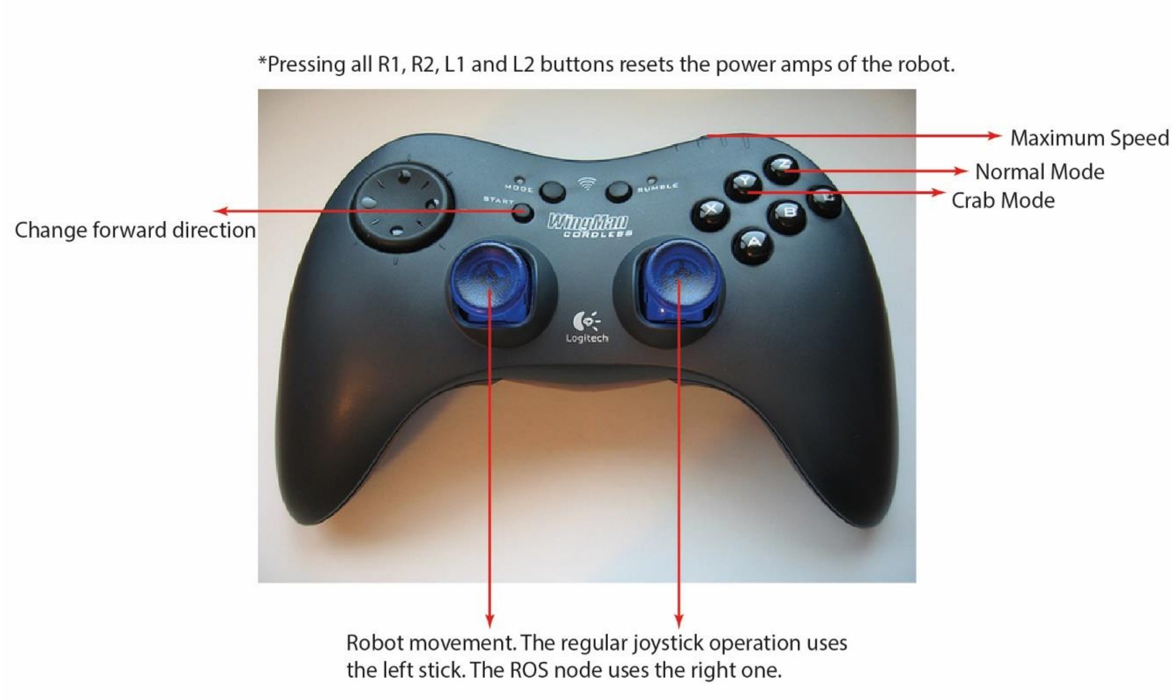


Figure 1. Joystick functionalities.

2. Hardware

2.1 Electronics board

The main board contain various electronic components for the cavecrawler and the wiring rails for the logic and communication between the sensors and the main computer. If either 5, 12, 24 volts or ground connections are required, they can be obtained from various points of the rails. The lines and its nomenclature in each rail is labeled in the wires and documented in the robot folders (see cavecrawler's hardware documentation provided in the lab).

The key elements for the movement of the robot (that can be accessed by the user) are the configuration switches and the fuses. The first ones control the power and logic to enable the drive and steering motor controllers. Figure 2 shows the switch configuration that enables the motors so that the robot can be operated. It also shows the electric diagram and the PCB of the board, to give insights about the setting controlled by each switch.

The fuses, on the other hand, are protections for the AC power lines for the motors, the DC lines for the computer and the "old" sensors. They are supposed to turn a led on when are burn but most of them don't, so it could be a potential starting point for troubleshooting.

2.2 Forward drive encoder and steering servo controller

Encoder readings from the forward drive motors are provided in the C4 connector of the controller. Figure 3 shows the C4 connector location and its pinout. Channel A or B are a frequency outputs that linearly changes with the speed of the motor. To read the frequency, an Arduino code is available, user only needs to connect the ***channel A or B +*** to one interrupt pin of the Arduino and the DC common (C4 pin 3) to the Arduino's ground. ***Channel A or B*** – must <u>not</u> be connected to the Arduino.

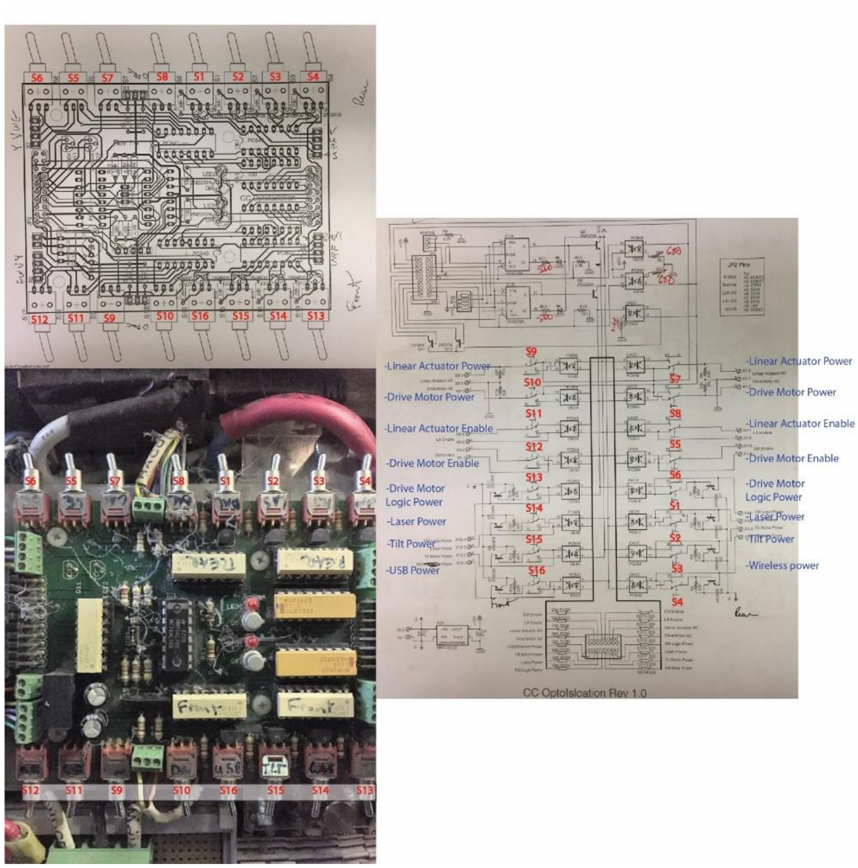
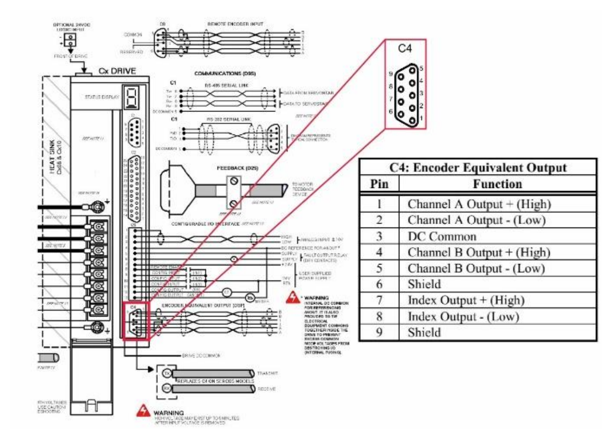


Figure2. Working switch configuration. PCB and electric diagram of the switch board.



C4: Encoder Equivalent Output	
Pin	Function
1	Channel A Output + (High)
2	Channel A Output - (Low)
3	DC Common
4	Channel B Output + (High)
5	Channel B Output - (Low)
6	Shield
7	Index Output + (High)
8	Index Output - (Low)
9	Shield

Figure 3. C4 output and its pinout for getting odometry readings from the forward drive motors.

Tests conducted with a free rolling setup and using the channel B (C4 pin 4) resulted in linear regressions for the right and left motors shown in Figure 4.

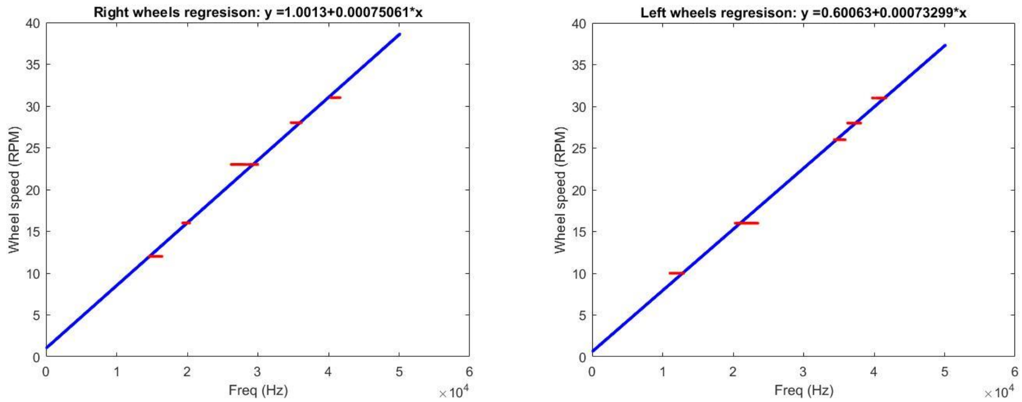


Figure 4. Linear regressions for right and left wheels encoders with a free rolling setup.

Regarding the steering feedback, tests showed that angle changes in the wheels are related with voltage changes in the ***Command -*** pin of the servo-controller, measured with respect to ground (ground connection of the steering controller is the same as the forward drive motors controller). Thus, 2 ADC boards (available in the lab, with a I2C communication protocol) can be used to measure such voltage changes using an Arduino or a similar board. Experimental tests must be carried on in order to find the relationship between voltage and actual wheel steering.

3. Software

3.1 Gamepad-like ROS interface

In order to use an Arduino as a joystick interface, the system has to recognize it as an HID device. HID are the "credentials" that an USB-based hardware requires to be recognized as a mouse, keyboard or a gamepad, as in this case. The HID library provided by *Nicohood* for Arduino (https://github.com/NicoHood/HID) allows to use boards Leonardo or (Pro) Micro directly as HID devices. It is only necessary to add the library to the Arduino IDE and include the proper header files in the sketch. For detailed instructions see the wiki page of the link provided before.

To use other boards like UNO or MEGA as HID devices, an extra procedure is required to change the bootloader settings of such boards. This procedure is also well documented in the wiki page of the *NicoHood* project. If using this approach, it must be considered that the employed Arduinos won't work to other purposes than HID interfaces. To get them back to normal, the original bootloader needs to be uploaded again.

The current implementation of this step employs an Arduino NANO with the master BT and a Leonardo board with the slave.

3.2 Odometry

A basic frequency meter is provided for any Arduino board to read the C4 output of the forward drive motors.

3.3 Accelerometer

A ROS node is available at: http://wiki.ros.org/xsens_driver

4. Miscellaneous

4.1 Wireless Access

There is an access point available to perform basic operations with the robot. The main use given for this interface is to properly turn off the robot before removing the power.

These are the steps required to access this interface:

- Set your IP address to 192.168.1.99
- Set the Subnet Mask to 255.255.255.0
- Default gateway: leave blank
- Connect to wireless network Cavecrawler or cavecrawler_a. The password for both is **rescue10**.
- Open the web browser and access to 192.168.1.15

4.2 Other stuff

If required, some interesting documentation for the cavecrawler kinematic modelling and velocity profiling can be found in:

- https://www.ri.cmu.edu/pub_files/pub1/shamah_benjamin_1999_1/shamah_benjamin_1999_1.pdf
- https://www.ri.cmu.edu/pub_files/2010/8/WMR_Kinematics.pdf
- https://www.ntu.edu.sg/home/edwwang/confpapers/wdwicar01.pdf
- http://www.frc.ri.cmu.edu/~jizhang03/Publications/IROS_2013.pdf

By: Francisco Yandun
March 2018