

GEORGIA INSTITUTE OF TECHNOLOGY

CENTER FOR ROBOTICS & INTELLIGENT MACHINES

HUMANOID ROBOTICS LAB

GT Mission Manual

M.X. GREY
ERIC HUANG
ANDREW PRICE
PETER VIEIRA



Contents

Code Repositories	2
Real-time Code	2
ROS Code	2
Tasks Overview	3
Wall Drilling Task	3
Procedure	3
Rubble Clearing Task	4
Procedure	4
Operation Instructions	5
Starting-up Hubo	5
Starting-up Perception	6
Walking	7
Manipulation	9
Troubleshooting	10

Code Repositories

The following code repositories are required in order to execute our tasks using our code. They consist of real-time processes that run on the robot, and code that uses ROS and runs mostly on a local computer.

Real-time Code

Code repositories that run in real-time on the robot.

- hubo-ach
 - <https://github.com/hubo/hubo-ach>
 - * Dependencies: *ach, libreadline*.
- hubo-motion-rt
 - <https://github.com/hubo/hubo-motion-rt>
 - * Dependencies: *hubo-ach, RobotKin, urdfdom, eigen3*.
- hubomz
 - <https://github.com/golems/hubomz>
 - * Dependencies: *hubo-ach, hubo-motion-rt, expat, freeglut3*.

ROS Code

Code repositories that use ros-groovy.

- hubo_init : GUI for initializing Hubo and controlling hubo-ach level settings
 - https://github.com/hubo/hubo_init/
 - * Dependencies: *hubo-ach*
- hubo_walk : GUI for ZMP walk generator
 - https://github.com/hubo/hubo_walk
 - * Dependencies: *hubo-motion-rt, hubomz*
- joystick_integrator : Converts SpaceNav joystick positions to 3D poses
 - https://github.com/a-price/joystick_integrator
- hubo_motion_ros : Provides joint-level and end-effector-level teleoperation, MoveIt-based kinematics server, and temporary ROS trajectory executor.
 - https://github.com/a-price/hubo_motion_ros
 - * Dependencies: *hubo_ros_core, joystick_integrator, drchubo, ros-groovy-control-msgs, ros-groovy-moveit-msgs*

Tasks Overview

Wall Drilling Task

Procedure

1. Identify the drill using robot/human judgement from robot perception feedback.
2. Walk to the drill and pick it up using vision and teleopration from the hubo_maniupaltion_planner panel.
3. Identify wall location using human judgement of robot perception.
4. Walk to the wall and face it using the hubo_walk panel.
5. Begin static balancing using hubo_walk panel.
6. Use hubo_manipulation_planner to identify points on the wall defining polygon to drill out and it's distance away from the wall. Step closer if needed.
7. Send the points in 6-dimensions to the manipulation daemon using the end-effector interpolation mode to drill the object out.
8. Remove drill from wall and place on table or drop.

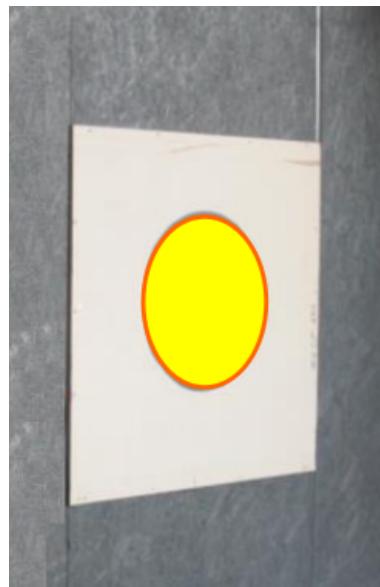


Figure 1: Picture of example target to drill out.

Rubble Clearing Task

Procedure

1. Identify the location of the rubble/door using robot/human judgement from robot perception feedback.
2. Walk to the rubble and face it using the hubo_walk panel.
3. Begin static balancing and squat to height which allows picking up rubble items.
4. Use dual-arm teleoperation to pick up the pieces of rubble and toss them to the side.

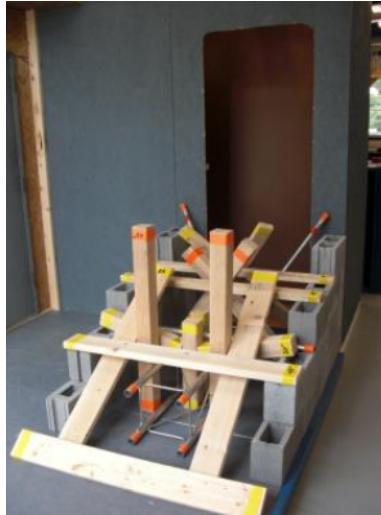


Figure 2: Picture of example rubble scene.

Operation Instructions

This section details how to operate Hubo from the initialization phase to execution of the tasks. The different phases include *Starting-up Hubo*, *Starting-up Perception*, *Walking*, and *Teleop Manipulation*.

Starting-up Hubo

This section describes how to start-up Hubo.

1. Connect to Hubo's computer using ssh.

```
$ ssh hubo@192.168.0.201
```

2. Turn on motor controllers with remote

```
$ ssh sudo service hubo-motion start
```

3. launch RViz

```
$ roscore  
$ rosrun rviz rviz
```

4. Click on the *HuboInitPanel* panel tab.

5. Ensure "IP Address" is correct.

6. Click "Connect" to send and receive ach channels between Hubo's computer and the remote computer.
If it won't connect see the Troubleshooting section.

7. Click "Home All" to home all of Hubo's joints, if not on the ground and not homed. If already homed and standing, continue skip to step.

8. If any active joints do not home, indicated by the color in the *Joint State* section not being gray, click the "Rehome". Button color meanings are

- GRAY: homed
- PURPLE: not homed
- RED: error
- WHITE: inactive

9. If any joints won't home, you can do one of two options:

- (a) Run sudo service hubo-motion stop, run sudo service hubo-motion start, and then start the process over again.
- (b) Click the *Reset* radio button and then click the joint(s) that won't home. Then click the *Home* radio button and click the joint(s) that didn't home. Then press the "Rehome" button again.

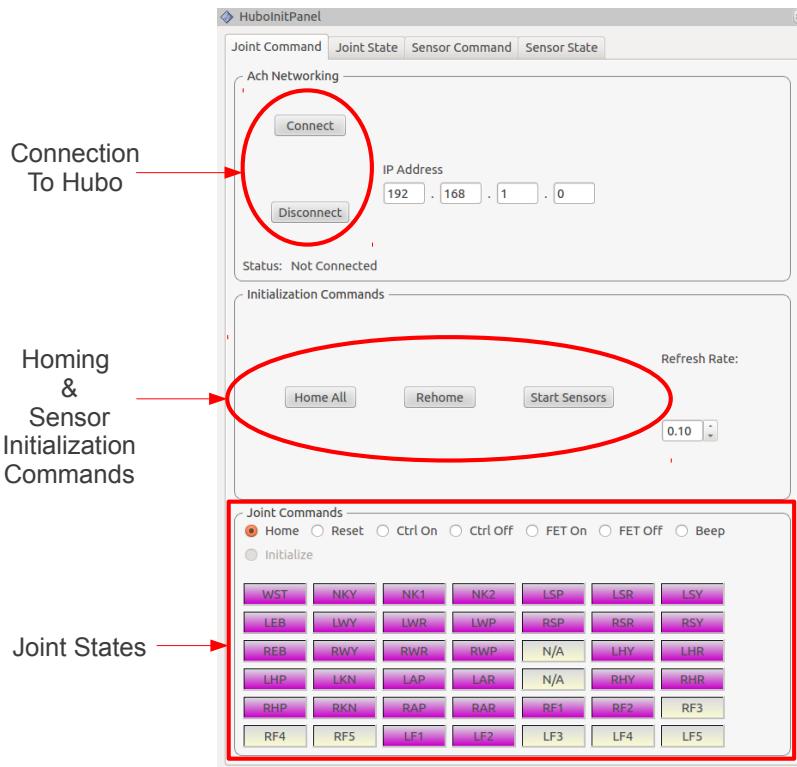


Figure 3: Picture of hubo_init panel.

10. Once the joints are all homed, press the "Start Sensors" button to initialize the IMU and Force/Torque sensors.
11. Click on the *Sensor State* tab and ensure the sensor values are updating and are reasonable. If they aren't, then restart the whole process.

Starting-up Perception

This section is heavily under construction. Generally, what you will need to do is probably something like this:

```
$ roslaunch openni2_launch openni2.launch
```

Or, hopefully when everything gets straightened out, it will be like:

```
$ roslaunch hubo_vision_common full.launch
```

Walking

This section describes how to make Hubo walk. And assumes the robot is already up and running and standing on the ground on two feet.

1. Click on the *HuboWalkPanel* tab.

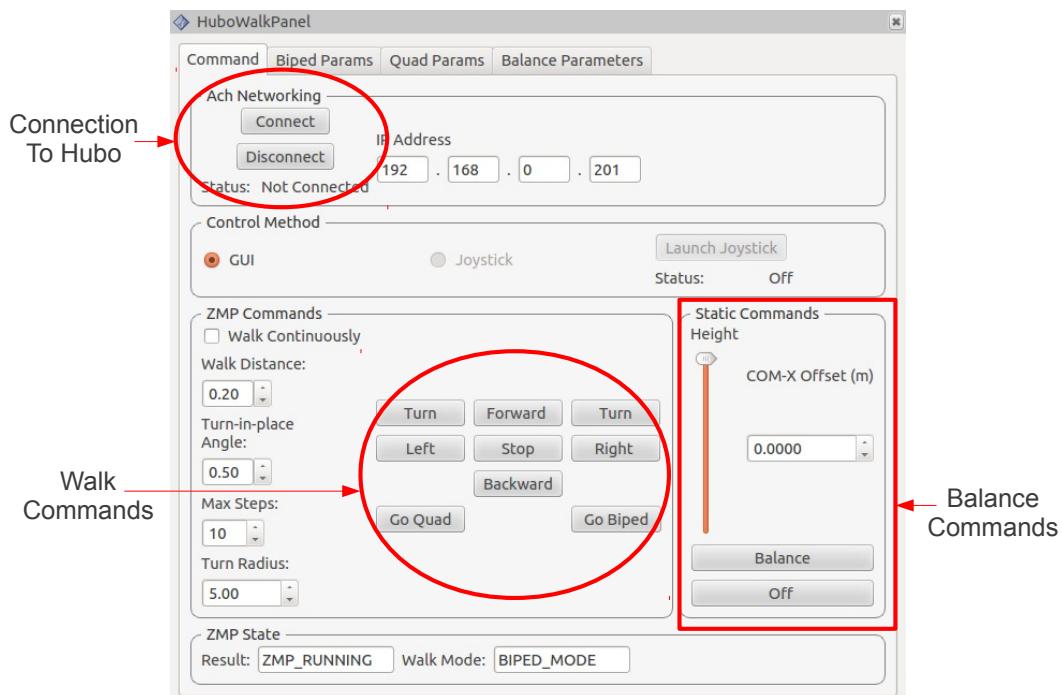


Figure 4: Picture of hubo_walk panel.

2. Ensure the IP Address is correct.
3. Click "Connect" to send and receive ach channels between Hubo's computer and the remote computer. If it won't connect see the Troubleshooting section.
4. Click the *Joystick* radio button if it is not already selected.
5. Based on vision feedback enter desired walk distance, max number of steps, and turn-in-place angle (for turning in place).
6. Click the "Go Quad" button to transition into quadruped mode.
7. Click the walk command desired to get to your desired goal. (Eventually the interface will use a desired location and orientation, and the ZMP walk generator will create a footstep plan to get there.) The *Walk Continuously* checkbox determines whether or not to walk the specified distance or walk until receiving a stop command. The *ZMP State* section at the bottom of the panel display the state of the ZMP walker generator.
8. Click "Go Biped" to get back up on two feet in order to manipulate objects.

9. In the *Balance Commands* section of the panel, click "Balance" to go into static balancing mode.
10. The x-location of the center of mass can be adjusted in the *COM-X Offset (m)* field. Then click the "Balance" button to send the command.
11. The squat height of the robot can be adjusted by sliding the *Height* slider up and down, which automatically sends the command to the robot.

Manipulation

This section describes how to teleoperate Hubo in order to dual-arm manipulate objects in the environment.
NB: First install all software outlined in Code Repositories

1. If you haven't already, start up ROS

```
$ roscore  
$ rosrun rviz rviz
```

2. Start the teleoperation interface

```
$ roslaunch hubo_motion_ros fullbody_teleop.launch
```

3. (Optional) Start up Vision system as discussed in Starting-up Perception.

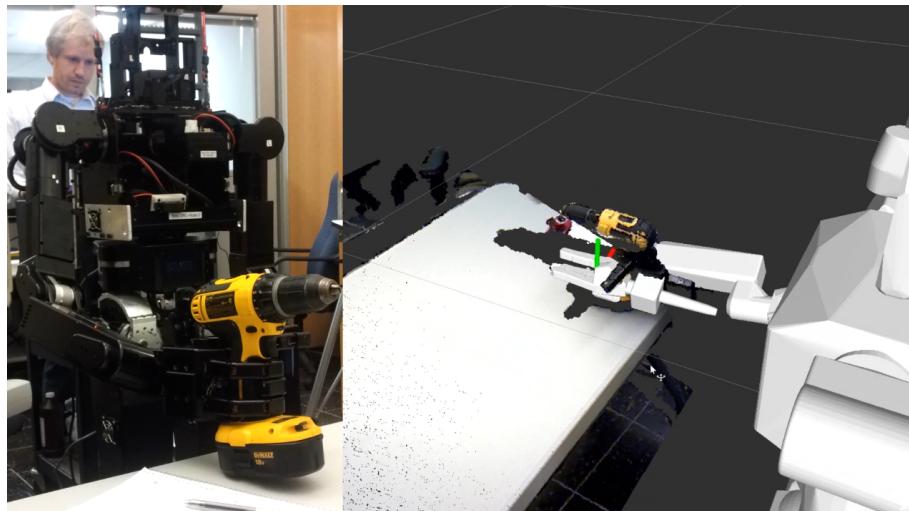


Figure 5: Picture of teleoperation of picking up a drill.

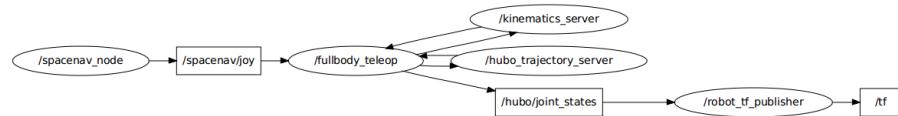


Figure 6: Simplified ROS Graph created by fullbody_teleop.launch.

Troubleshooting

Table 1: Troubleshooting Table

Symptom	Causes	Solution
Can't ssh to Hubo's PC	Not on correct network. Hubo's PC is stuck in the Grub menu	Connect to the correct network. Connect a monitor and keyboard to Hubo's PC and debug, or just a keyboard and press enter.
Can't connect to the robot in RViz panel	Not on the correct network. The daemons are not running on Hubo	Connect to the correct network. Ensure daemons are running by typing <code>ps aux grep daemon</code> or restarting Hubo and running <code>sudo service hubo-motion start</code> .
Receiving ACH_OVERFLOW from rviz command. HuboInitPanel joint names don't showing up correctly.	Don't have matching or most up-to-date installs of hubo-ach, hubo-motion-rt, hubomz or possible another repo.	Make sure you're on the right branch in each repo and reinstall on both machines.
Robot homes only the upper or lower body, or neither.	CAN line got corrupted.	Shutdown and turn off the Hubo PC and turn it back on and try again.