
Documentation of a Domestic Assistant Mobile Manipulator Robot

Alphonsus Adu-Bredu

ALPHONSUS.ADU_BREDU@TUFTS.EDU

Tufts University, 419 Boston Ave., Medford, MA, 02155, USA

Abstract

We present RASCAPP, a mobile manipulator robot for performing domestic activities, a detailed description of its hardware and software systems and finally, a description of the various algorithms it uses in performing domestic tasks.

1. Introduction

RASCAPP (Robotic Assisted Self-Care Activities for People with Paralysis) is a mobile manipulator robot whose upper body is the Baxter robot[1] from Rethink Robotics[2] and whose mobile base is made from a re-purposed electric wheelchair[3]. The electric wheelchair was chosen over an off-the-shelf mobile robot base because, most off-the-shelf mobile robot bases are quite expensive. In this paper, we describe both the software and hardware alterations we made to the electric wheelchair to convert it into a mobile robot base. We also describe how we interfaced the baxter robot with the mobile robot base.

2. Related Work

This work was based largely on prior work by Wen et al. [4]. The hardware setup is almost identical to that of Wen et al. with a few differences in the circuitry and safety mechanisms we developed, and the host of sensor suites we included in our system to make the robot more aware of its environment. The software system architecture and the algorithms we employ are however quite different from the work done by Wen et al.

3. Hardware system

3.1. The mobile robot base

We made the mobile robot base from a re-purposed electric wheelchair [Figure 1]. The wheelchair was first stripped off its seating and metal supports, leaving just the power base [Figure 2]. The motors and the wheelchair battery were then disconnected from the drive module[5] of the wheelchair. The electrical connection of the joystick of the wheelchair to the drive module [Figure 3] was however kept since that was the means through which we would charge the wheelchair series-connected batteries. The drive module, while still connected to the joystick is removed from the power base and set aside. The two motors of the wheelchair as well as the series-connected batteries were connected to a Roboclaw 2x60A DC motor driver[6]. The motor driver is set to the right baudrate(57600) and operation mode(serial mode) and then connected to an arduino-mega over serial through the tx, rx pins on the arduino. The arduino is connected to a USB hub which is connected to an on-board laptop that performs all the processing. This laptop [7] is a Gigabyte Aero 15x laptop with 32GB RAM, 512GB SSD storage, Nvidia GTX 1070 GPU with 8GB RAM, and a core i7 processor. The laptop runs Ubuntu 16.04. and has a battery life of just around 2 hours of constant usage.

A 0.5in thick hard-steel plate[8] is machined and holes for mounting onto the mobile base as well as mounting Baxter onto it are drilled. The CAD files for this plate can be found at the link in [9]. This steel plate is mounted onto the power base as shown in figure 4.

3.2. The backpack

The backpack is an aluminum[10] plate mounted onto the back of Baxter to house all the electronics of the robot system. We cut out a rectangular shape from the aluminum sheet and make mounting holes at vantage points on the sheet. CAD file can be found at the link at [11]. Once the cuts are made, the surface of the cutout sheet is insulated with foam core to prevent



Figure 1. Wheelchair with only its seat stripped off



Figure 2. Wheelchair with metal supports stripped off

the components mounted on it from accidentally short-circuiting due to the conductive nature of aluminum. The backpack is then mounted onto the back of Baxter [Figure 5] using U-bolts[12].

3.3. The circuitry

The electrical system of the robot is split into two parts; the high voltage section and the low voltage section. The high voltage section supplies 24V from the battery of the power base to the electronic components that require 24V power supply and the low voltage section, which is the logic section is responsible for the micro-processing operates predominantly

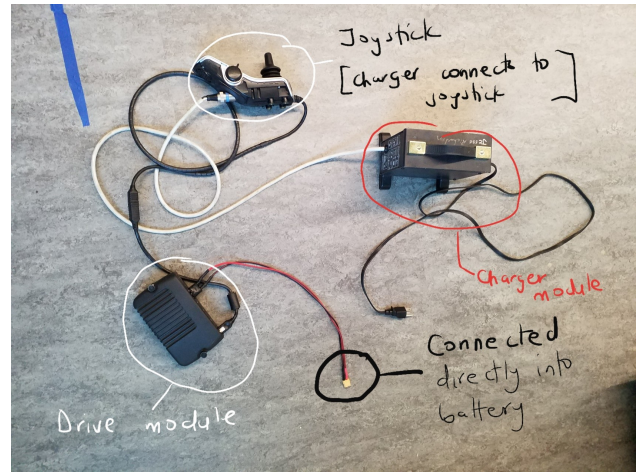


Figure 3. The charging system, featuring the drive module, original wheelchair joystick and the charger module



Figure 4. Hard-steel plate mounted onto power base

on 5V power supply from a power bank. The low voltage section comprises of mainly micro-controllers and sensors while the high voltage section is made up of the motor driver, the inverter and the relays from the kill-switch system. This sectioning of the electrical system is done to protect and isolate the logic components from the unpredictable spikes in current produced by the motors of the power base. A schematic of the low voltage section is in Figure 6 and the high voltage section in Figure 7.

3.4. The kill-switch system

The kill-switch is a safety feature implemented to instantly stop the motors when activated by the user. The user activates the kill-switch from a handheld radio-transmitter controller by pressing the stop button on the controller. Once, the stop button is pressed, an encrypted 915MHz frequency radio signal is sent



Figure 5. Backpack mounted onto the back of Baxter

from the transmitter to the RF receiver module connected to the two Primary relays in Figure 7. The RF receiver module in turn activates both Primary relays, causing them to short-circuit the positive and negative terminals of both motors together. This action instantly stops the motors. Once the continue button is pressed, an encrypted signal is sent to the receiver module which in turn re-connects the motor terminals to the motor driver, causing the power base to resume normal functionality. If the receiver doesn't receive a heartbeat signal[13] from the transmitter in 5 continuous seconds, the motors are short-circuited and the power base stops operating. Once a heartbeat from the remote transmitter is received by the receiver, the power base resumes normal operation.

3.5. The Sensor suite

The sensor suite consists of an ultrasonic range sensor[14] for collision-prevention, a 2D lidar [15] for de-

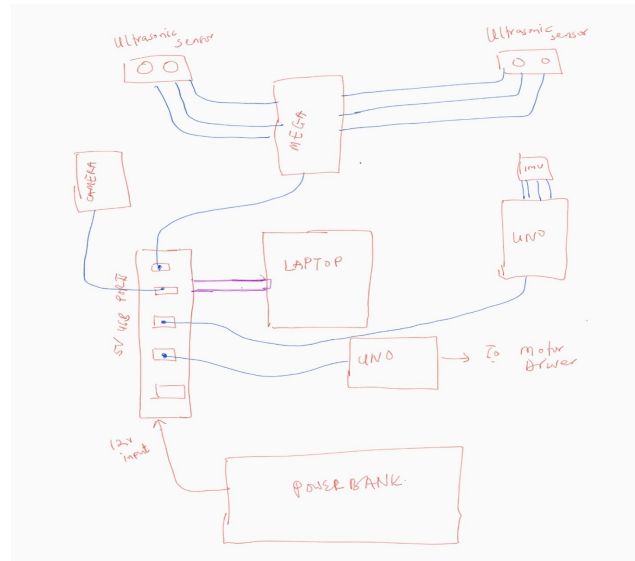


Figure 6. Schematic of the low voltage section

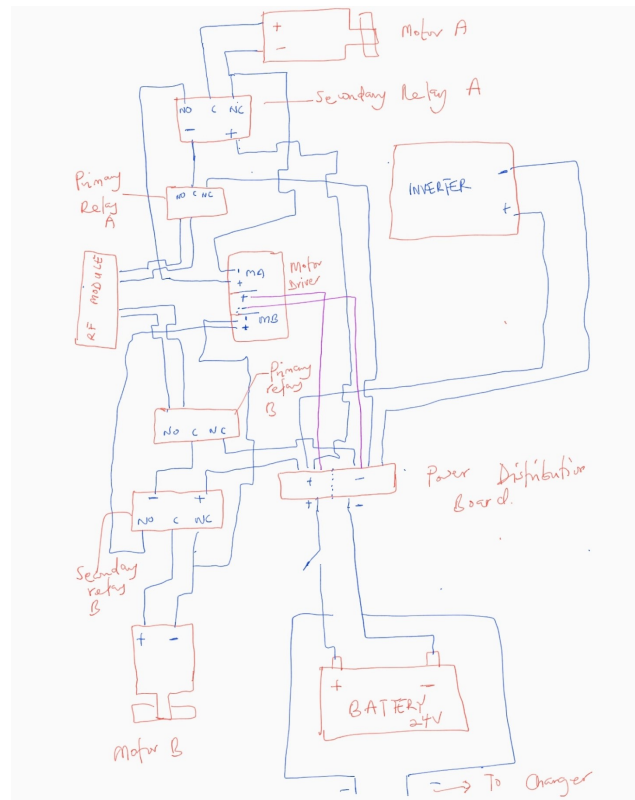


Figure 7. Schematic of the high voltage section

termining the position of objects in the robot's neighborhood, a 3D camera[16] for locating objects of interest in 3D space, a microphone[17] for receiving voice commands from the user and a web camera [18] for providing camera feed at the base of the robot. The

mobile base has wheel encoders on its wheels to measure its traveled distance as well as its local orientation. The robot has an ultrasonic beacon[19] mounted on its head to communicate with the indoor GPS system[19] set up in the lab to enable the robot to localize itself within the lab and know its global x,y position at any point in time. The ultrasonic beacon mounted on the robot's head also has an inbuilt IMU which enables the robot to determine its global orientation.

4. Software system

4.1. The ROS Network

The entire robot system operates on a ROS[20] architecture. The multiple parts of the software system are ROS nodes that communicate to each other by publishing and subscribing to messages on specific ROS topics. The entire system has one ROS master which runs on the onboard laptop[7] and multiple ROS slaves that run on different computers. A schematic of the ROS setup for the entire robot system is in Figure 8.

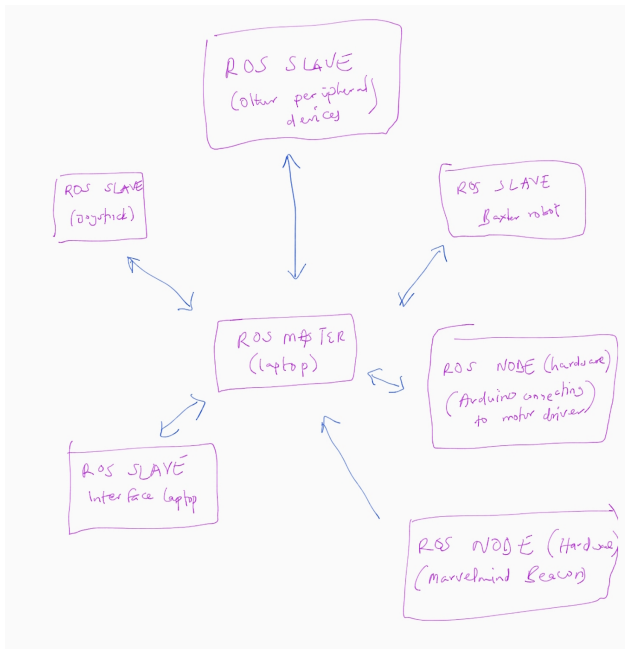


Figure 8. Diagram of the ROS system of the robot

Since the ROS slaves are running on different computers (the joystick runs on a raspberry pi and the Interface runs on a laptop), they would have to be on the same wireless network with static and consistent IP addresses in order to be able to be accessed on the ROS network. Since we are not assured a consistent IP address on the school's WIFI network, we decided to setup a local WIFI connection using a router[21]. This

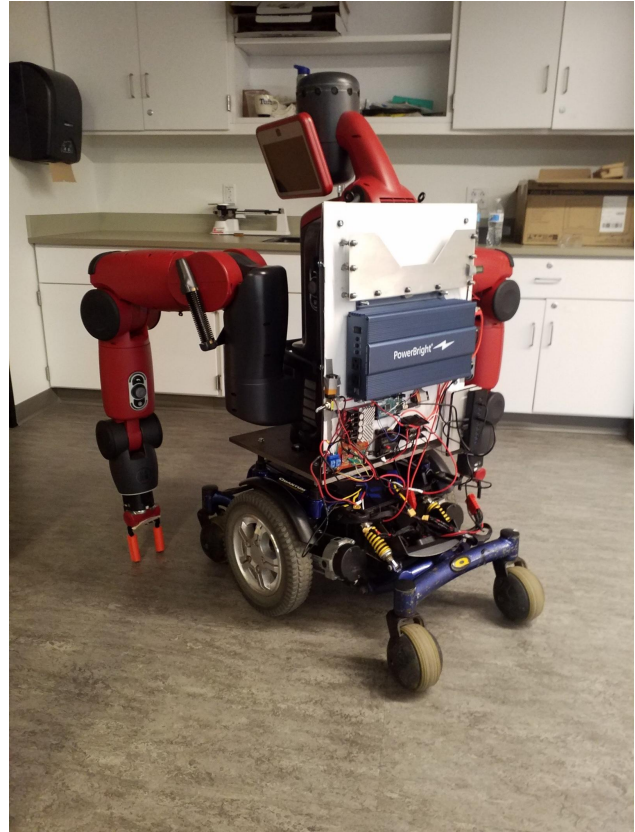


Figure 9. Baxter mounted on mobile base

way all the devices on the network can be assured of a static IP address. Also, this technique shields the robot system from slowing down in case there is traffic on the school's WIFI network. By having a separate WIFI network for the robot and its peripheral devices and not exposing them to the school's WIFI, we protect them from external sabotage. We can easily include other peripheral devices into the network just by manually assigning them IP addresses and logging into the robot's network using a special username and password.

4.2. The Mobile robot base

The mobile base is controlled by publishing Twist messages[22] onto the /cmd_vel ROS topic. The rosserial node which runs on the arduino subscribes to this topic. Once it receives the message, an arduino driver we wrote on the arduino translates the Twist message into velocity commands on the left and right wheels of the power base. These commands are sent to the motor driver[6] which supplies the corresponding currents to the motors to move them accordingly.

Algorithm 1 Kitchen mode algorithm

```

commands array consists of a list of viable commands
repeat
  Input: voice command in English
  for  $i = 1$  to  $\text{length}(\textit{commands})$  do
    if  $\textit{commands}[i] == \text{Input}$  then
      call function corresponding to  $\textit{commands}[i]$ 
      break for loop
    end if
  end for
until program is killed

```

4.3. The Joystick

The joystick system consists of an arcade 4-button joystick, each button sharing a common ground and each button terminal connected to a micro-controller board. The micro-controller is flashed with USB joystick drivers and connected by USB to a raspberry pi. The raspberry pi runs Ubuntu, an Ubuntu distro, and has ROS installed. Turning the joystick knob to a direction sends a directional impulse to the micro-controller which in turn translates the impulse into a joystick signal and sends it over to the raspberry pi. We wrote a ROS joystick node which reads the joystick signal, formulates it into a Twist message and publishes this message onto the /cmd_vel ROS topic.

The ROS node running on the raspberry pi is a ROS slave to the ROS master running on the laptop as depicted in Figure 8.

4.4. The Algorithms

When in the kitchen mode, the robot runs a primary algorithm which works as in [Algorithm 1].

Corresponding command functions [Algorithm 1] also have a general pattern where the robot first gets the 3D pose of the object using its 3D cameras, performs a transformation of the 3D pose to put it in the robot's reference frame and sends this transformed 3D pose to the Inverse Kinematics solver which solves the required joint angles needed to get the robot's end effector to the goal pose (the transformed 3D pose). If there is a valid solution, the robot moves its arm to the goal pose. If there is no valid solution, the robot's arm remains stationary and the robot reports this failure by saying "cannot move my arm to this position". This is summarized in [Algorithm 2].

Videos of RASCAPP performing kitchen tasks can be found in [23].

Algorithm 2 General arm motion algorithm

```

1. 3D camera finds the xyz pose of object
2. xyz pose is transformed into a pose in the robot's reference frame
3. Transformed xyz pose is submitted to the IK solver
if solver == success then
  Robot moves arm to goal pose return
end if
else Robot reports failure
return

```

5. Future Work

Future work would largely be centered on reducing the weight of the robot. It currently weighs 300lb and as such draws a lot of battery power to move around. We also plan on replacing Baxter with a much lighter and more accurate robot arm.

References

[1] Baxter robot. Rethink Robotics Company. URL: <https://www.rethinkrobotics.com/baxter>

[2] Rethink Robotics Company. URL: <https://www.rethinkrobotics.com/>

[3] Quantum Q6000z. Pride mobility. URL: <http://www.pridemobility.com.au/index.cfm/products/quantum-power-chairs/sp/q6000z/>

[4] L. Lu, J.T. Wen (2017). Baxter-On-Wheels (BOW): An Assistive Mobile Manipulator for Mobility Impaired Individuals. in Trends in Control and Decision-Making for HumanRobot Collaboration Systems, Ed. by Y. Wang and F. Zhang, Springer-Verlag, London, U.K., 2017.

[5] Quantum Power Wheelchair Control Module. URL: <https://mobilityequipmentforless.com/products/quantum-power-wheelchair-control-module-ctldc1540-1740-2017/>

[6] RoboClaw 2x60A Motor Controller(V5). URL: <https://www.pololu.com/product/1499>

[7] Gigabyte AERO 15X laptop. URL: <https://www.gigabyte.com/Laptop/AERO-15X--i7-8750H#kf>

[8] Hard Steel Plate. URL: <https://www.mcmaster.com/catalog/125/3845>

[9] Baseplate CAD file. URL: https://github.com/alphonsusadubredu/rascapp_robot/blob/master/baseplate.dxf

[10] Aluminum Plate. URL: <https://www.mcmaster.com/>

com/89015k48

[11] Backpack CAD file. URL: https://github.com/alphonsusadubredu/rascapp_robot/blob/master/backpack.dxf

[12] U-Bolts. URL: <https://www.mcmaster.com/catalog/125/1635>

[13] Aguilera, Marcos Kawazoe; Chen, Wei; Toueg, Sam. "Heartbeat: A Timeout-Free Failure Detector for Quiescent Reliable Communication". Distributed Algorithms. Springer Berlin Heidelberg. Retrieved 4 March 2015. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/1997/09/wdag97_hb.pdf

[14] Ping Ultrasonic Sensor. URL: <https://www.parallax.com/product/28015>

[15] RPLidar A1 360^O Laser Range Scanner. URL: <http://www.slamtec.com/en/lidar/a1>

[16] Primesense Carmine 1.09. URL: <http://xtionprolive.com/primesense-carmine-1.09>

[17] Pyle Pro PUSBMIC52 - Wireless Heaset USB Microphone System. URL: https://www.bhphotovideo.com/c/product/1315634-REG/pyle_pro_pushmic52_uhf_wireless_usb_microphone.html?ap=y&gclid=Cj0KCQiApvbhBRDXARIsALnNoK335qdEk4JmuZL1VFYLBmsvB7pUXbk08oB4vYWT1RXB3Rs_S-yJ3IaAqd8EALw_wcB&lsft=BI%3A514&smp=Y

[18] Logitech C920 HD Pro Webcam. URL: <https://www.logitech.com/en-us/product/hd-pro-webcam-c920?crid=34>

[19] Marvelmind Indoor GPS system. URL: <https://marvelmind.com/product/starter-set-hw-v4-9-imu-plastic-housing/>

[20] Robot Operating System. URL: www.ros.org

[21] NetGear AC1000 Dual Band Smart WiFi Router. URL: https://www.amazon.com/gp/product/B071J24FNY/ref=ppx_yo_dt_b_asin_title_o08_o00_s00?ie=UTF8&psc=1

[22] Twist Message. URL: http://docs.ros.org/api/geometry_msgs/html/msg/Twist.html

[23] RASCAPP videos. URL: <http://alphonsusadubredu.com/videos/>