Carnegie Mellon University

16-681A

MRSD Project I

Individual Lab Report 03
Team C: COBORG

Author: Gerald D'Ascoli

Remaining Team C Members:
Jonathan Lord-Fonda | Yuqing Qin | Husam Wadi | Feng Xiang

Sponsor:
Biorobotics Lab

March 18, 2021

Table of Contents

1 Individual Progress

For this progress review, I finished up the main functionality of the voice subsystem and integrated it as a ROS node into the main system. I also worked with Jonathan to develop the main_state_machine node for the overall ROS structure so that the voice recognition node had something to communicate with for testing.

The voice recognition node communicates with our ROS system as purely a publishing node. It listens through the system audio input, processes possible commands, and publishes to "/voice_commands" when it receives a valid command. This publishing structure had to be designed to publish asynchronously whenever a valid command was recognized so the voice_recog node could not be structured like standard ROS node examples that used ros.spin(). This structure is implemented in conjunction with audio feedback from the speaker so that the user can be told whether or not a command is being processed by the system. This feedback element of the design proved necessary when testing yielded users saying commands repeatedly and at times when the system could not properly listen. Audio feedback notifying the user when the keyword was recognized so that it is listening for a command and when a command was either recognized or missed encourages proper verbal command structure.

The main_state_machine node handles the internal system commands controlling the current actions of the robot, including states for functions hold object, compact (move to home), e-stop to fully cut the power to the motors, or caution freeze the arm in place. This e-stop is our software e-stop that issues signals to the motors to cut off, we have other hardware e-stop designs in mind for events when the software is hung up. The main_state_machine node controls function of the other subsystems through publishing to the "/state_output" topic which will be read by other nodes as a main controller signal. Jonathan developed the structure and pseudocode for the main_state_machine ROS node, I helped by writing the code in python and debugging python and ROS issues.

Once both nodes were compiled and functional, we tested the communication between the voice_recog node and the main_state_machine node by using "rosrun" to run each node's script. We listened in on the voice_recog node using "rostopic echo /voice_command" to confirm that it was successfully publishing recognized voice commands to the topic. We then confirmed that the main_state_machine node was receiving and processing the commands by listening to the "/state_output" with rostopic echo. This is our first integrated functionality for our system, so it was a big step forward towards getting a full demonstration up and running.

I have also been working on the custom CoBorg PCB. I worked with Husam to develop the conceptual design. Conceptual design show in Figure 1 below. Since then, I have taken over for part selection, finding library files, and designing the schematic on EagleCAD. I came up with the idea to design a power distribution board that will be able to switch power to the CoBorg between an AC wall outlet and our on-board batteries. This would be able to operate untethered on battery power per our system requirement, but also be able to function plugged in while also recharging the batteries. Downstream of the power supply switching circuit, the power flows into the motors at the original 36VDC, and also into a ~19.5VDC switching regulator and a 5VDC switching regulator. The circuit will be protected with overcurrent fuses and overvoltage TVSs to ensure safety of electrical components. There will also be indictor LEDs that show that power is present out of each regulator.

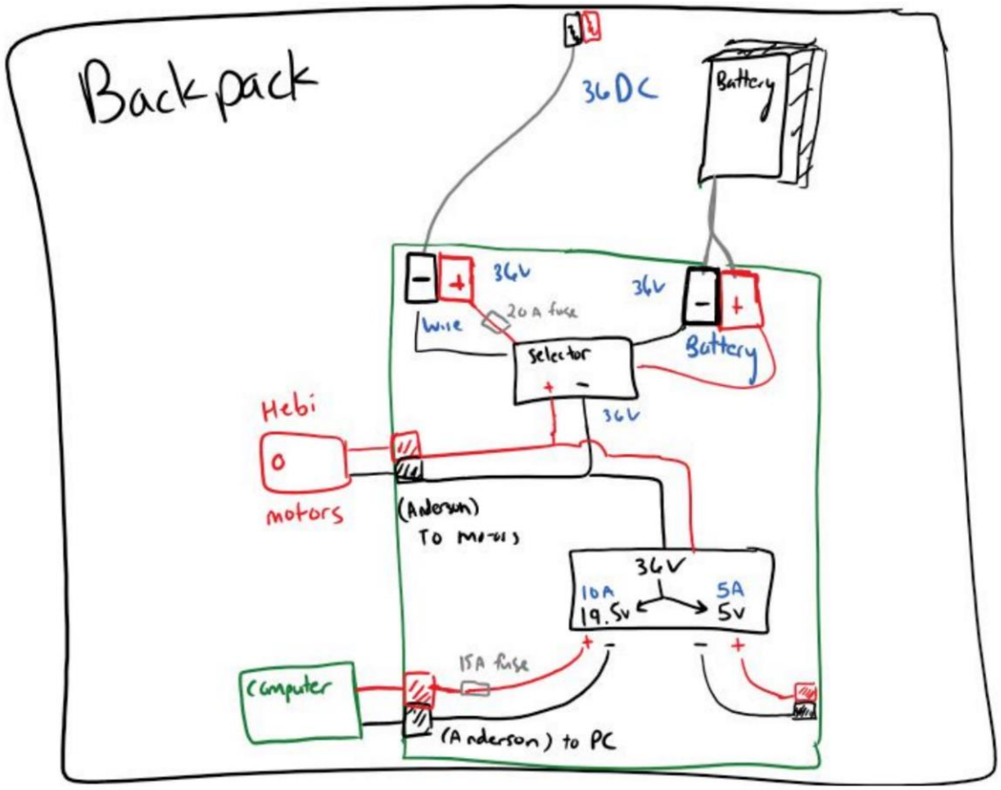

Figure 1. Conceptual Design for CoBorg Power Distribution PCB

2 Challenges

After integrating the voice_recog node with the main_state_machine node, we ended up with a real software framework structure that was now susceptible to breaking. This is exactly a challenge that arose when we tried to test integrating other subsystems on our Zotac computer: integrating the other system broke our functional ROS interaction between our initial nodes. This took about an hour of debugging to work out balancing the required packages. Moving forward, we are going to have to run functionality tests on previously standing frameworks after updates or further system integration to ensure it all remains functional.

In designing the concept for the PCB, we had some challenges establishing the scope of the design. We debated what parts should be external and what functions the PCB should serve. The discussion was mostly centered around if the board should have an on-board AC-DC converter and what kind of cables and input ports will be needed. We agreed to unmount the AC-DC converter and battery charger to keep as an external cable. Within the scope of the PCB, we decided we did not necessarily have to decide what kind of cabling and housing connectors would be required for this, only the connectors on-board had to be established. Additionally, I had some trouble finding a circuit design that would allow continuous power with switching sources. It was not necessarily a matter of complexity, just a lack of resources and examples that show how to accomplish this. Luckily, in trying to find an instructional resource, I stumbled onto the LTV4412 chip that accomplished this function for me with only the need of one external P-channel MOSFET.

3 Teamwork

Team Member	Teamwork Progress	Challenges	Future Work
Feng Xiang	-Tied T265 and D435 cameras to URDF model -Able to move URDF model live relative to global odom frame in RViz simulator	-Develop a functional and robust stabilization model so robot arm can maintain global position through time	-Move robot arm relative goal pose to D435 camera link frame -Implement OMPL+CHOMP motion planner for a more iterative updating motion planning model -Develop 3D goal pose to update stabilizer robot arm relative to global odom frame
Jonathan Lord-Fonda	-Connected and implemented main_state_machine node with voice_recog node -Semantically wrote out all nodes -Updated ROS Node Map with proposal -Worked with Jason on Actuated Manipulation -Read about Elastic Bands -Finished setting up Linux and personal ROS -Met with Kelvin to review ROS Node Map	-Manipulators are far more challenging than mobile robot bases. The original Elastic Bands was for mobile robot bases. -ROS bugs	-Research the other Elastic Bands papers -Research and implement CHOMP (or Elastic Bands) with Jason -Update Main State Machine node -Add the speaker to requirements, validation, etc. -Run through requirements and validation plans, updating them and checking in with each subgroup to see how well we're matching up
Gerry D'Ascoli	-Integrated new microphone for improved single user voice recognition -Helped Jonathan develop the main_state_machine node -Developed voice_recog node ROS wrapper for voice recognition to feed commands to main_state_node -Tested successful communication and proper functionality between the voice_recog node and main_state_node -Updated website -Developed conceptual design for CoBorg PCB with Husam	-Finding stocked parts & .lbr files for CoBorg PCB	- Designing schematic & PCB Layout for CoBorg PCB -Order parts for CoBorg PCB -Run accuracy tests on voice recognition systems to validate requirements are met

Yuqing Qin	- Implement 3D YOLO in ROS - Output the 3D position of multiple hands - Combine D435i and T265 launch file - Postprocess the average 3D position (goal_getter node)	- Make sure the accuracy of 3D position	- Further improve the accuracy of averaged 3d position - Calibrate the depth camera - post-process depth image (add filtering)
Husam Wadi	-Assisted with ROS main node development -Assisted Jason with T265 tracking camera output -Ensured team used Github	-Finding the best path planning algorithm in Move-it for our robot motion. Comparing RRT-Connect with Chomp	-Deep dive testing with robot arm + Realsense cameras to discover the link between them. -Continue progress on goal state publisher

4 Plans

I plan on running tests to get real numerical statistics on the accuracy of the voice recognition subsystem. These statistics are necessary to confirm that the system in functioning with the requirements' performance parameters. These tests will involve attempting to give valid commands to the CoBorg system and taking binary results of successful recognition versus failure. The overall command recognition must be over 60% so these tests will confirm if the system is up to the requirements or if more tuning is necessary.

I plan on finishing the PCB in accordance with the upcoming deadlines for it, then building and integrating it on the CoBorg system. This will require finishing up the schematic, then designing a PCB layout that can be fit in and be securely mount to our current 10" by 7" by 4" enclosure. I want this PCB layout design to be versatile enough that it will not need to be re-manufactured for potential changes in the enclosure or other hardware on the CoBorg system.

I plan on assisting Feng with the actuated manipulation subsystem as that has proven to be our most critical path in development for this semester. This would entail helping develop functionality for the robot arm to articulate to a given point and update said destination point based on movement from our localizing tracking camera. I will also be helping Jonathan with integration and validation by assisting with ROS development and diagnostic testing to verify requirements are met by the subsystems.