Carnegie Mellon University

16-682

MRSD Project II

Individual Lab Report 06
Team C: COBORG

Author: Gerald D'Ascoli

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

Sponsor:
Biorobotics Lab

September 16, 2021

Table of Contents

1 Individual Progress

For this progress review, our main focus was migrating our system from the Zotac computer to the Nvidia Jetson Xavier and getting our SVD encore demo to function on the new system. I assisted the process of setting up the Xavier with initial Xavier setup and installing system dependencies for the voice subsystem. In this process, I took the chance to update functionality, features, and documentation of the voice subsystem based on recommendations of John Dolan from SVD. We also discussed plans for this semester and the work potential new features would entail. As part of these plans, we are most likely going to be adding one or two new HEBI motor(s) to the Coborg arm so as the lead on the electrical system I had to evaluate our current electrical system to determine if it would meet the new power draw requirements or if a new PCB V2.0 would be needed. As the new lead on the object avoidance functionality, I discussed the current state of the planning system with my teammates and started research on potential algorithms and software solutions that could drive this functionality.

To set up the Xavier, I had to flash the Jetpack OS to the Xavier using the Nvidia developer software on a separate Linux system. I hit several difficult challenges in this seemingly simple process that will be discussed later. We later found out that the Jetpack OS installed was insufficient for our needs, so we had to re-install a different version of the OS. Once the Xiaver was up and running, we downloaded the Coborg GitHub repository, but the subsystems still required installation of all of the various dependencies. As the voice subsystem lead, I took over and installed all of the dependent packages and repositories. I also took this opportunity to edit the voice subsystem based on recommendations from SVD. I changed the voice subsystem trigger keyword from “Coborg” to “Hey Coborg” to compensate for some of the comments saying that the trigger word was too sensitive and would trigger at unprompted times. I added a timeout functionality based on Dr. Dolan’s recommendation so now once the keyword is recognized and the system goes to command recognition mode, if there is no recognized command within 5 seconds, it times out and goes back to keyword recognition mode with the standard failure sound feedback.

The plan to add one or more degree(s) of freedom to the Coborg arm raises the power draw to the motors from the electrical subsystem. Since the power distribution PCB was initially designed with only 3 motors in mind, the PCB had to be re-analyzed to decide if a new PCB V2.0 would need to be designed and manufactured.

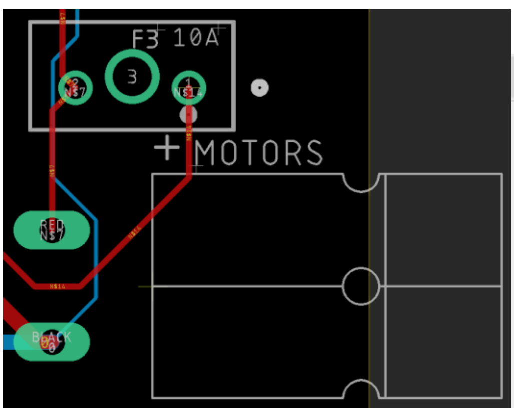

Figure 1a. Copper Traces to “Motors” Output = 16mils

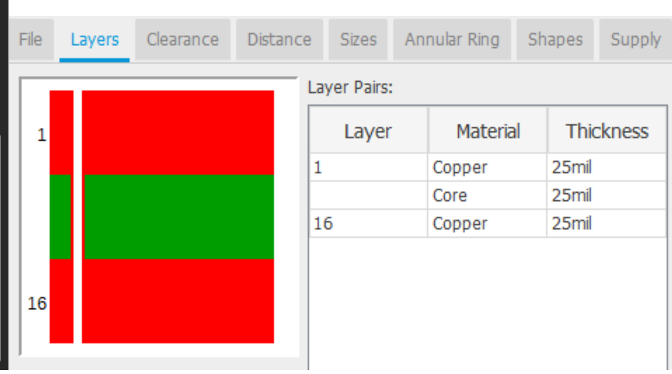

Figure 1b. Copper Layer Thickness of PCB = 25 mils

Motor	Type	Peak Current Draw @ 36V	Cont. Current Draw @ 36V	
1	X8-16	3.0A	1.3A	NEW
2	X5-9	1.6A	0.5A	Optional
3	X5-9	1.6A	0.5A	
4	X5-9	1.6A	0.5A	
5	X5-4	1.6A	0.5A	
Total:		9.4A	3.3A	

Table 1. Current Draw of 4-5 Motors (amperage based on HEBI X-series datasheet)

Based on the calculations shown in Table 1, the peak current draw of the 5-motor arm would be 9.4A if all motors were at peak draw (which is highly unlikely to occur) while the more relevant continuous current draw would be 3.3A. The 16mils x 25mils trace shown in Figure 1a & 1b feeding the “Motors” output of the PCB can support ~11A (traces from input to this output support much higher) so there is a 10A fuse for this output to make sure this limit is not reached for the safety of the traces. Both the calculated peak and continuous currents of the 5 HEBI motors are lower than this limit so the current PCB design can continue to be used for the higher DOF arm. We are still discussing redesigning the board anyway for risk mitigation and aesthetic/form factor changes.

As the new technical lead on object avoidance, I began researching methods of object detection and replanning that would be necessary for this functionality. So far, I have looked into implementations of Octomap integrated with Move-It for this solution. Future steps will be discussed in the future work section.

2 Challenges

The major challenge of this progress review stemmed from problems flashing the OS onto the Jetson Xavier. Initially, we tried to flash the OS from our current Coborg computer using the Nvidia Developer Jetpack software, but it wouldn’t work because the flash required a local system “sudo apt update” command to run successfully and there was a package error on the local Coborg computer that prevented that from running. Then, I tried to run the flash software from the ubuntu instance on my laptop, but I hit a wall when I discovered the OS flash software required more storage space than my ubuntu partition had available. To get around this issue, I tried storing the OS flash files on external drives, which worked for the 1+ hour long download, but failed during the install because of permission issues that require the OS files to be on the local system primary storage. The next step was to backup all of my files on my ubuntu partition, wipe ubuntu entirely, expand the partition from 20GB to 100GB on my laptop, then reinstall the ubuntu partition. On the new expanded partition, the 2-hour long OS download and install was success (after 2 more failures that just required a restart). Unfortunately, we later discovered that due to dependencies we needed an earlier version of the OS and Nvidia tools so we needed to re-flash the OS. This time the space required was ~50GB and for reasons still unknown to me, the Nvidia Developer Jetpack software said that this 50GB of data wouldn’t fit in my available 67GB of space (confirmed by df -h). At this point, it was easier to solve the “sudo apt update” issue on the original Coborg computer, so that package issue was resolved by Yuqing.

3 Teamwork

Feng Xiang installed and configured actuation manipulation nodes into the Xavier Jetson; created URDFs or proposed robot arms and performed simulated SVD testing through Rviz; and created arms out of cardboard in preparation of hands-on testing of task space testing. Some challenges he is looking to face is we may need to tweak the design of the robot arm to be able to meet task space requirements which will require collaborating closely with the vision subsystem as improvements to the camera's field-of-view are being made. He plans to build and finalize a trade study for the proposed robot arm designs; work with Jonathon to create a plan for mechanical system improvements to solve workspace issues; and develop the CAD and BoM for the new robot arm.

Jonathan wrote new task space requirements; created a program to check mechanical workspace against task space; and drew out our fall schedule. A challenge he realized is it appears that our current system will not sufficiently cover our task space, so potentially significant system redesigns may be required. He plans to finalize mechanical workspace program; work with Jason to create a plan for mechanical system improvements to solve work space issues; check the visual work space against the arm's task space; evaluate potential improvements of the smart manipulation program; and debug the smart manipulation code.

Yuqing migrated the vision subsystem to the Xavier; integrated the subsystems together to recreate the SVD demo on the Xavier; cleaned up her vision subsystem nodes/code; and optimized the vision pipeline to extract 3D position internally therefore removing the need for the "goal getter" node. She plans to work with Jonathan to check vision system task space; adjust and test various camera angles for optimal performance; and test the performance of two YOLO v4 tiny instances running on the system for two separate cameras.

Husam created the CAD design for Coborg V2 hardware; prototyped the camera mounting position to implement within CAD design; procured parts to order for new hardware; and created & populated the project management Kanban Board for Fall 2021. He faced challenges because the CAD design requires iteration as the first draft and parts may not work correctly which will take time that the project can't spare. He plans to print the 3D CAD designed parts; integrate and test the new Coborg V2 hardware; and, if necessary, iterate on 3D printed and ordered parts.

4 Plans

The next step for the obstacle avoidance function is to develop and test Octomap for object detection. Due to limitations of system resources, I need to get a demo of Octomap up and running to determine if it can run with high enough resolution and a high enough framerate to function in the Coborg design. This functionality is primarily going to be used to detect and avoid the user's arm, but a demo will also be useful to explore usable depth accuracy to potentially use this function for general task space obstacle avoidance.