Carnegie Mellon University

16-682

MRSD Project II

Individual Lab Report 09
Team C: COBORG

Author: Gerald D'Ascoli

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

Sponsor:
Biorobotics Lab

October 28, 2021

Table of Contents

1 Individual Progress 3
2 Challenges 4
3 Teamwork 5
4 Plans 6

1 Individual Progress

This progress review saw many efforts that I have lead reach some form of fruition that was able to be demonstrated. I primarily worked on finalizing the initial build of resolved rate stabilization and supporting the hardware upgrade effort.

For the stabilization task, I worked with Jason Xiang to get an initial test demo of the functionality based on an unmounted T265 tracking camera to move while the Coborg was stationary mounted on the rack. There was a lot of debugging to do once we got the code to finally compile, mostly involving the tf module used to transform the goal position and current end effector position into the t265_odom “global” frame. Once we got the tf frame bugs figured out, we were able to control the end-effector movement in cartesian space by moving the unmounted T265 camera in corresponding cartesian space. Unfortunately, this control was mostly opposite of what we needed, in that moving the camera forward in turn moved the arm forward; similar behavior right and left. This adds to the displacement instead of counteracts it. Oddly, the movement was properly inverted in the z-direction (up and down) meaning that when the camera went up, the arm went down. After some analysis, we realized that we had flipped the sources and destinations of the tf transforms in the code. Once we switched them back and flipped the sign on the z transform, the end-effector had proper counteractive motion relating to the motion of the T265 camera. For this demo, we had been using a fake, static goal position, so the next step was to integrate the stabilization with an actual goal position set by the new and improved vision-based goal-getter node. Once integrated, we several problems arise. The increased computation introduced a delay between iterations of resolved rate which caused very jerky movement and a delay before motion recognition. Our next steps with this functionality is to integrate it into the full actuated manipulation so that Move-It can plan to the goal then resolved rate stabilization will maintain the goal position from there. We also need to resolve the latency and jerky motion issue as best as we can.

The hardware upgrade has been progressing very nicely in the short amount of time since we finally had our parts delivered. Husam has been pulling long hours assembling the mechanical elements so I helped out by taking electrical upgrades off of his plate. So far, I’ve unwired and rewired the arm several times to measure out power cables with tighter tolerances for less interference of arm motion and larger gauges for increased current added by the new fourth motor. These cables are shown in Figure 1 with emphasis on the gauge of the cable and the tightness of the routing. The Ethernet cables are temporary as we had challenges building the custom length cables before this Progress Review. I’ve supported the mechanical build when possible by helping Husam with painting as much as possible, adding the backpack straps to the finished frame, and mounting 3D printed components to the frame. The full hardware upgrade should be finalized by this weekend following this Progress Review to allow for better integration and testing of the software on the final system.

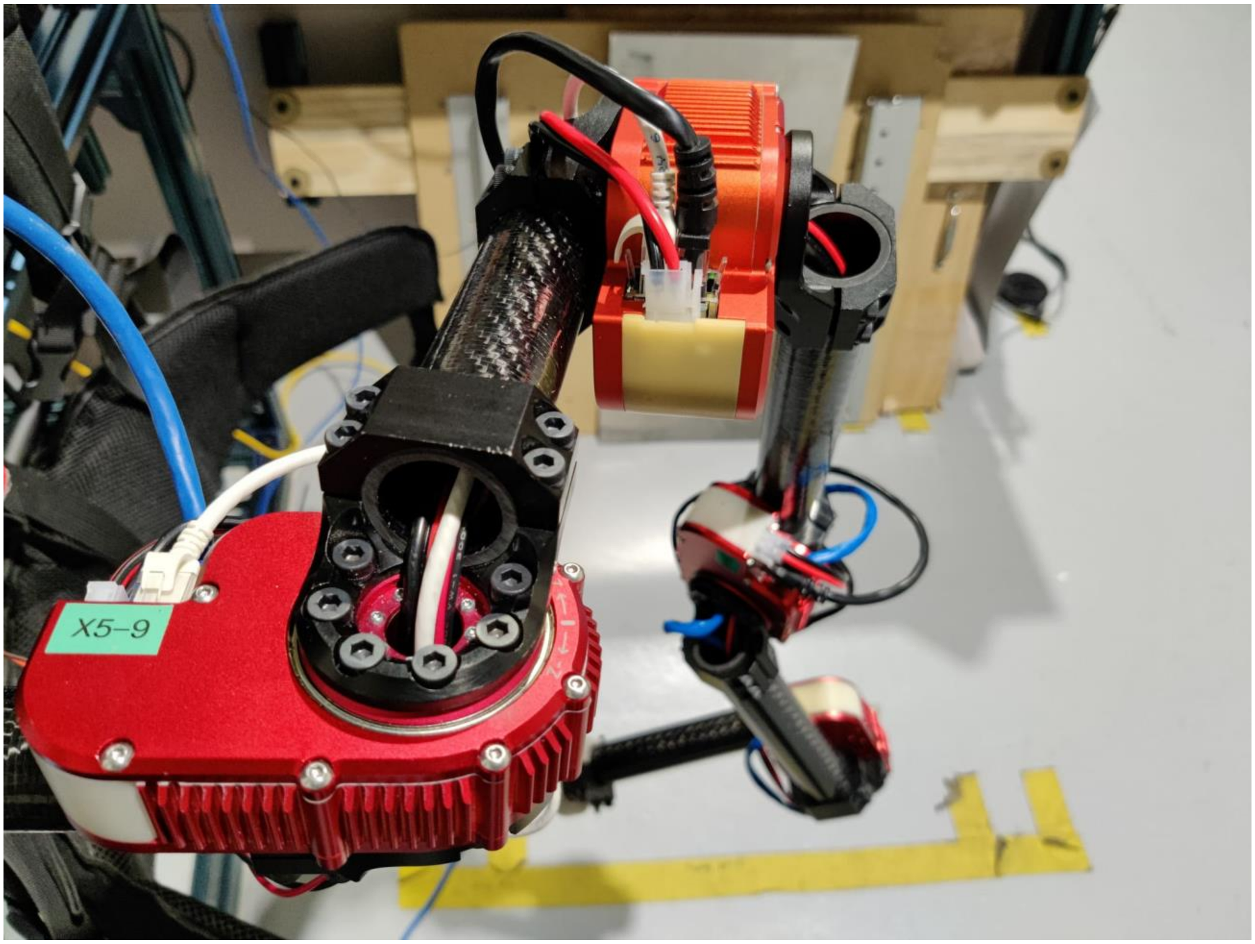

Figure 1. New Power Cable Routing

2 Challenges

Resolved rate stabilization posed many significant challenges in development as discussed in the previous section. There are still many challenges to be faced regarding integration with other nodes. When the user travels too far and the goal moves out of the task space of the arm, a strange behavior arises where resolved rate continually generates significant joint angle changes in an attempt to reach the unreachable goal causing very erratic flailing of the arm. We hope this can be fixed by putting limits on goal values when in stabilization so resolved rate either gives up or maintains a realizable goal when the goal is moved out of range. We are also facing issues with technical inaccuracies of the T265 camera. When the camera is moved, the internal visual slam has significant drift, causing sometimes significant movement of the goal in global space where it should be stationary. When the arm “stabilizes” to this point, it is not in the originally intended goal position and could therefore drop the held part in the use case. Now that the resolved rate stabilization is being integrated with the other subsystems, there is a significant computational reduction causing delays in recognition of user motion and counteractive arm actuation. This delay can cause the use case to fail if a held part to be dropped because the arm does not move fast enough to compensate for the user motion or if the part gets damaged by the jerkiness of the motion due to the delay in computation. Essentially, resolved rate has been a recent addition to the full integrated use case and therefore faces many bugs and challenges that must be resolved.

Another problem that arose in software integration is issues with obstacle avoidance potentially causing Jonathan's path stitching functionality to crash. For his development, we disabled the obstacle avoidance to remove it as a factor, but for both functionalities to be implement, we must face the challenges that may arise in integration.

Some challenges arose in the electrical side of the hardware upgrade. One major challenge was an issue with the parts we had available for building the custom cat6 network cables. The cable we were cutting and building off of did not have proper labelling on the individual wires so it would have taken significant effort to identify each wire and build each cable. To rectify, we purchased cat6 cable designed to be cut and crimped instead of cutting the pre-made wires we already had. We also had a challenge arise when running the Nvidia Jetson Xiaver untethered powered from our batteries. Something about this power supply system has reliability issues which cause the Jetson to random power cycle. This is obviously a critical issue as a random power cycle would kill our use case. This problem has arisen for other teams using the Jetson Xavier as well. We have discussed a solution involving potentially adding a capacitor bank for the computer power supply output of our power distribution PCB in order to stabilize that charge with an increased transience. This power issue will be further researched to better diagnose the issue, then we will determine a valid solution.

3 Teamwork

For this Progress Review, Feng worked with various team members to help finalize their independent functionality. He working with Yuqing on the goal-getter node, me with resolved rate stabilization, and Jonathan with the smart manipulation. He faced challenges helping resolve critical edge cases for each aforementioned system that could each pose significant risk to the use case. He also faced challenges with failed or improper booting of the realsense cameras, an issue that would definitely crash our entire use case. He plans to work on integration of the various independent functionalities into the overall Coborg system use case. He then plans to test and validate to our requirements.

Jonathan spent this Progress Review debugging smart manipulation in both the full state machine and the custom path stitching functionality as well as discussing changes to the software pipeline with the team. He faced challenges debugging path stitching which took a lot of time and resulted in the functionality's exclusion from the system (pending further developments). He plans to finalize the smart manipulation integration and tuning; update the main state machine to match the new software pipeline; continue work on the path stitching; build up the testing structure for FVD; and test the integrated system.

Over the past few weeks, Yuqing worked with Jason to develop the goal-getter node and worked with me and Jonathan to integrate it with other systems. She faced challenges dealing with tf transforms in development of the goal-getter node; dealing with constant changes with the goal-getter node as downstream systems change in debugging for integration; and figuring out optimizations for the goal-getter node so it takes less time and computational resources. She plans to continue integration of goal-getter with other systems; optimize the goal-getter node; measure and recalibrate the camera frames for the URDF model based on the new hardware setup; and test the Coborg system.

Since the parts were finally delivered, Husam has worked hard on the hardware framework. This includes the assembly of the robot frame, the 3D printing (and reprinting) of parts, the spray painting of parts to match the overall theme, and the assistance in wire assembly diagnosis with me. He also handled the procurement of parts to mitigate risk for FVD. He has faced challenges with the amount of time hardware assembly has been taking as well as problems that have arisen with mistakes from trying to rush the process. He plans to finalize the hardware build, integrate the new e-stop system, mount the electrical components mechanically to the Coborg, and assist in testing for the FVD.

4 Plans

As a team, we have pivoted focus towards integration of the various functionalities we have each been developing. Our goal for Progress Review 11 is to have a dry run for our Fall Validation Demonstration so the integration must be in a functional state by then. My plans are to help with this integration process with debugging and testing, particularly regarding integration of the functionalities that I have taken the lead role on: resolved rate stabilization and obstacle avoidance.

In addition to the software integration, the hardware must be assembled and functional by then. This requires me to build and install the new networking cables and route all remaining power and signal cables. The remaining power cables include all peripherals to the Coborg Power Distribution PCB such as power to the Ethernet switch, power to the Jetson, and power to the PCI-e slot USB expansion on the Jetson. We already have other cables required for USB communication and power to all three cameras.

A major problem that arose during integration is the power reliability issue with the Nvidia Jetson supplied by the battery supply. I plan to research and test the problem to better diagnose the issue, then develop a solution to ensure power reliability to the Jetson and prevent any unexpected power cycles of the Coborg’s main computer when untethered.