Individual Lab Report - Progress Review 3

Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Parker Hill

Team C:

Parker Hill | Kaushik Balasundar | Anthony Kyu

Sundaram Seivur | Gunjan Sethi

March 24th, 2022

Carnegie Mellon University

The Robotics Institute

Contents

1 Individual Progress

1.1 Reaming End-Effector

The main thing I worked on for this past progress review was finishing the design, 3D printing, and assembling of the entire reaming end-effector structure, such that it could be attached to the robot arm in time for the progress review. This required a decent amount of work, as previously the most I had done is prove that the reaming handle could be held in place utilizing the grooves. I started by making a motor adapter which can house the motor as well as attach to a force torque sensor closer to the end-effector, and attach to the long plate which is necessary to interface with the reaming handle. I then redesigned the plate for the reaming handle to interface with this part, and finally designed an adapter which allows the force-torque sensor to attach to the robot arm itself. The finalized design looked as such in Solidworks in figure 1, with the motor housing made transparent to see the motor inside.

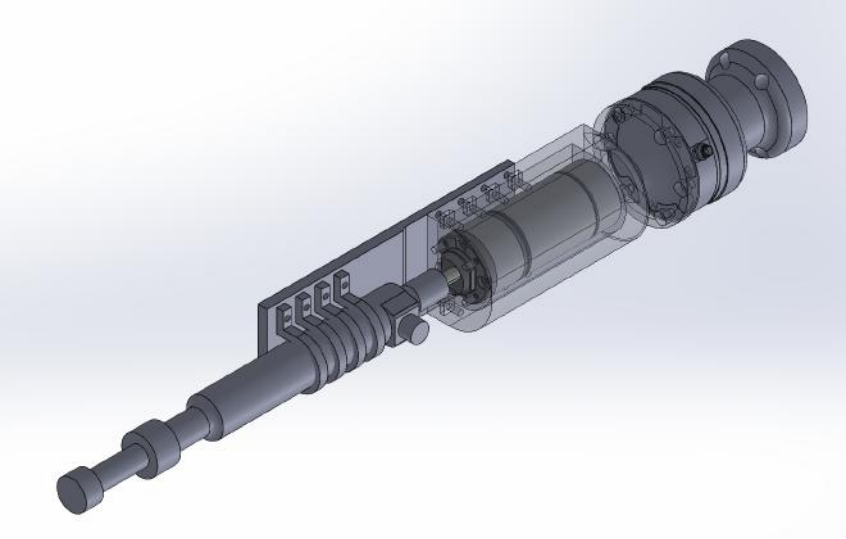

Figure 1: Solidworks Full Assembly

With the design complete it came time to 3D print all these components. We saw the need to 3D print these components rather than manufacture them out of steel or aluminum due to the payload of the Gen-3 being relatively low due to being a collaborative arm. All the parts were printed on my personal Ender 3 Pro system and after around 40 hours of prints (all of which completed with a nozzle diameter of 0.4 mm and a layer height of 0.2 mm) the finalized reaming end-effector could be put together. The resulting end-effector was attached to the Kinova Gen-3 with all necessary screws and nuts, and can successfully be moved around by the robot arm, as can be seen in figure 2.

1.2 Motor Control PCB

The motor control PCB required a decent amount of work to get finalized for this assignment. It was an iterative process of going back and forth with submissions and receiving feedback, allowing us to change things before moving forward with the assignment prior to submitting again. I began with finalizing all the components we needed for our design, taking special care to verify that all

Figure 2: Reaming Handle End-Effector

the parts I was ordering (specifically the female header pins and male header pins) could interact with one another correctly and had correct dimensions on the board. I then finalized the board and bill of materials and submitted the results to FreeDFM. However, Anthony pointed out that I was not utilizing all the functionalities of the Pololu board such as the fault detection and current sensing, which could be useful for controlling the motor, thus I added those traces and resubmitted to FreeDFM. The results of the FreeDFM verification can be seen in figure 3, and the results from FreeDFM was that there were no show-stoppers on the board and just some silkscreen line width issues which could be automatically fixed.

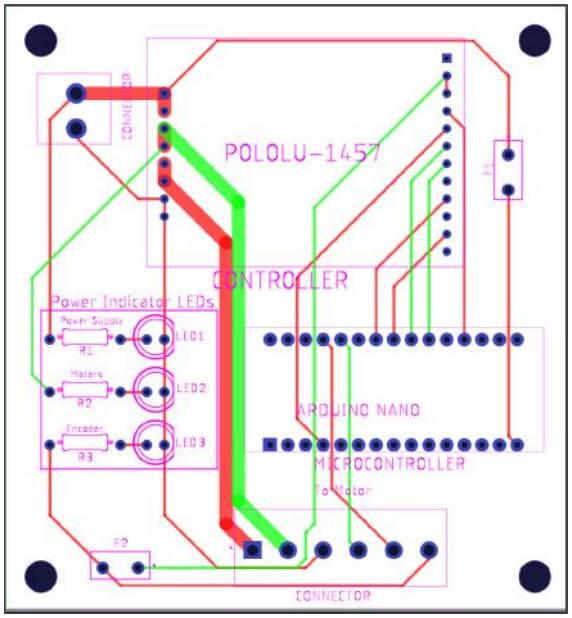

Figure 3: FreeDFM Results

1.3 Motor Speed/Torque Test

The motor speed and torque test was part of our tests we needed to demonstrate during our progress review presentation. This test was devised as a way to test the mechanical setup prior to the full mechanical system test which will be demonstrated in the next progress review. However, it really boiled down to just demonstrating that the motor we bought advertised it's performance correctly. As such, very crude tests were devised to determine if the speed and torque were correct. The no load speed was measured using a video camera capable of capturing the rotation of the shaft in slow motion, and it was determined that the advertised 612 rpm no load speed is accurate to the true performance of the system as we measured 600 rpm. The no load current was also determined to be around 0.5 A as advertised. The torque test was much more crude that the speed test as an adapter like a mallet was 3D printed to fit onto the motor. Using this mallet, a scale was utilized to measure the force the motor could exert at a specified distance, allowing for a rough torque measurement. This torque was found to be 14 kgcm which is close to the advertised 16kgcm stall torque. As a result our motor passed our motor speed/torque test, proving that it could perform well enough to work for our system.

1.4 Julia ROS Node

The final item I worked on during these past two weeks was to help Anthony and Sundaram with understanding how to connect the motion planning and controls subsystems. This required a way of interfacing ROS with Julia, and inversely, Julia with ROS. Anthony proposed that I utilize RobotOS.jl for this task, as it provided some functionality to ease the ability for Julia and ROS to communicate with one another. This library made it incredibly easy to interface the two, and as a result I was able to take in a JointEffortController message type from one topic and output a different edited JointEffortController onto a separate topic. As a result, a generalized template for the ROS to Julia bridge was created and can be filled in with more information as the MPC becomes more finalized.

2 Challenges

2.1 Reaming End-Effector

There were a myriad of problems with the reaming end-effector and there continue to be challenges that we have to face in the future. For one, the 3D printing process was difficult on account of my personal printer having some small issues, leading to a couple failed prints of several hours. When assembling the reaming end-effector, it was discovered that the force-torque adapter did not have deep enough holes to interface with the force-torque sensor properly, which requires screws of a very particular length to interface with. The long plate for interfacing between the motor housing and reaming handle is too thin and flimsy, leading to a lot of play in the system as the robot arm is moving, which could lead to inaccuracies in surgery. We determined that many of these challenges are a result of the reaming handle just being much too long for what we need it to be doing, and in fact it's length has led the entire reaming end-effector to be greater than half a meter in length.

2.2 Motor Control PCB

The biggest challenge with regard to the motor control PCB was the parts elicitation aspect of the assignment. Several times I struggled to find libraries and models which I could import into EAGLE and have the correct symbol and footprint. While Library Loader and Mouser were helpful with this to some degree, I eventually just forced myself to get more comfortable with creating custom parts in EAGLE by referencing the provided dimensions of specific parts. I did this for the Pololu-1457 and fixed the labeling on the pins for the Arduino Nano. This ended up being a much more efficient method for getting custom parts into EAGLE, so long as the parts I designed end up being accurate to their true dimensions.

2.3 Julia ROS Node

No real struggles with this except for the fact that I'm new to both Julia and ROS, and so working with both in tandem required some extra time that it may not have for others. Overall working on this part of the project helped improve my understanding of both, so I feel this was a worthwhile diversion from my primarily mechanical focus so far in the project.

3 Team Work

- Anthony: Anthony worked on further developing the Model Predictive Controller, refactoring the code to use previously overlooked functions implemented in the RigidBodyDynamics.jl Library, and simplifying the dynamics model to only include joint positions and velocities in the state to simplify the overall MPC to get an initial code base. Then, in order to get the MPC to converge, Anthony worked to create an appropriate initial guess for the input/torque trajectory based on the given state trajectory, working on several approaches. Once the MPC converged Anthony worked on visualization, optimization and initial code implementation of the MPC into ROS using RobotOS.jl library in Julia. Anthony also collaborated with Sundaram to get a sample trajectory to work with and sanity checked Parker's PCB.

- Gunjan: Gunjan worked on developing the landmark capture functionality along with Kaushik. During this, the main tasks involved understanding the PointCloud2 message type in ROS, prototyping a simple pointcloud collection function and visualizing it in RViz. Later, they performed a test to verify the scale of the collected pointcloud and to study the effect of orientation of registration probe on the collected points. Gunjan also prepared slides for and presented the project management portion of the PDR.

- Kaushik: Kaushik worked alongside Gunjan in developing the landmark capture capability to convert the marker pose into pointcloud2 messages. This was then visualized on RViz and verified with ground truth to ensure the data is of the right scale. He also modified the URDF to incorporate new design changes and also added a force-torque plugin into the Gazebo simulation. He assisted Sundaram in extracting a trajectory planned by the Pilz planner for preliminary testing of the MPC controller.

- Sundaram: Sundaram worked on generating the trajectory for the arm to follow and worked on setting up the necessary packages and functions to do so. He first worked on setting up the arm and connecting it with a computer via ethernet. To do this, he set up the IP address of the system and validated if the joint parameters are visible via the KINOVA API. Sundaram also worked on setting up the Pilz Industrial Motion Planner in MoveIt! to be used as a trajectory planner. This was a involved process as a completely new planning-pipeline had to be set up and had to be included in the convoluted code written by Kinova. He also changed the IK solver from KDL to IKFast for which he set up a docker image and created an IKFast plugin. With all the improvements, Sundaram was successfully able to generate deterministic trajectories and send these commands to the arm to validate it's motion. Sundaram also collaborated with Anthony to send joint-state trajectories to the MPC controller and computed the cartesian pose of the end-effector.

4 Plans

4.1 Reaming End-Effector

The reaming end-effector requires a redesign. While it will be nice to receive a 6mm to 7mm flex coupling to couple the motor shaft to reaming shaft and show that the current end-effector can be utilized as a reamer, our final design will need to use a better end-effector that is shorter in length and more robust. A good idea we received from the TAs and from our sponsors is to flip the motor around and create a belt transmission between the reaming handle and the motor, thus allowing the motor and the reaming handle to be parallel with one another, which would solve the length and robustness problem.

4.2 Motor Control PCB

Hopefully there is not much else to really do with the PCB, or else that would mean there is a problem in our design. Once we receive the PCB, we'll need to solder on all of our components, program the Arduino nano, and begin testing it with the motor to verify that the system works as designed. With the PCB verified to work we can then mount it to our Vention setup and utilize it as part of our full hardware system.

4.3 Julia ROS Node

We'll need to utilize the template I've created so far to properly interface the motion planning subsystem with the controls subsystem, allowing for easier testing of both and subsystem integration.