Individual Lab Report 9 - Progress Review 10

Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Anthony Kyu

Team C:

Kaushik Balasundar | Parker Hill | Anthony Kyu
Sundaram Seivur | Gunjan Sethi

November 3rd, 2022

Carnegie Mellon University
The Robotics Institute

Contents

1 Individual Progress

Several tasks were completed since the last progress review, including redesigning some of the end-effector parts for more stiffness and for manufacturability, integrating the Watchdog Module with the controller, and porting and testing the controller from simulation to the real arm.

The first task that I worked on was to modify and even redesign some end-effector parts to be both more rigid and to be easier for manufacturing. Besides some slight modifications on some parts to make it lighter weight, the main part that I redesigned was the adapter between the arm and the rest of the end-effector. In the previous design, it was braced on only one axis, which caused the design to be prone to vibrations along the axis it wasn't braced on. The redesign makes it much more rigid, and can be seen in the figure below (Figure 1). I also created drawings for parts that required high tolerances so that the manufacturer would know what we required.

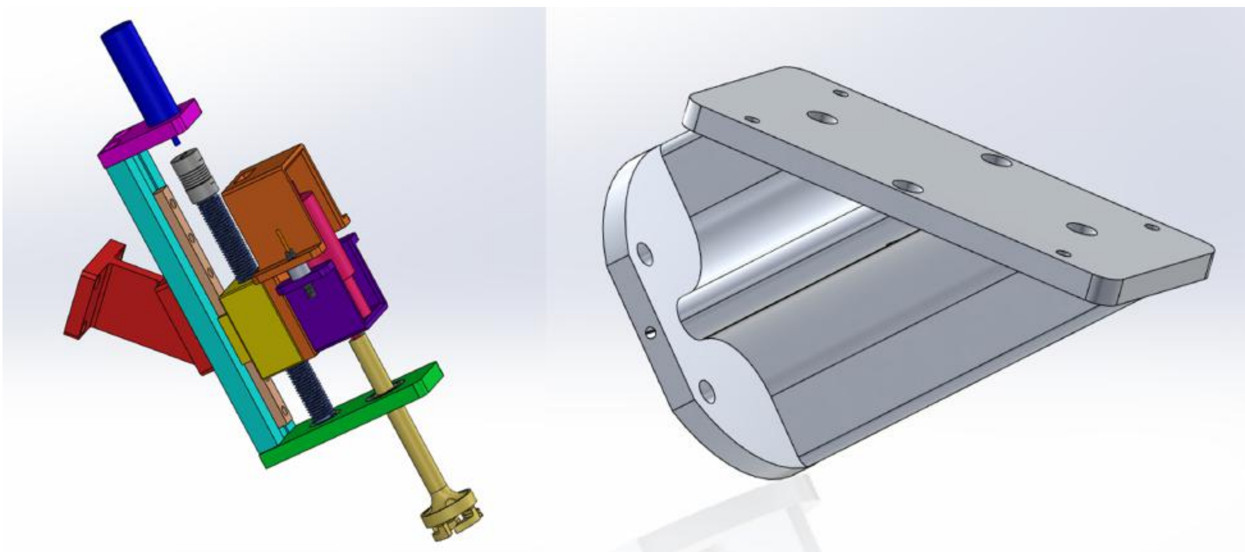

Figure 1: Comparison between end-effector adapters. The old one is on the left (red), and the new one is on the right.

The next task I worked on was working with Sundaram to integrate the Watchdog and Controls subsystems. This involved creating an interface between the subsystems using publishers and subscribers to send and receive information on whether the Watchdog should stop the system by setting a fault or what the tracking error is for aligning the reamer with the acetabulum.

The last task that I worked on was setting on the Controls subsystem to work on the real arm, and then test the algorithms tested in simulation such as joint limit avoidance, singularity damping, singularity avoidance, collision detection, and task prioritization. From last progress review, the only new algorithm is the collision detection algorithm, which was implemented using MoveIt's Collision framework to detect collisions with itself and with the environment. The figures below show joint limit avoidance, collision detection, and task prioritization with singularity avoidance.

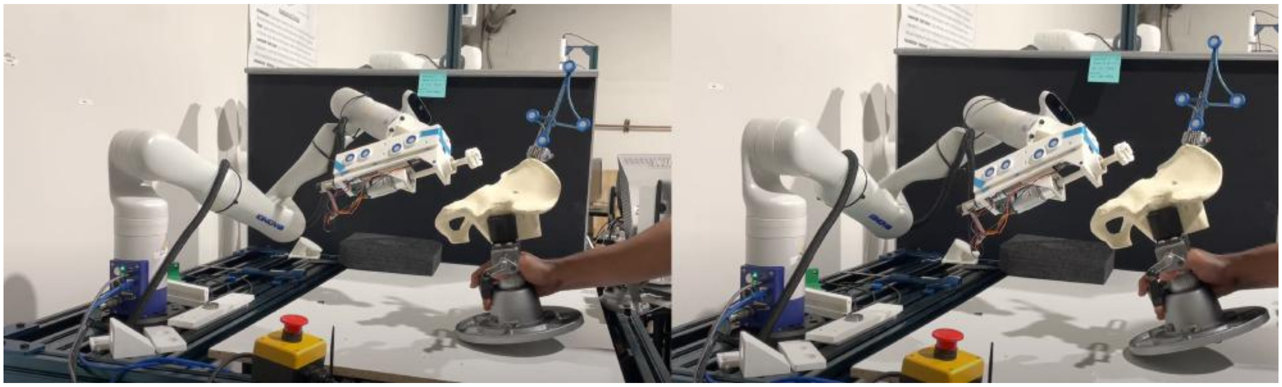

Figure 2: Joint Limit Avoidance Test. On the left, joint 2 is getting close to its joint limit, so the IK controller stops using joint 2 and tracks pelvis using other joints (right).

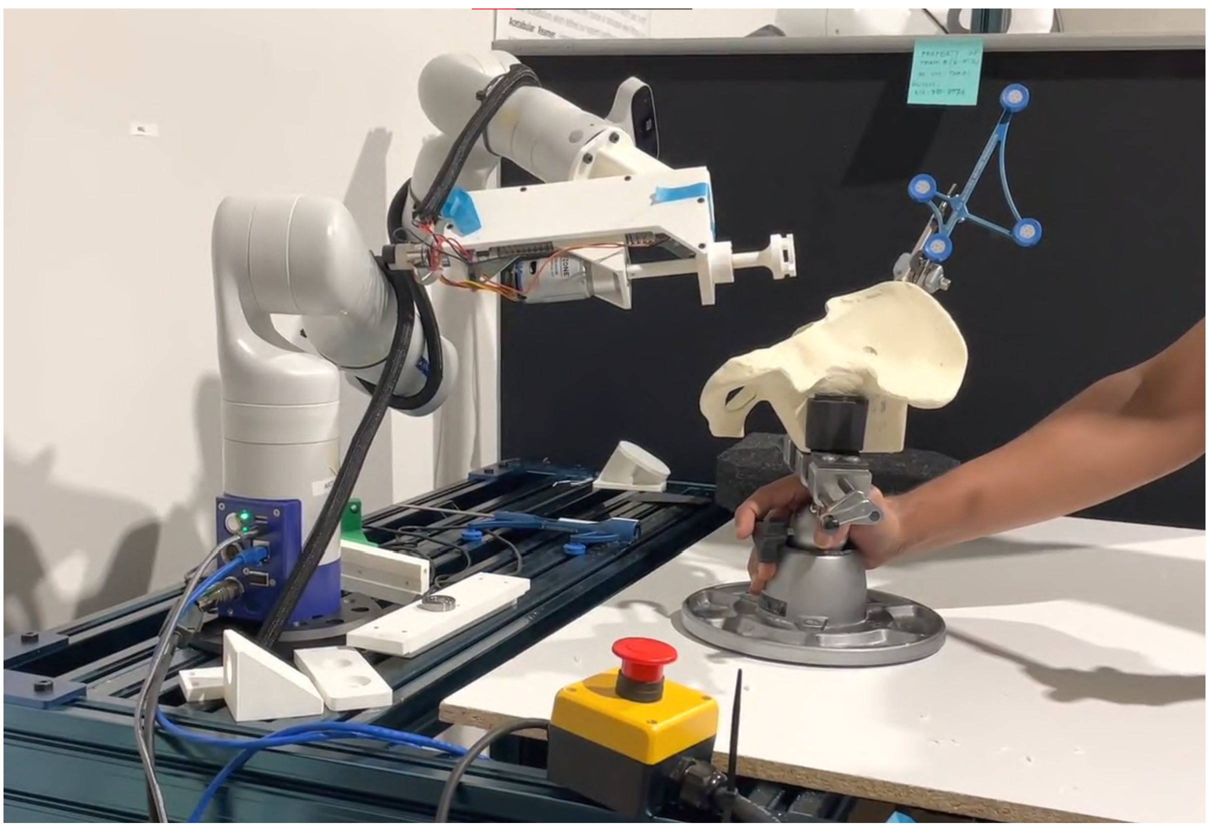

Figure 3: Collision Avoidance Test. When the arm gets close to colliding with itself, or the table, it throws a fault and stops moving. Here, the end-effector is about to collide with the arm, and the controller stops to prevent doing so.

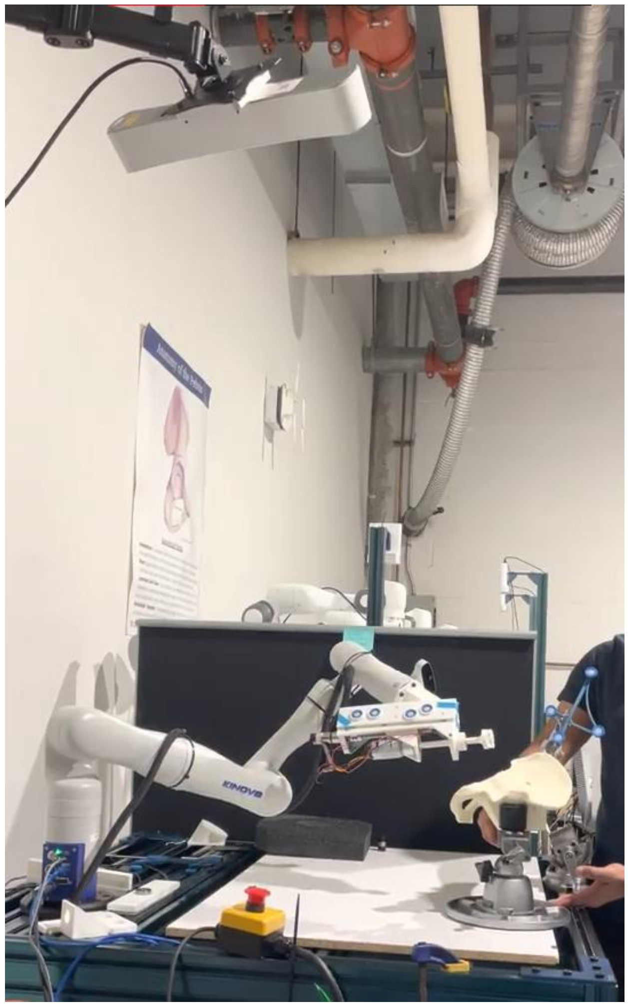

Figure 4: Task Prioritization Algorithm In Action. The arm's primary task is to track the pelvis position and orientation. With the remaining DoF, it moves the end-effector markers to align with the camera as much as possible, and moves the arm in null space to avoid singularities.

2 Challenges

The major challenges for this progress review were mainly with testing the controller safely on the new arm. Since we were testing a ton of new algorithms on the real arm for the first time, there were bound to be edge cases that could occur while testing on the real arm.

The two edge cases that were most obvious were, one, the pelvis moves outside of range and the controller gets stuck in singularity, and two, the arm collides with itself or the environment (table, wall, etc). So before testing on the real arm, the algorithms to prevent collisions or detect targets out of range had to be implemented, which not only took a lot of time, but the algorithms, at least for detecting out-of-range, were unclear as to what needed to be implemented to get the correct behavior.

Once implemented, testing was relatively smooth, but performing some tests was challenging. For instance, with the new end-effector angle, it was hard to get the robot into singularities even with singularity avoidance off to test singularity damping. It was also hard to test for joint limit avoidance while keeping markers visible for the Watchdog Module to not trigger the e-stop. These were eventually tested successfully, but it took several tests to get a proper testing setup, which speaks to the robustness of the system to avoid these edge cases in which these safety algorithms are activated.

3 Team Work

3.1 Anthony Kyu

Anthony worked with Parker and Sundaram to finalize the end-effector design and get it ready for manufacturing. He also brainstormed with Parker and Kaushik to design an end-effector controls architecture. Furthermore, he worked with Sundaram to integrate the Watchdog with the Controls subsystem, and to test the Controls subsystem. He implemented collision detection to prevent the robot from colliding with itself and environmental objects such as the table. And lastly, he implemented a detection algorithm to throw a fault if the pelvis is out-of-range of the arm or is in a bad configuration relative to the arm.

3.2 Parker Hill

Parker worked with Anthony and Sundaram to redesign and prepare the end-effector parts for manufacturing, and then helped order these parts from Xometry. He then worked with Kaushik and Anthony to brainstorm the end-effector controls architecture, and helped Kaushik implement this architecture within the Arduino. He also worked with Kaushik to implement the current sensors into the electrical subsystem, and worked with Anthony to set up the ATO load cells. And lastly, Parker worked on integrating the Watchdog with the UI subsystem.

3.3 Sundaram Seivur

Sundaram worked on the Watchdog Module, and worked with Anthony to integrate it with the Controls Subsystem, and also helped Anthony test the controls subsystem to understand edge cases and to double check desired functionality of Watchdog-Controls behavior. He also worked with Parker and Anthony to help prepare the end-effector for manufacturing, redesigning some pieces. And lastly, he worked with Gunjan and Parker to guide integration of Watchdog with the UI and to help decide UI functionality.

3.4 Kaushik Balasundar

Kaushik worked with Parker and Anthony to establish a general control strategy and framework for the end-effector controls. He also worked with Parker to calibrate the current sensor to be used for force measurement of the end-effector. And lastly, he assisted Gunjan with integrating the pointcloud collection and landmark selection pipelines to the user interface.

3.5 Gunjan Sethi

Gunjan refactored the UI code base for better modularity, readability and easier debugging. She also worked with Kaushik to integrate the point cloud collection and landmark selection tool ROS node with the UI. Furthermore, she helped Parker with integration between the UI and Watchdog subsystems. And lastly, she worked heavily on the development of the UI frontend to incorporate desired functionality interfaces with other subsystems.

4 Plans

For the next progress review, I plan on finalizing integration of the controls with the watchdog, which is mostly finished except for a few small details. For example, the controller has its own fault flag that the watchdog or UI must reset.

In addition, I will also be cleaning up the controller code and fix the collision detection feature that was actually discovered during this progress review presentation. This involves creating environmental objects (either through moveit commands or custom stls through Solidworks) and putting them into a urdf to test in simulation and on the real arm.

I will also be redesigning the end-effector marker array to have a less colinear design for better and more accurate tracking. I may also design an end-effector cover while I am designing the marker array.

Lastly, I will be working with Sundaram and Kaushik to define very detailed steps for validating our reaming operation, which involves learning how to use Open3D to subtract models and coming up with a metric to compare actual versus desired reaming.