# Individual Lab Report - 7

## Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR


Kaushik Balasundar

Team C:

Kaushik Balasundar | Parker Hill | Anthony Kyu

Sundaram Seivur | Gunjan Sethi


September 28th 2022


Carnegie Mellon University

The Robotics Institute

## Contents

# 1 Individual Progress

## 1.1 Simulation for Controls Testing

One of the key improvements that we aim to make this semester for our project is in the controller. Earlier, we had a wrench controller that commands the desired wrench at the end-effector. However, we would like to move towards individual joint control. Upon some research, I realized that the Kinova API only support a velocity controller, not a direct joint-level position control. As a result, we would have to send velocity commands to the robot. However, before we can test anything on the real robot, to make sure we do not damage the arm, I set up the velocity controller in simulation using the same ROS service as would be needed for the real arm. I also set up a dummy frame that the velocity controller can track. The motion of this frame makes the arm susceptible to both joint limits and singularities, and therefore would be a good test setup to ensure the arm motion is appropriate when a target command is in those regions. Figure 1 below shows the final simulation setup with the dummy frame to track.

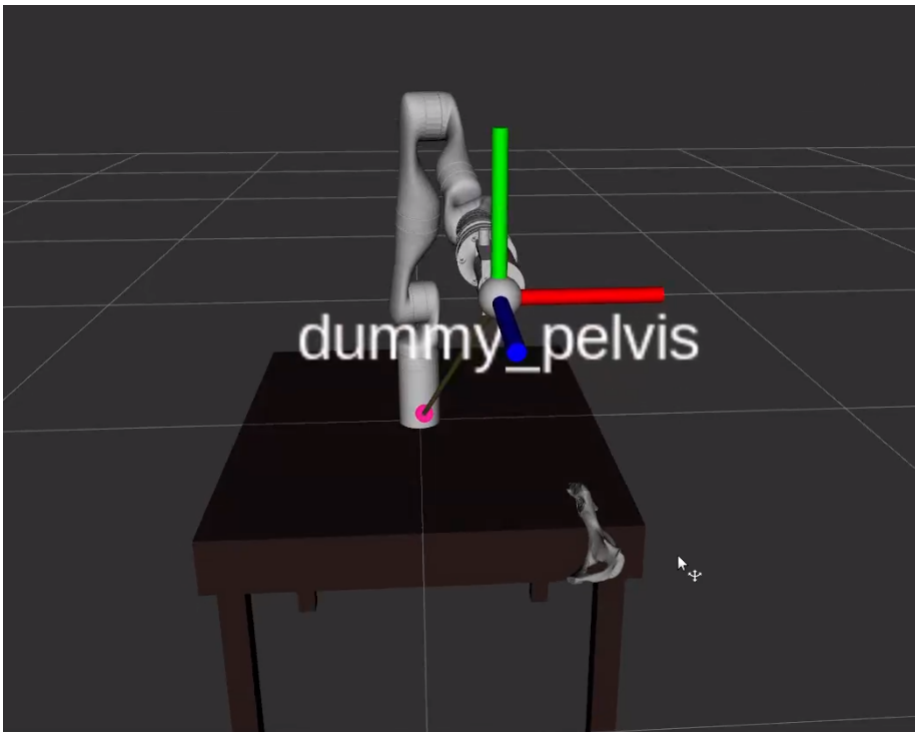


**Figure 1:** Simulation Environment setup with velocity command plugin and dummy pelvis frame to track

## 1.2 Controls Framework

I then worked closely with Anthony to set up a controller framework for the position controller using joint velocity commands which also takes into account joint limit and singularity avoidance. We utilize the redundancy of the robot arm to carry out null-space projection to facilitate the execution of different tasks. Since we needed to have access to the Jacobian at every iteration, we were not able to use any of the off-the-shelf inverse kinematics libraries. We therefore used some helper functions from the C++ KDL library and started writing our own inverse kinematics solver

from scratch. We implemented the Jacobian pseudo-inverse method for computing the inverse kinematics. We also implemented a method for joint limit avoidance and singularity avoidance using equation to run in parallel with the other tasks.

### 1.3 Repair of 3D models

I also stitched and repaired the 3D models of the pelvis we had obtained earlier from Prof. Shimada's lab to use for the new pelvis models that we purchased this semester.

## 2 Challenges

### 2.1 Debugging inverse kinematics controls framework

Since we were required to write the entire inverse kinematics framework from scratch, there were several implementation related bugs in our code that we needed to work through. For instance, the robot model had continuous joint with infinite joint limits. However, in our implementation, assigning joint limits to infinity caused computation errors. We therefore had to assign some artificial limits, and backtracking to this error required effort. There were also many tunable parameters in the controller framework that we needed to determine through trial-and-error.

### 2.2 Test-setup in the real arm

When testing the controller on the real arm for our PR test, we needed to have the robot arm follow a marker target. However, during practical execution of this test, we noticed that the arm blocked the field-of-view of the camera causing the test to fail. We therefore had to create a second frame with a translational and rotational offset with respect to the detected marker frame to make the test more practically feasible.

### 2.3 Bottleneck with controller frequency

The ROS API for the Kinova robot arm is bottlenecked at 40Hz, and this is a limitation for how fast the arm is able to track the pelvis frame. Since the pelvis does not move too much during, the current controller rate should be sufficient for dynamic compensation. In the future, if time permits, we will look to explore decoupling the controller by directly accessing the low-level API and bypass ROS for the controls.

# 3 Team Work

| Team Member | Contribution |
|---|---|
| Kaushik Balasundar | I set up a simulation environment to serve as a testbed to implement and validate the working of the velocity controller. I then worked closely with Anthony in implementing the new joint velocity controller in simulation with singularity damping and joint limit avoidance. I further helped validate the controller's performance and tune the gains for the real robot arm. |
| Gunjan Sethi | Gunjan developed the necessary script to convert STL-filetype pelvis scans to PCD format to facilitate usage in the current system pipeline. Further, she worked on assessing the feasibility of using RQt and Open3D for UI development. Gunjan also began development on the watchdog module. |
| Parker Hill | Parker worked with Sundaram and Anthony to finalize the CAD for the new linearly actuated end-effector design and sourced, printed out and assembled all components for the first version of the design. He also developed an outline and began sourcing parts for the electrical subsystem. Finally, he spent some time with Kaushik learning more about the software aspects of the project to be able to help more with the user interface in the future. |
| Anthony Kyu | Anthony worked with Kaushik to implement a basic joint velocity controller on both simulations and on the real arm, implementing inverse kinematics, singularity damping, and joint limit avoidance algorithms. He also worked with Kaushik to test the performance of this controller, testing how well it could track a pelvis marker and tuning the PID gains to do so. Furthermore, Anthony worked with Parker to help finalize the CAD design, sourcing key components such as the motor, load cells (and load cell electronics), and the linear motion mechanism. He also helped Parker calibrate his 3D printer. Lastly, Anthony also put together a knowledge-sharing session with the team to explain the math and algorithms behind the Task Prioritization controller to be implemented. |
| Sundaram Seivur | Sundaram worked on finalizing the watchdog architecture and started implementing features for the watchdog. He made changes to the architecture based on the feedback provided by our sponsors. He worked on creating the wireframes for the User Interface and conceptualized the critical components that need to be visualized on the UI. He also assisted Parker in finalizing the design for the end-effector and helped evaluate the performance of the 3D printed assembly. |

## 4 Plans

In the next couple of weeks leading up to SVD, I plan to work on the following:

1. Work with Anthony to implement a task for the robot arm marker to continuously face the camera. This task would have a lower priority to the task of the arm aligning itself with the pelvis. We will continue to have joint limit avoidance and singularity damping in the framework.
2. Further test the velocity-command based position controller, and evaluate if it can be adopted for the final system.
3. Support Gunjan and Parker in Open3D to UI communication implementation.