Individual Lab Report - 9

Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Kaushik Balasundar

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

1.1 Integration between the UI and perception subsystems

I worked with Gunjan to begin the integration between the user interface and the perception sub-system. Earlier, we used separate windows for the selection of manual correspondences in the collected point cloud and the pelvis 3D model to perform registration. This was made possible by an Open3D functionality to create an editable visualization window. However, with the UI, these features needed to be integrated into a single window. This introduced new challenges since we could no longer utilize the native tools provided by Open3D. Instead, Gunjan and I wrote a custom function for the selection of manual correspondences, which was involved and required multiple iterations of debugging to get right. Moreover, the point cloud collection process was done by publishing the data through a PointCloud2 ROS topic. To visualize these points through our Open3D GUI, we further implemented a functionality that converts the ROS messages to native Open3D messages.

1.2 Interfacing current sensor

The hall-effect current sensor in our end-effector reads the current drawn from the linear-actuator motor. This current measured is proportional to the axial force being applied by the linear actuator on the saw-bone. To begin with, we first calibrated the current sensor by applying a known amount of current and measuring the corresponding value of analog reading between 0 and 1023 from the Arduino Mega. We then determined a linear relationship representing the transfer function relating the applied current to the observed voltage in the sensor.

1.3 End-effector controls framework

Once the current sensor was calibrated, we moved on to the development of the controls framework for the end-effector. The general objective of this controller involves being able to actuate the linear actuator until it reaches the reaming end point while maintaining a predefined amount of axial force, and allowing the reamer to rotate at a predefined constant velocity. Our controller has a high-level state machine that includes the following states:

1. State 1 - Calibration: This is a calibration routine wherein the linear actuator moves back until it hits the limit switch. This initializes the position of the end-effector that will be tracked as it actuates during reaming.

2. State 2 - Wait for command: This is an idle state where the end-effector waits until it receives a command to start reaming from the task prioritization controls.

3. State 3 - Move until contact: Once a command is received for reaming, the linear actuator moves until it makes contact with a given amount of force against the acetabular surface.

4. State 4 - Start reaming: Once contact is made, the reamer motor turns on and maintains a constant reaming velocity until the goal state is reached or dynamic compensation becomes necessary.

5. State 5 - Dynamic Compensation: When the pelvis error exceeds the predefined position or orientation error thresholds, the end-effector retracts to allow repositioning and then subsequently changes state to continue the reaming process as usual.

6. State 6 - Finished reaming: Final state when the reaming is completed.

These high level states are used to determine the respective motor position and velocity commands for the linear actuator and reamer motors respectively. These are executed by the PID position and velocity controllers for the linear actuator and reamer motor respectively. The flowchart in figure 1 summarizes the functionality of our end-effector controller.

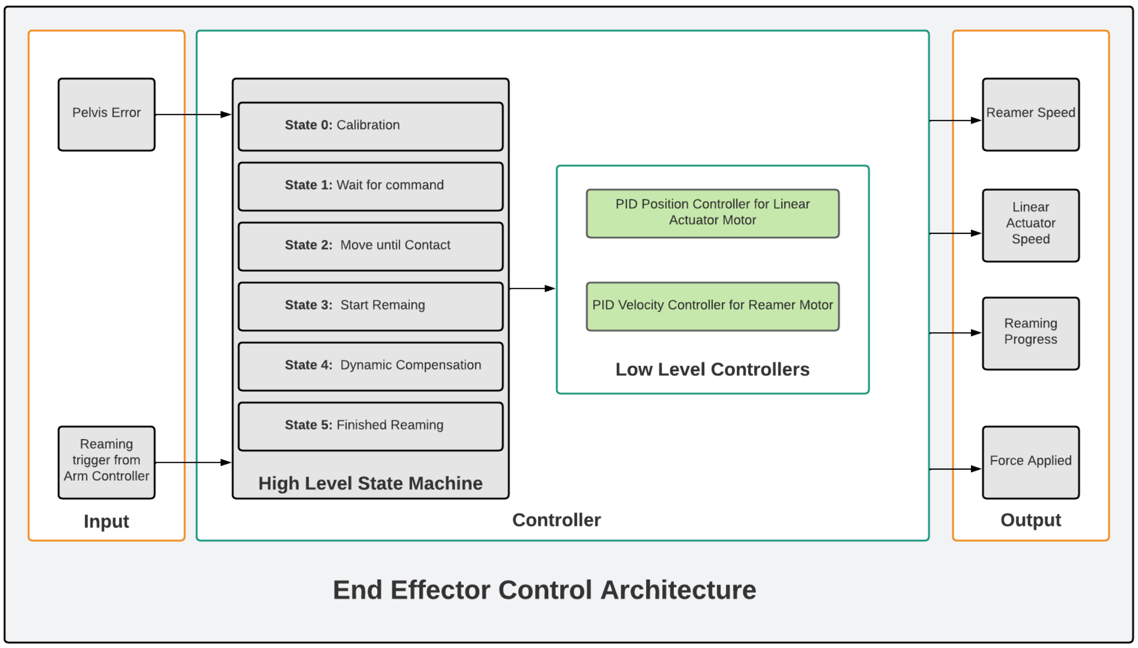

Figure 1: End-effector Controls Architecture

2 Challenges

2.1 Challenges with the UI

To visualize the collected point cloud in Open3D, we need to convert the transformations to the native Open3D point cloud format. This conversion process introduces latency in the visualization. Moreover, since we implemented a custom point cloud collection functionality within the user interface, the accuracy of the selection is not as good as that made available by Open3D. We need to tweak parameters to improve mouse sensitivity and selection region considerations to make the system more usable. However, as a fallback plan, we will shift our focus toward the alignment of the pelvis point cloud with the implant, since this is the biggest value addition to the user interface in our system. In this case, the perception actions will continue to happen independently from the UI. The UI will also display key health parameters of the system as communicated by the watchdog.

2.2 End effector controls

Currently, our implementation is messy and requires restructuring moving forward. We will create individual libraries for each actuator and sensor to make the code more modular. Moreover, we will have separate libraries for the low-level controls and high-level controls. As a next step, we also need to corroborate our current sensor measurements to the force applied. This will need to be done through a calibration procedure using load cells. We will also need to redo our wiring to be more robust to ensure reliability.

3 Team Work

Team Member	Contribution
Kaushik Balasundar	I worked with Gunjan in integrating the perception with the user interface. This involved implementing the point cloud collection and manual correspondence selection. We are still facing multiple bugs in implementing this, and are actively working towards resolving them. In addition, I worked with Parker to set up the end-effector controls framework, integrate the current sensor, and start the integration of the arm's controls and end-effector's controls. I also brainstormed integration of the end-effector controls with the watchdog.
Gunjan Sethi	Gunjan's efforts were primarily directed toward the integration of the UI with the various subsystems. She and I developed a point cloud collection and landmark selection capabilities natively on the user interface. It also integrates with ROS nodes to display real-time point cloud collection on UI. Both these tasks are currently in progress and require active debugging to finalize. She also worked with Parker on the watchdog and UI integration. Finally, she worked towards improving the front-end to incorporate more user tools/widgets for point cloud collection, registration, etc.
Parker Hill	Parker worked with Anthony and Sundaram to finalize the end-effector design and get it ready for manufacturing. He then ordered parts for manufacturing from Xometry. He also brainstormed the end-effector controls architecture with Anthony and I. He also helped me integrate the current sensor and begin the implementation of the end-effector controls code on the Arduino. He helped Gunjan in integrating the watchdog and end-effector controls with the UI and also worked with Anthony in setting up the ATO load cells.
Anthony Kyu	Anthony worked with Parker and Sundaram to finalize the end-effector design to get it ready for manufacturing. He also brainstormed the control system architecture with Parker and Kaushik. He then worked with Sundaram to integrate Watchdog and Controls Subsystems. They also worked with Sundaram to test the Controls Algorithms and Watchdog on the real arm. He then implemented self-collision checks into the controller to prevent collisions with the table for the arm itself. He also implemented out-of-range detection in the controller to stabilize controller behavior for pelvis targets out of range of the arm.
Sundaram Seivur	Sundaram worked with Parker and Anthony in finalizing the end-effector design and getting it ready for manufacturing. He worked with Anthony in integrating the watchdog with the controls subsystem and testing its functionality on the real arm. He also helped test the control sub-system with Anthony. He then worked with Gunjan and Parker in integrating the watchdog with the UI. He also brainstormed the requirements for the watchdog with me. Finally, he spent time getting team swag quotes for FVD.

4 Plans

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

1. Restructure code and complete development of the end-effector controls sub-system.
2. Work with Gunjan to assist in better system integration between the perception sub-system and UI.
3. Assist in performing full-system integration between the remaining sub-systems.
4. Work with Parker and Anthony in setting up an end-effector online calibration routine with the newly 3D printed marker geometry.
5. Evaluation of controls framework and rigorous testing of the entire system.