Individual Lab Report - 4

Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Kaushik Balasundar

Team C:

Kaushik Balasundar | Parker Hill | Anthony Kyu

Sundaram Seivur | Gunjan Sethi

March 24th 2022

Carnegie Mellon University

The Robotics Institute

Contents

1 Individual Progress

For this week, my focus was in assisting my teammates meet their PR goals since I was not an owner for any PR tests this week. In this regard, my primary contributions were towards writing the code along with Gunjan in converting a marker transform to a PointCloud2 message, helping Sundaram set up the Pilz motion planner, integrate the reamer handle with the arm in simulation, setting up the force-torque sensor plugin in our Gazebo simulation, as well as provide sample trajectories for validation of the MPC controller. Figure 1 shows the pointcloud collection process, Figure 2 shows the pointcloud visualized in Rviz and Figure 3 shows the pointcloud visualized using the Open3D viewer.

Figure 1: Pointcloud collection process from surface of acetabulum

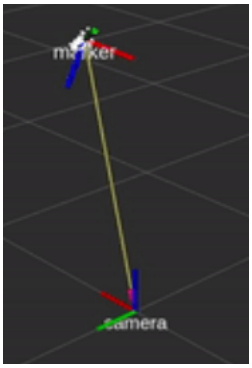

Figure 2: Pointcloud from acetabulum visualized on RViz

Figure 3: Pointcloud from acetabulum visualized on Open3D

1.1 Landmark Collection

There were two main approaches we could think of for doing this. The first being using the fiducial marker closest to the tip for registration. The second is using the centroid of the detected marker geometry. We decided to go for the latter since it is more robust and reliable. After finalizing an approach, we extracted the 3D coordinates of the point being registered from the C++ API for Atrycsys and then copied those values to the PointCloud2 message. We then varied various publishing frequencies and number of points to see how the result changes. I then saved the pointcloud into a .pcd file using the point cloud library, and visualized this on Open3D.

1.2 Updating URDF and Force-Torque Plugin in Gazebo

Our MPC controller takes in the joint positions of the various joints along the trajectory, the end-effector position during the trajectory, and the force-torque readings from the end-effector. In order to obtain the force-torque sensor readings, I interfaced the force-torque sensor Gazebo plugin into our URDF and verified that we were getting readings that made sense. This will form a part of the test used for validation of our MPC controller. Next, Parker had designed the new end-effector mount and adapter for our FT sensor. I integrated that into Gazebo, and computed the required inertias for these newly designed links, whose dynamics had to reflect accurately in Gazebo. This will form the basis of some tests performed in simulation to verify the working of our MPC controller.

1.3 Trajectory Extraction

I assisted Sundaram in extracting a trajectory generated by the Pilz planner using Moveit!, and use this information to validate the efficacy of the MPC controller. This involved identifying the right topic, recording a bag file, and also the transforms of the robot.

2 Challenges

2.1 Landmark Collection

It was a challenge to identify examples on how to use the C++ pointcloud messages. However, since Gunjan had worked on something similar in her previous workplace, it accelerated our progress. We also had to choose between whether to record only one fiducial close to the marker tip or multiple fiducials that form a rigid body. We chose the latter due to the improved robustness. While we are able to construct a pointcloud now, the collected pointcloud is from the center of the marker, rather than from its tip. The next week will involve extracting the pose of the marker tip using the known marker geometry.

2.2 Registration

In the next couple of weeks, we hope to use the pointcloud we collected to register with the CAD model created pre-operatively. However, the sparseness of the pointcloud is concerning. Moreover, the FPFH feature detector is not as robust when you have non-unique features. In our case, we try to register two hemispheres. Given the inherent symmetry of the geometries, unless we are able to capture unique features this will not be very robust. While this worked in simulation, using cross-source pointcloud data will need to be investigated. I plan to get an initial correspondence between the two pointsets by defining some known points in the 3D model and finding those points in pelvis. This will avoid any ambiguity in having non-unique solutions.

3 Team Work

Team Member	Contribution
Kaushik Balasundar	I 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. I also modified the URDF to incorporate new design changes and also added a force-torque plugin into the Gazebo simulation. I assisted Sundaram in extracting a trajectory planned by the Pilz planner for preliminary testing of the MPC controller.
Parker Hill	Parker worked on creating an end-effector reaming adapter for this progress review, going through the process of designing, 3D printing, assembling, and analyzing the assembly. He also worked on the PCB assignment, going through the steps to create the board, find parts, and verify performance for submission. Finally, Parker helped to perform the motor speed and torque test as well as helped to create a ROS node which runs using Julia.
Gunjan Sethi	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 the registration probe on the collected points. Gunjan also prepared slides for and presented the project management portion of the PDR.
Sundaram Seivur	Sundaram worked on setting up the packages and base code for generating the trajectory for the arm to follow. 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. He also worked on setting up the Pilz Industrial Motion Planner in MoveIt! to be used as a trajectory planner. He also changed the IK solver from KDL to IKFast. He was successfully able to generate deterministic trajectories and send these commands to the arm. He also collaborated with Anthony to send joint-state trajectories to the MPC controller and computed the cartesian pose of the end-effector.
Anthony Kyu	Anthony worked on developing the Model Predictive Controller - refactoring the code to use previously overlooked functions implemented in the RigidBodyDynamics.jl Library. He simplified 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, he worked to create an appropriate initial guess for the input/torque trajectory based on the given state trajectory.. Once the MPC converged, he 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.

Figure 4: Contributions by each team member

4 Plans

In the next couple of weeks leading up to the progress review 4, I plan to work on the following:

1. Improving pointcloud collection: I will work with Gunjan on refining and improving the pointcloud collection process. This will involve getting the tool tip transformation, and extracting the pointcloud from the tip rather than from the center.

2. I will then test out poincloud registration using this collected pointcloud. If the performance

is bad, I will develop the code for initial correspondence matching to get the preliminary transformation needed for ICP registration.

3. I will further work with Sundaram and Anthony in setting up the required ROS infrastructure for validating the planning and MPC controller in simulation with additional inputs such as end-effector poses and force-torque sensing.

References

[1] Huang, Xiaoshui, et al. ”A comprehensive survey on point cloud registration.” arXiv preprint arXiv:2103.02690 (2021)