HIPSTER

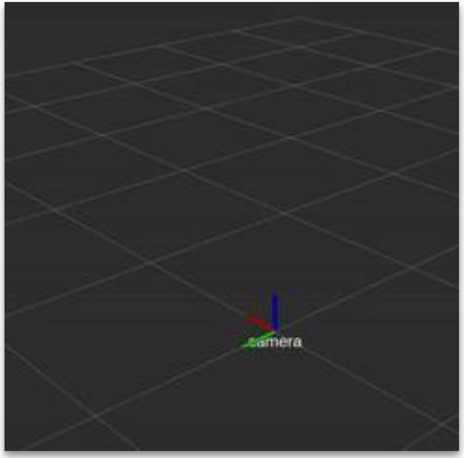
Autonomous **R**eaming for **T**otal **H**ip **R**eplacement **(ARTHuR)**

Progress Review - 4

Team C: Kaushik Balasundar, Parker Hill, Anthony Kyu, Sundaram Seivur, Gunjan Sethi

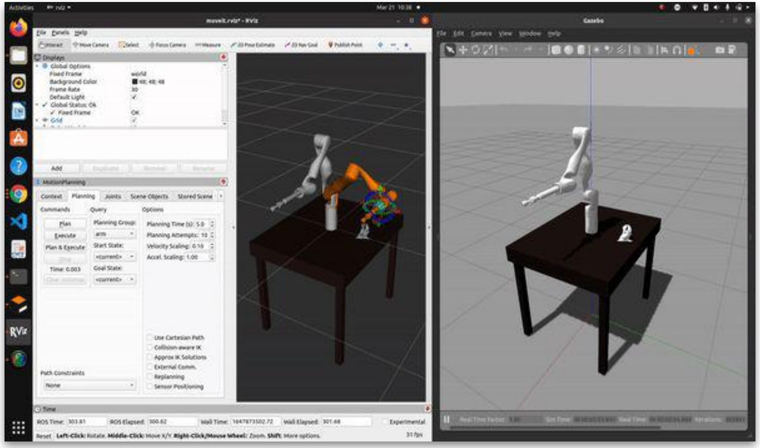
6 April, 2022

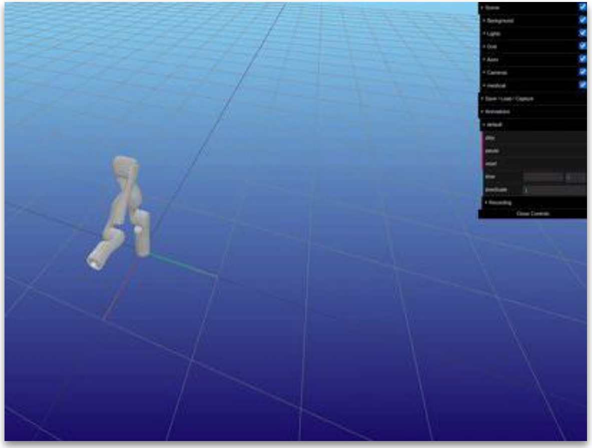
Previously



Capture Point Cloud

Hardware Setup





Planning & Controls

Schedule

Schedule			
Identifier	**Capability Milestone(s)**	**Associated Tests**	**System Requirements**
Progress Review 1 **2/16/2022**	- Test camera health and camera discovery via a ROS Node.	Test 1	M.F.1
Progress Review 2 **3/2/2022**	- Broadcast marker pose as a ROS transform & ROS topic - Validate the preliminary performance of the registration algorithm chosen	Test 2 Test 3 Test 4	M.F.1 M.F.3 M.N.1
Progress Review 3 **3/23/2022**	- Probe is able to be used to create a point cloud which can be visualized - Control method is capable of being used with robot manipulator virtually - Waypoint and trajectory generation working in ROS - Hardware verified for use in reamer assembly	Test 5 Test 11 Test 13 Test 17	M.F.1 M.F.2
Progress Review 4 **4/6/2022**	- Test registration using acquired point cloud and CAD models of pelvis - Evaluate quantitatively effectiveness of using ICP for cross-point sets registration - Create error and MPC validation pipeline - Implement additional states into MPC and full test in simulation - Assemble and test Power Distribution Board -Redesign end-effector for improved rigidity and decreased length	Test 6 Test 12 Test 14 Test 16	M.F.1 M.F.2 M.F.3 M.F.5 M.N.1

Progress Review #4 Tests

- ✓ Test registration using acquired point cloud and CAD models of pelvis — Perception and Sensing
- ✓ Evaluate quantitatively effectiveness of using ICP for cross-point sets registration — Perception and Sensing
- ✓ Create error and MPC validation pipeline — Planning and Controls
- ✓ Implement additional states into MPC and full test in simulation — Planning and Controls
- ✓ Assemble and test Power Distribution Board — Hardware
- ✓ Redesign end-effector for improved rigidity and decreased length — Hardware

Progress Review - 4 Tests

Acquiring Frame at Probe Tip

Figure: New Registration Probe

Navio files associated with Probe	Additional Information
900212_ProbeTWall.mat -1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000 105.0036 0.0000 1.0000 0.0000 11.176	900212_ProbeTWall.mat defines the relative position of where the burr axis touches the jig wall relative the the Probe origin
900212_ProbeTTip.mat 1.0000 0.0000 0.0000 0.0000 0.0000 1.0000 0.0000 288.6583 0.0000 0.0000 1.0000 8.07974	900212_ProbeTTip.mat defines the center of the 1 mm sphere at the end of the probe relative to the Probe origin

Figure: Transformations within Registration Probe

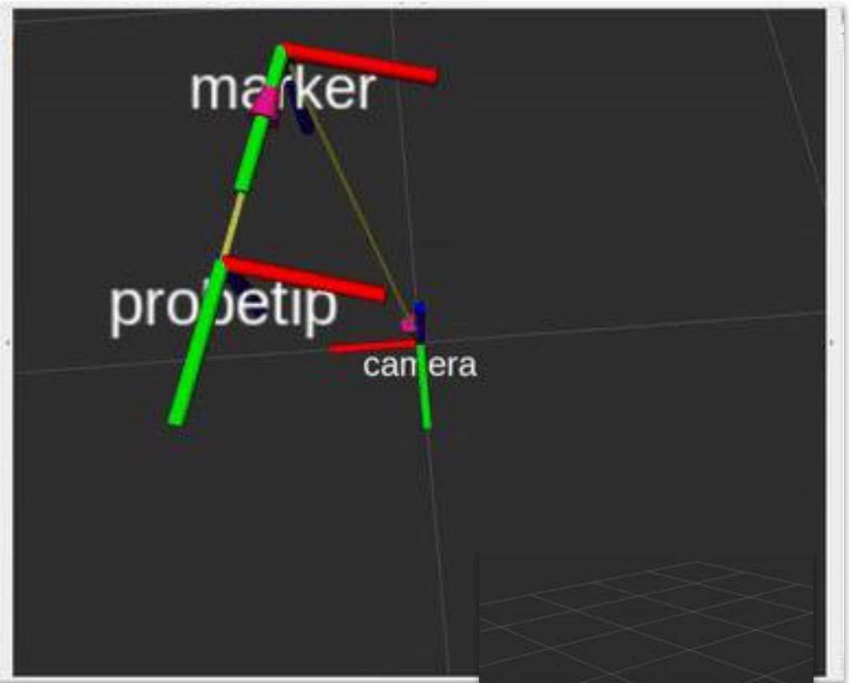


Figure: Camera, Marker and Probe Tip Frames

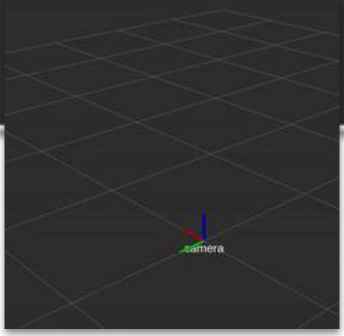


Figure: Previously

New Pointcloud Acquired Using Registration Probe

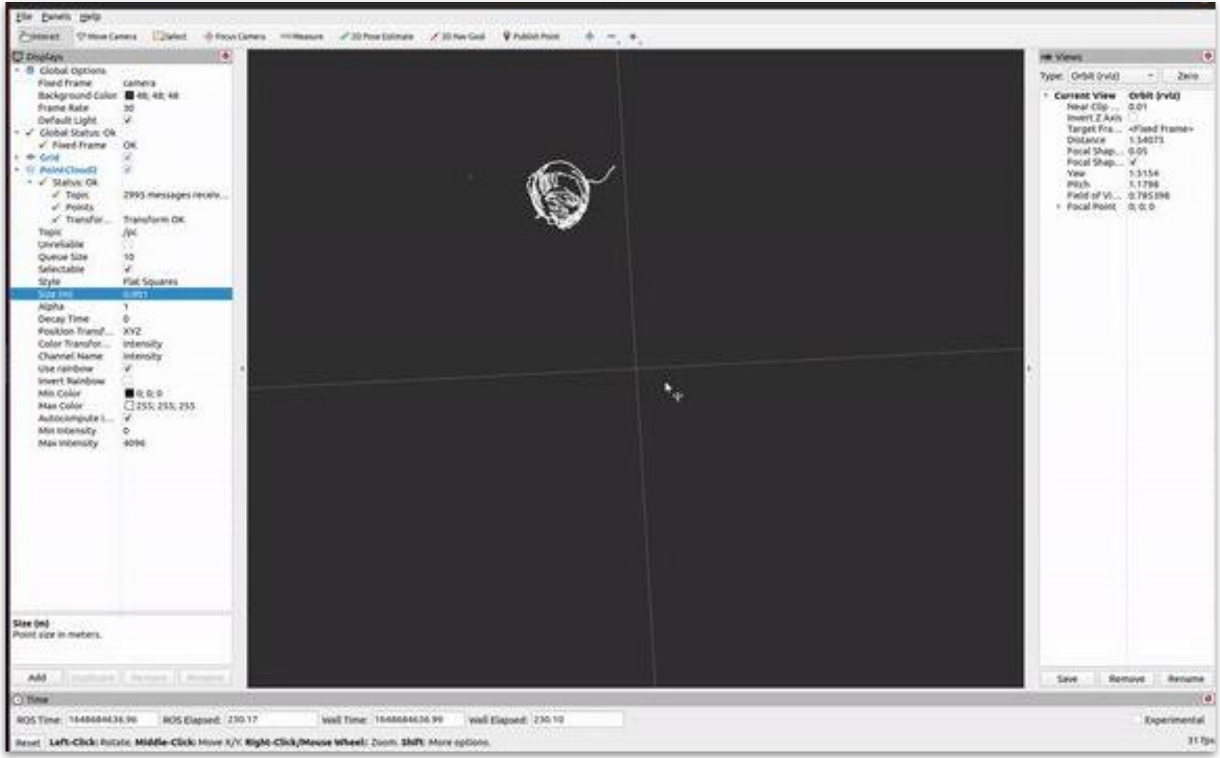
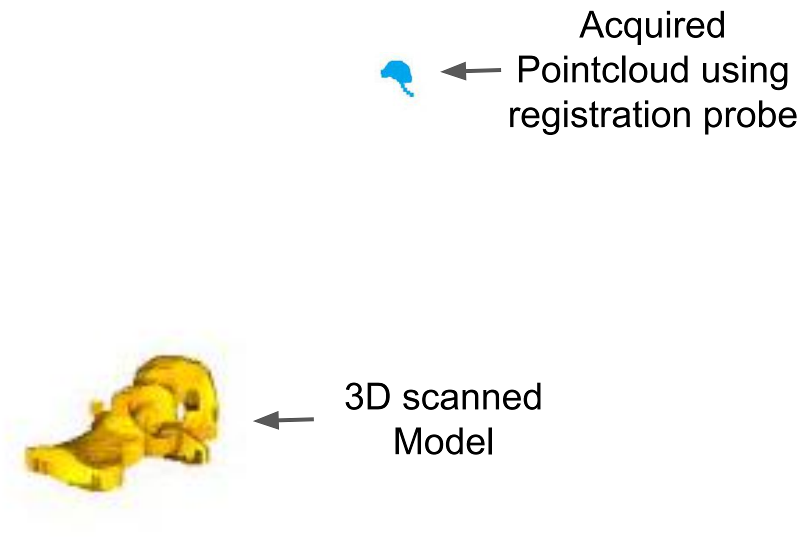
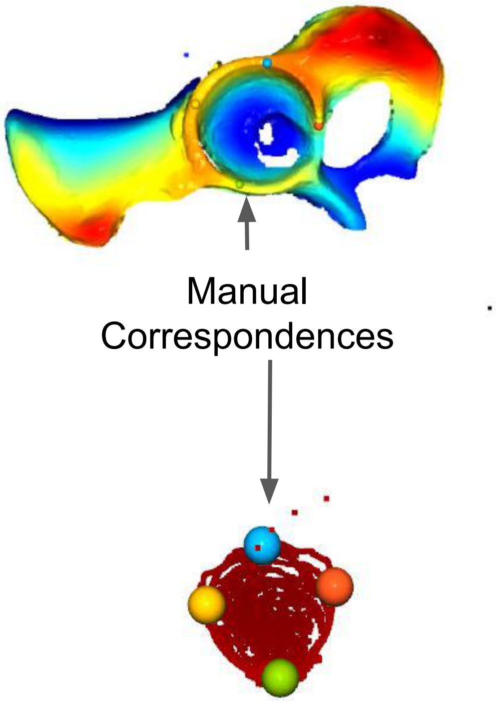


Figure: Pelvis Pointcloud Collection

Test 4: Evaluation of ICP using Real Pointcloud

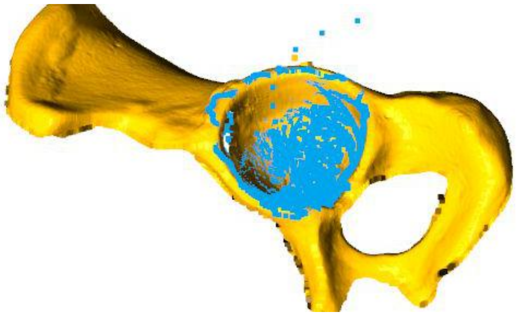


Before Registration



Manual Correspondence Matching for initial guess

Test 4: Evaluation of ICP using Real Pointcloud



Registration Result

```
Transforamtion is :  [[ 9.39464344e-01 -2.50722367e-01 -2.33548798e-01 -1.56972721e+02]
 [-3.42032678e-01 -7.26981768e-01 -5.95408394e-01 -3.63044569e+02]
 [-2.05035161e-02  6.39246277e-01 -7.68728693e-01  1.14271478e+02]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Evaluation:  registration::RegistrationResult with fitness=2.292607e-04, inlier_rmse=7.093910e-02, and correspondence_set size of 13
Access transformation to get result.
```

Registration RMSE Evaluation

Test 14: Adding Additional States in Trajectory

- ➔ Created custom message type
  - ◆ Message sending trajectory information: Joint positions, joint velocities, joint accelerations, cartesian positions, cartesian velocity
- ➔ Computed end-effector position using Forward Kinematics
- ➔ Wrote a class with functions to get current joint states, publish trajectories for modularities

```
oem@sas:~$ rosmsg info arthur_planning/arthur_traj
trajectory_msgs/JointTrajectory traj
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  string[] joint_names
  trajectory_msgs/JointTrajectoryPoint[] points
    float64[] positions
    float64[] velocities
    float64[] accelerations
    float64[] effort
    duration time_from_start
geometry_msgs/PoseArray cartesian_states
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  geometry_msgs/Pose[] poses
    geometry_msgs/Point position
      float64 x
      float64 y
      float64 z
    geometry_msgs/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
geometry_msgs/Point[] cartesian_vel
  float64 x
  float64 y
  float64 z
```

Test 14: MPC

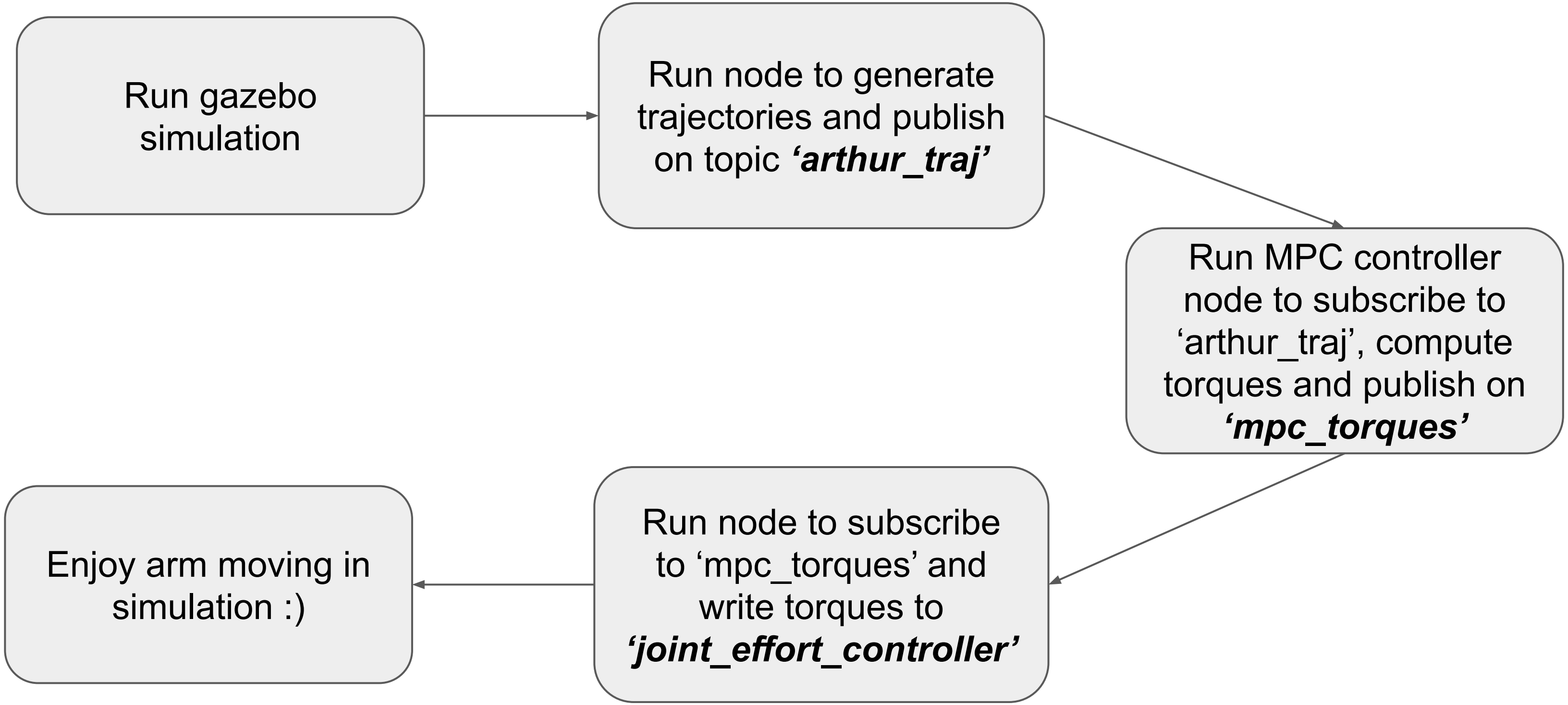
Goal:

- ✓ Integrate MPC into Gazebo Simulation
- ✓ Test MPC in real-time simulation
- ✓ Add additional states to MPC

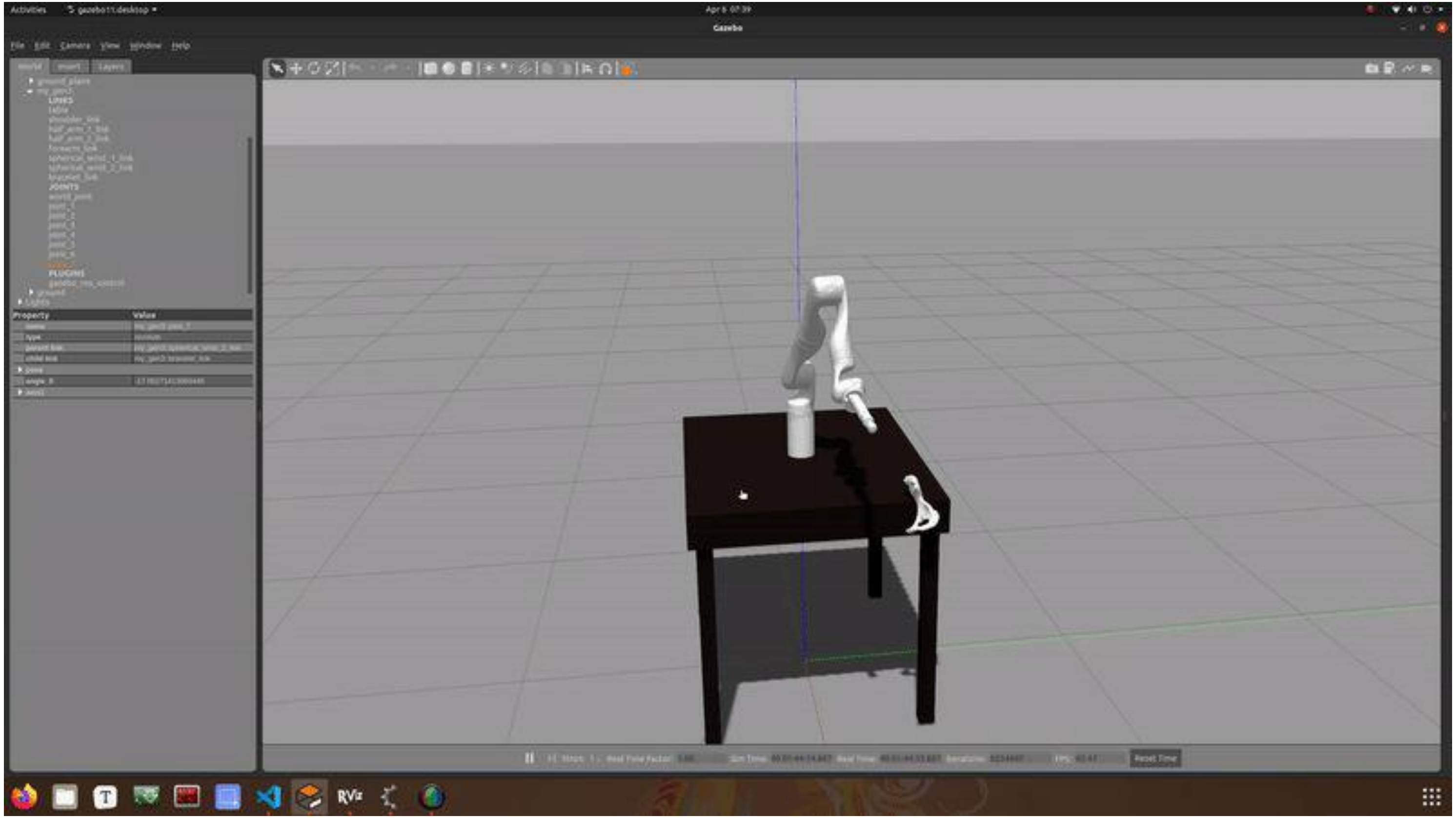
Current progress:

- ✓ Integrate MPC into Gazebo Simulation
- ✓ Tracking PD Impedance Controller is implemented in real-time Gazebo simulation
- ✓ Test MPC in real-time Gazebo simulation (still work inprogress)

Test 14: Planning & Controls Integration



Test 14: Tracking PD Impedance Controller



Planning & Controls Challenges

Challenges:

- ➔ Building a custom message and populating it with all necessary states
- ➔ Pilz linear planner takes a lot of time to generate trajectory, which will break the dynamic motion planning
- ➔ Altro solver takes too long to converge, slowing down controller response in real-time
  - ◆ Algorithm for calculating dynamics is not fast enough (may need to swap to C++ libraries in the future)
- ➔ Dynamics model mismatch between MPC algorithm and Gazebo simulation

Test 16: Hardware Setup

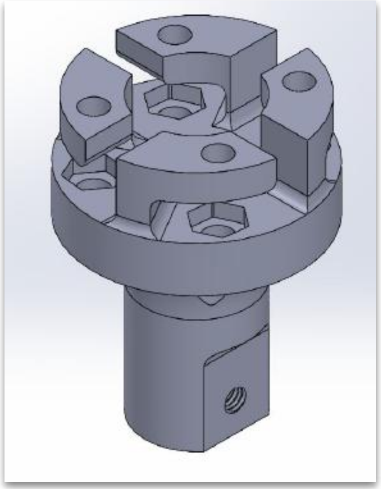
Goal: To finalize the hardware setup such that that arm is fully capable of reaming the acetebulum when properly controlled

Approach: 3D print adapters for the reamer head and motor to attach to the end-effector and develop a custom PCB to control it

Test 16 :	
Hardware Setup Test	
Objective	
To test the finalized hardware setup in order to determine that all hardware is rigidly secured and functioning properly during motions that the manipulator would undertake in typical reaming operations	
Equipment	System with Hipster Test Environment (MRSD Desktop 2), Robot Manipulator (with attached reamer end-effector), Marker Arrays, IR Markers, Atrascys camera, Sawbone pelvis, Panavise mount
Elements	Full Hardware System
Personnel	Parker Hill
Location	NSH Basement
Procedure	
1. Attach the reamer assembly to the end-effector of the robot manipulator and connect all wires to power 2. Attach Panavise mount to the manipulator table and clamp the sawbone pelvis in place 3. Attach marker array to the reamer assembly as well as to the pelvis 4. Place the Atracsys camera in an orientation to be able to view both the reamer and pelvis marker array 5. Actuate the robot manipulator using a built in controller to follow a similar motion path to what would be expected during the procedure 6. After placing the acetabular reamer into the sawbone pelvis's acetabulum, turn on the reamer slowly move along the reaming axis	
Validation	
- All hardware is secured properly and does not have any play as the manipulator moves around - Camera is capable of seeing both marker arrays throughout the procedure - Reamer is powered and capable of removing material from the sawbone pelvis	

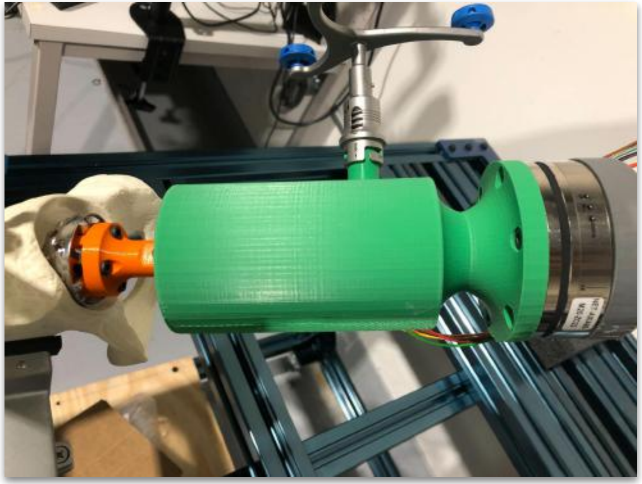
Test 16: End-Effector Redesign

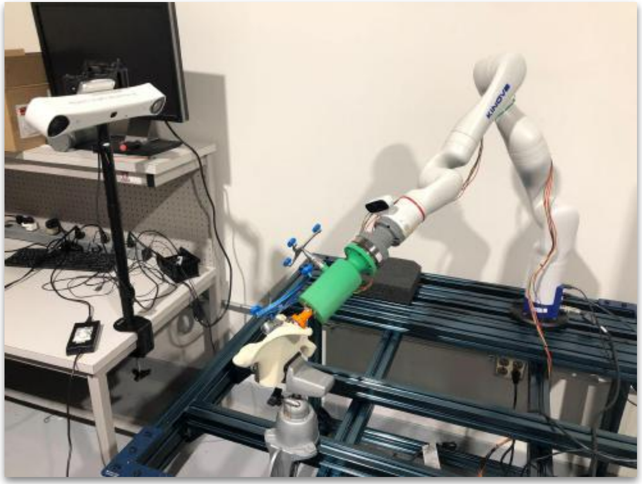
- Previous design was validated to be capable of actuating the reaming head
- This design remained too long and not rigid enough for our purposes (although the addition of a flex coupling did improve it’s rigidity)
- New end-effector design was created which mimics the geometry of the reaming handle without any unnecessary length
- Custom reaming head adapter was created and the motor holder was reprinted to include a marker holder



Test 16: Hardware Setup

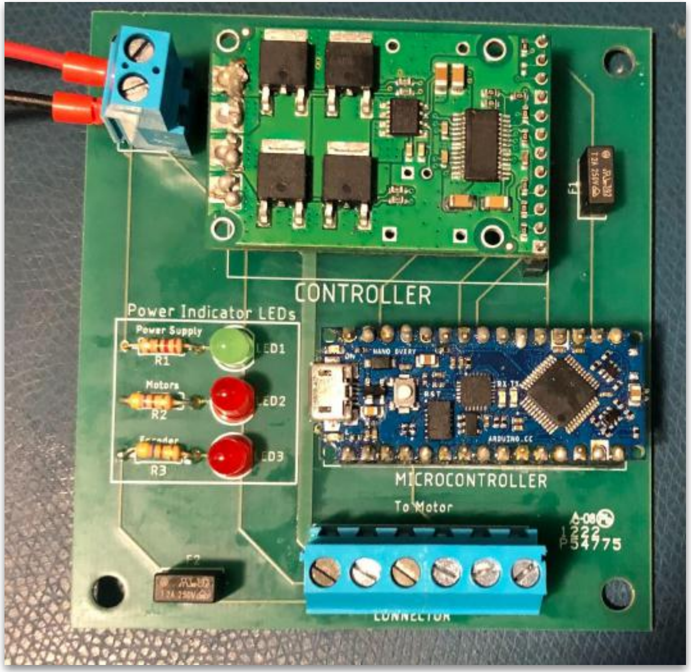
- Testing of finalized prints showed that the redesign yielded a more stable end-effector
- Preliminary reaming tests on a cardboard box showed large reverse torques which the robot's joints would have to adapt to, however the design held up
- Despite this we would like to manufacture the reaming head adapter out of aluminum in the future





Test 16: PCB Finalization

- PCB was put together successfully and connectivity was tested throughout the board to verify performance
- However, when doing a second pass and fixing some sloppy solders something changed in the motor controller
    - The PWMH pin which should be held at LOW by default was oscillating between 1 and 3 V
    - This led the arduino getting fried and the motor controller becoming unreliable



Test 16: PCB Finalization

- Replacement motor controllers have been ordered
- Replacement arduinos have been brought in from home in case there is further issues with the board
- Motor wires have been extended and routed along the arm
  - Will be improved in the future with better routing and wires held tighter together
- Ideally the board should be resoldered and functional early next week.

Plans: Spring Validation Demonstration

Future Work - Perception

- Integrate multiple-marker tracking into current code
- Standardize point-cloud collection and correspondence-matching procedure
- Integrate registration and point-cloud collection sub-systems
- Error detection pipeline

Future Work - Planning & Controls

- Port planning node from python to C++ to reduce planning latency
- Incorporate transforms from perception subsystem to plan trajectory
- Write scripts to plan new trajectory when error threshold is crossed
- Continue tuning MPC to make it more robust
- Improve communication frequency between various subsystems

Hardware Future Work

- Finalize PCB
- Develop arduino code to control reamer speed using PID velocity control
- Mount power supply and PCB to Vention table
- Reroute wires along arm
- Determine transformation between marker on arm to last link

Thank you!

Stay tuned for SVD :)