# Team C Spring 2022 Test Plan

## Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR


Team C:

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


February 14th, 2022


Carnegie Mellon University
The Robotics Institute

## Contents

## 1 Introduction

This document describes the various tests to be performed on the Hipster system (and subsystems in simulation/reality) throughout the fall semester in order to validate and verify that the system and subsystems are meeting stated functional and performance requirements. The tests are designed such that there is an incremental increase in the complexity of the test and the necessary state of the system in order to properly perform the task, which will provide a good deadline for finishing functionalities. The results of these tests will be reported during the progress reviews. Each test has a name/number, an objective, elements, a location, equipment, personnel, procedure, and verification criteria. By the spring validation experiment, our full reaming system should be functioning in simulation and all subsystems should be working in reality without integration.

## 2 Logistics

All of these tests as well as the Spring Validation Demonstration will take place in Newell-Simon Hall in room B512. The Spring Validation Demonstration will be presented via a live demo, while the rest of the plans will be demonstrated via videos or reports on the results of the tests during progress reviews. All team members will be present for the Spring Validation Demonstration, and while it would be ideal for all members of the team to be present for all tests, it is only be necessary for the system lead and one other person to be present during the testing. The following equipment would be necessary for the majority of our tests:

- **Desktop Workstation**: necessary for interfacing with the robot manipulating
- **Monitor**: necessary for displaying GUI information and camera output
- **Robot arm**: manipulator arm coupled with a custom reaming end-effector
- **Atracsys camera**: camera which can detect the location of marker arrays and computer their location into transformations
- **IR markers**: markers which can be detected by the Atracsys camera
- **Marker arrays**: arrays which hold the IR markers in unique orientations such that they can be detected by the Atracsys camera
- **Vention table**: rigid table for the robot arm and all tasks to be performed upon
- **Sawbone pelvis**: foam replica of a pelvis to be used with physical validations of the system
- **Panavise mount**: vise to hold pelvis during testing

Further equipment for specific tests will be specified in the testing plans. Some of these performance requirements and tests are subject to hardware we plan on receiving, and given the uncertain nature of our hardware acquisition currently, some of these requirements and tests may change.

## 3 Schedule

| Schedule | | | |
|---|---|---|---|
| **Identifier** | **Capability Milestone(s)** | **Associated Tests** | **System Requirements** |
| **Progress Review 1**<br>**2/16/2022** | - Integration of the camera with ROS and the ability to detect marker poses in Atrascys SDK | Test 1 | M.F.1 |
| **Progress Review 2**<br>**3/2/2022** | - Markers can be read into ROS and create a point cloud comparable to the pelvis geometry<br>- Control method is capable of being used with robot manipulator virtually | Test 2<br>Test 3<br>Test 4<br>Test 13 | M.F.1<br>M.F.3<br>M.N.1 |
| **Progress Review 3**<br>**3/23/2022** | - Probe is able to be used to create a point cloud which can be visualized<br>- Waypoint and trajectory generation working in ROS<br>- Hardware verified for use in reamer assembly | Test 5<br>Test 11<br>Test 17 | M.F.1<br>M.F.2 |
| **Progress Review 4**<br>**4/6/2022** | - Probe is able to create a point cloud which is comparable to the pelvis geometry<br>- Waypoint generation compensation is implemented virtually<br>- Control method is capable of being used with robot manipulator physically<br>- Robot manipulator end-effector and setup is fully assembly | Test 6<br>Test 12<br>Test 14<br>Test 16 | M.F.1<br>M.F.2<br>M.F.3<br>M.F.5<br>M.N.1 |
| **Progress Review 5**<br>**4/20/2022** | - Probe is able to create a point cloud which is sufficient for use in generating waypoints according to surgical plan<br>- Control method and motion planning fully integrated<br>- Detect all movement of the pelvis<br>- Manipulator able to be maneuvered in free motion mode | Test 8<br>Test 9<br>Test 10<br>Test 15 | M.F.1<br>M.F.2<br>M.F.3<br>M.F.4<br>M.F.6 |

# 4 Tests

## 4.1 Test 1

| Test 1: | |
|---|---|
| Camera Setup Test | |
| **Objective** | |
| Test camera health and camera discovery via a ROS Node. | |
| **Equipment** | MRSD Desktop 2, Atracys SpryTrack 300 Camera |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Run the camera_node ROS Node and wait for the node to discover the camera.<br>2. Wait for the ROS Node to load the geometry file. | |
| **Validation** | |
| - Camera's serial number is printed.<br>- Geometry file is loaded. | |

## 4.2 Test 2

| Test 2: | |
|---|---|
| Marker Pose Detection Test | |
| **Objective** | |
| Test fiducial marker detection via a ROS Node. | |
| **Equipment** | MRSD Desktop 2, Atracys SpryTrack 300 Camera, Markers |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Run the camera_node ROS Node and wait for the node to discover the camera.<br>2. Wait for the ROS Node to load the geometry file.<br>3. Place markers in front of the camera. | |
| **Validation** | |
| - Camera's serial number is printed.<br>- Geometry file is loaded.<br>- The marker pose is printed. | |

### 4.3 Test 3

| Test 3: | |
|---|---|
| Marker Pose Visualization Test | |
| **Objective** | |
| Test the publishing of marker poses onto a ROS topic and visualizing markers using RViz. | |
| **Equipment** | MRSD Desktop 2, Atracys SpryTrack 300 Camera, Markers |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Run the camera_node ROS Node and wait for the node to discover the camera.<br>2. Wait for the ROS Node to load the geometry file.<br>3. Place markers in front of the camera.<br>4. Run rostopic command-line tool to view messages on the marker-pose topic in ROS.<br>5. Run RViz | |
| **Validation** | |
| - Camera's serial number is printed.<br>- Geometry file is loaded.<br>- rostopic command line shows marker poses.<br>- Markers appear on RViz. | |

### 4.4 Test 4

| Test 4: | |
|---|---|
| Preliminary Point Cloud Registration Test | |
| **Objective** | |
| Validate the selection of the registration algorithm for the use-case and test the ability of the registration algorithm to register the simulated fiducial points onto the pelvis point cloud. | |
| **Equipment** | MRSD Desktop 2, Atracys SpryTrack 300 Camera, Markers |
| **Elements** | Perception Subsystem |
| **Personnel** | Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Load two pointsets, one of the complete pelvis model and the other of the downsampled point cloud from the surface of the acetabulum.<br>2. Run the selected registration algorithm and visually validate the transformation from pointcloud A to pointcloud B. | |
| **Validation** | |
| - Pointcloud B needs to be roughly overlapping with pointcloud A's acetabulum region to indicate that the registration has taken place. | |

## 4.5 Test 5

| Test 5 : | |
|---|---|
| Landmark Capture Test | |
| **Objective** | |
| Test the use of the registration probe to record fiducial landmarks on pelvis and test the ability to use Open3D to store the selected points as a pointcloud. | |
| **Equipment** | Atracys Sprytrack 300 Camera, Markers, Registration Probe MRSD Computer 2, Model Pelvis |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi & Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Use the registration probe to slide through the acetabuluar surface in the field of view of the camera.<br>2. Once done, run ROS script to visualize the captured points as Open3D pointcloud. | |
| **Validation** | |
| - The resulting visualization must be in the form of an Open3D visualization window. It must display the captured points that depicts the surface of the acetabular surface. | |

## 4.6 Test 6

| Test 6 : | |
|---|---|
| Secondary Pointcloud Registration Test | |
| **Objective** | |
| Test the ability of the registration algorithm to register the physically probed points onto the pelvis pointcloud and validate qualitatively. | |
| **Equipment** | Atracys Sprytrack 300 Camera, Markers, Registration Probe MRSD Computer 2, Model Pelvis |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi & Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Use the registration probe to slide through the acetabuluar surface in the field of view of the camera.<br>2. Once done, run ROS script to register the pointcloud with stored pelvis pointcloud and validate qualitatively. | |
| **Validation** | |
| - Pointcloud B needs to be roughly overlapping with pointcloud A's acetabulum region to indicate that the registration has taken place. | |

## 4.7 Test 7

| Test 7 : | |
|---|---|
| Quantitative Registration Test | |
| **Objective** | |
| Test the ability of the registration algorithm to register the physically probed points onto the pelvis pointcloud with inlier RMSE < 0.05 | |
| **Equipment** | Atracys Sprytrack 300 Camera, Markers, Registration Probe MRSD Computer 2, Model Pelvis |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi & Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Use the registration probe to slide through the acetabuluar surface in the field of view of the camera.<br>2. Once done, run ROS script to register the pointcloud with stored pelvis pointcloud and validate qualitatively. | |
| **Validation** | |
| - Pointcloud B needs to be registered with pointcloud A's acetabulum region with inlier RMSE < 0.05 | |

## 4.8 Test 8

| Test 8 : Surgical Plan Tranformation Test | |
|---|---|
| Surgical Plan Transformation Test | |
| **Objective** | |
| To test the ability of the system to convert a surgical plan specified in thepelvis model to world coordinates that can be taken as input | |
| **Equipment** | Atracys Sprytrack 300 Camera, Markers, MRSD System 2, Model Pelvis |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi & Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Use the registration probe to slide through the acetabuluar surface in the field of view of the camera to register the pelvis acetabulum with the 3D model.<br>2. Specify the pose of the acetabular implant (x,y,z,roll,pitch,yaw) to be at the pelvis marker's centroid.<br>3. Run ROS script to transform to world frame coodinates with registration result | |
| **Validation** | |
| - The ROS script's pose output must be the same as the pelvis marker reading read by the Sprytrack camera with a maximum error of 3 mm in position and 3 degrees in orientation combined. | |

## 4.9 Test 9

| Test 9 : | |
|---|---|
| Pelvis Motion Detection Test | |
| **Objective** | |
| Test to verify that perception subsystem is capable of processing data to determine if the pelvis has moved past a specified error threshold | |
| **Equipment** | Atracys Sprytrack 300 Camera, Markers, MRSD System 2, Sawbone Pelvis |
| **Elements** | Perception Subsystem |
| **Personnel** | Gunjan Sethi & Kaushik Balasundar |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Place the pelvis in the field of vision of the camera.<br>2.Specify the surgical plan as a 6D pose and convert to world coordinates.<br>3. Manually move/tilt pelvis beyond the specified error threshold while ensuring that the pelvis marker remains the camera field of vision. | |
| **Validation** | |
| - The ROS script must detect and notify user on change in pelvis position when threshold is exceeded with a latency of <500ms. | |

## 4.10 Test 10

| Test 10: | |
|---|---|
| Motion Planning and Controls Integration Test | |
| **Objective** | |
| To integrate the planning and control subsystems such that the end effector follows the trajectory generated without exceeding the force threshold | |
| **Equipment** | MRSD System 2, Arm, Markers, Reaming tool |
| **Elements** | Planning & Controls |
| **Personnel** | Sundaram Seivur |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Move end-effector close to the site of operation.<br>2. Call node to generate trajectory using MoveIt between current point and end point.<br>3. Call control node to make the arm move along generated waypoints.<br>4. Measure and analyze if manipulator follows trajectory and reaches goal state within stated error limits and not exceeding force limits. | |
| **Validation** | |
| - Check visually and in simulation if arm starts following generated trajectory.<br>- Read force-torque sensor readings to validate that the applied forces are within the stated limits. | |

## 4.11 Test 11

| Test 11: | |
|---|---|
| Waypoint/Trajectory Generation Test | |
| **Objective** | |
| To test MoveIt to generate a trajectory from a start point(any point robot is left in space) to the end point | |
| **Equipment** | MRSD System 2, Arm, Markers, Reaming tool |
| **Elements** | Motion Planning |
| **Personnel** | Sundaram Seivur |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Move manipulator close to the pelvis model using free motion mode<br>2. Mark current pose as start point for manipulator.<br>3. Run ROS script to invoke MoveIt to generate trajectory between start and end poir | |
| **Validation** | |
| - Check in simulation if arm is moving along generated trajectory.<br>- Check visually if arm is moving along similar axis in reality. | |

## 4.12 Test 12

| Test 12: | |
|---|---|
| Waypoint Generation Compensation Test | |
| **Objective** | |
| To generate new trajectories when pelvis movement crosses a threshold of 1mm position error and 1.5 degress orientation error. The trajectory generated should be at a latency of less than or equal to 150ms | |
| **Equipment** | MRSD System 2, Arm, Markers, Reaming tool |
| **Elements** | Motion Planning |
| **Personnel** | Sundaram Seivur |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Read continously from perception node to detect a change in position and orientation of more than 1mm and 1.5 degrees respectively.<br>2. Call trajectory planning node to generate new trajectory. | |
| **Validation** | |
| - Physically move pelvis to new position to check if new trajectory is generated.<br>- Measure delay in generating new trajectory. | |

## 4.13 Test 13

| Test 13 : | |
|---|---|
| Position and Force Control in Simulation Test | |
| **Objective** | |
| To test the ability of the MPC controller to move to desired positions without exceeding a specified force in simulation before implementing in reality | |
| **Equipment** | System with Hipster Test Environment (MRSD Desktop 2) |
| **Elements** | Controls & Actuation Subsystem |
| **Personnel** | Anthony Kyu |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Setup simulation environment with manipulator and an obstacle (a simple block).<br>2. Set and send a desired end effector pose to the manipulator that is within the obstacle.<br>3. Measure and analyze the pose and force of the end effector over the duration of the simulation. | |
| **Validation** | |
| - If the force during the simulation ever exceeds the force threshold (< 30N), the test fails. This is evaluated quantitatively.<br>- If the difference from the final axial (z) position to the desired axial (z) position is minimized while still maintaining its x and y positional constraints (+/- 2mm), the test passes. This is evaluated quantatively for x and y error and qualitatively for z axis. | |

## 4.14 Test 14

| Test 14 : | |
|---|---|
| Position and Force Control in Reality Test | |
| **Objective** | |
| To test to see if the MPC controller performs as expected in reality, matching specs and requirements when testing in simulation | |
| **Equipment** | Wrist Force/Torque Sensor, Robotic Manipulator, Obstacle, Atracsys Camera, Marker Mounts, System with Hipster Test Environment (MRSD Desktop 2) |
| **Elements** | Controls & Actuation Subsystem |
| **Personnel** | Anthony Kyu |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Setup UR5 arm with ROS Moveit and place an obstacle (rectangular block) within the workspace of the arm.<br>2. Set and send a desired end effector pose to the manipulator that is within the obstacle.<br>3. Measure and analyze the pose and force of the end effector over the duration of the test (30 seconds). | |
| **Validation** | |
| - If the force during the test ever exceeds the force threshold (< 30N), the test fails. This is evaluated quantitatively.<br>- If the difference from the final axial (z) position to the desired axial (z) position is minimized while still maintaining its x and y positional constraints (+/- 2mm), the test passes. This is evaluated quantatively for x and y error and qualitatively for z axis. | |

## 4.15 Test 15

| Test 15 : | |
|---|---|
| Free Motion Mode Test | |
| **Objective** | |
| To test the ability of the manipulator to allow the an external agent to move the end effector freely, moving the arm without gravitational resistance, and without the controller trying to hold a single position | |
| **Equipment** | Robotic Manipulator, Force/Torque Sensor, System with Hipster Test Environment (MRSD Desktop 2) |
| **Elements** | Controls & Actuation Subsystem, State Machine Design |
| **Personnel** | Anthony Kyu |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Set up UR5 with Free Motion Mode.<br>2. Have personnel push the end-effector in random directions within the workspace.<br>3. Qualitatively assess whether the end-effector moves without resistance (except for possible arm momentum), and without the need for the personnel to provide gravitational assistance. | |
| **Validation** | |
| - The robot end-effector moved with ease or without resistance (qualitatively).<br>- The robot arm didn't need gravitational assistance from the personnel (qualitatively).<br>- The joints of the arm moved in a predictable manner (qualitatively). | |

## 4.16 Test 16

| 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<br>2. Attach Panavise mount to the manipulator table and clamp the sawbone pelvis in place<br>3. Attach marker array to the reamer assembly as well as to the pelvis<br>4. Place the Atracsys camera in an orientation to be able to view both the reamer and pelvis marker array<br>5. Actuate the robot manipulator using a built in controller to follow a similar motion path to what would be expected during the procedure<br>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<br>- Camera is capable of seeing both marker arrays throughout the procedure<br>- Reamer is powered and capable of removing material from the sawbone pelvis | |

## 4.17 Test 17

| Test 17 : | |
|---|---|
| Reamer Motor Speed and Torque Test | |
| **Objective** | |
| To test the torque and speed of the motor and gearbox and verify it's ability to output the necessary torque for the reamer to properly function to ream the acetabulum | |
| **Equipment** | System with Hipster Test Environment (MRSD Desktop 2), motor and gearbox from reamer assembly, power supply, video camera, long piece of wood, weights |
| **Elements** | Reaming subsystem of the hardware system |
| **Personnel** | Parker Hill |
| **Location** | NSH Basement |
| **Procedure** | |
| 1. Hook up the motor to the power supply and increase the applied voltage to 24V<br>2. Using a video camera, measure the approximate no load speed<br>3. Turn off the power supply and attach a long piece of wood of a specified measured length to the motor shaft<br>4. Add weight to the end of the piece of wood and measure the applied torque, turn on the power supply and determine if the motor is capable of moving past an orientation where the wood is parallel to the floor<br>5. Repeat step 4, increasing weight each time until the motor is incapable of moving past the specified orientation, mark the resulting torque as the peak torque of the motor<br>6. Using the no load speed and peak torque, approximate the max torque at a speed of 400 rpm | |
| **Validation** | |
| - With a speed of approximately 400 rpm, the associated maximum torque of the system should be greater than 1 Nm | |

### 4.18 Spring Validation Experiment

**Objective**: The general goal for this semester is that by the spring validation, all our subsystems should be working together in simulation. This includes the perception and sensing subsystem, the motion planning subsystem, and the controls subsystem. The spring validation experiment should see these subsystems which were tooled to work in simulation, be ported over to working with the physical system. Thus, with these experiments, we hope to validate the functionality of all of our individual subsystems on the physical robot manipulator.

**Equipment**: All equipment mentioned in Logistics will be utilized during the testing.

**Elements**: Perception and sensing subsystem, motion planning subsystem, controls subsystem

**Personnel**: Entire Hipster Team

**Location**: NSH Basement

**Procedure and Validation for Each Test**:

1. **Sensing and Perception Test**
   - **Procedure**:
     (a) Place a marker on the robot's end-effector.
     (b) Record the robot's end-effector pose through encoder values and transformation matrices. Record time to get end-effector pose.
     (c) Record the end-effector marker's pose from the camera.
     (d) Place a marker on a plane and record its initial position.
     (e) Move the marker to 3 new positions and record the time needed to get new marker positions.
     (f) Record computed error for the 3 new marker positions.
   - **Validation**:
     (a) The 2 recorded poses must match with an error $\leq$ 3mm and orientation error $\leq$ 3 degrees.
     (b) The robot must record new marker positions with a latency $\leq$ 500ms.
     (c) The robot must compute errors for each new marker position.
2. **Motion Planning and Controls Test**
   - **Procedure**:
     (a) Command the robot to go to an end-point.
     (b) Record the time taken for the robot to generate trajectory.
     (c) Run the Quantitative Trajectory Evaluator and examine the results folder to determine if the results are acceptable according to our performance requirements.

(d) Measure the force output over the trajectory and ensure that it does not go above the maximum force threshold.

- **Validation**:

(a) The robot end-effector must generate a new trajectory within 500ms.
(b) The robot end-effector reaches the end point within a threshold without moving through singularities.
(c) The maximum error at any point during the trajectory must be within a certain error threshold.
(d) The maximum force at any point during the trajectory must be less than the maximum force threshold (30N).

# 5 Appendix

## 5.1 Functional and Performance Requirements

| Functional Requirement | Performance Requirement | Justification |
|---|---|---|
| **M.F.1** The system shall localize the robot arm in real-time with respect to the pelvis before and during surgery | **M.P.1.1** The system will localize the robot arm in real-time with respect to the pelvis before and during surgery with a latency less than or equal to 50 ms | Latency of Atracsys Sprytrack 300 is less than 25ms; Processing time about 25 ms |
| | **M.P.1.2.1** The system will localize the robot arm in real-time with respect to the pelvis before and during surgery with a position error of less than 1 mm | Survey sent to surgeons and literature review suggest a desired position error of less than 2 mm. Combining **M.P.1.2.1** and **M.P.3.1** will result in a combined position error of less than 2 mm. |
| | **M.P.1.2.2** The system will localize the robot arm in real-time with respect to the pelvis before and during surgery with an orientation error of less than 1.5-degrees | Survey sent to surgeons and literature review suggest a desired orientation error of less than 3-degrees. Combining **M.P.1.2.2** and **M.P.3.2** will result in a combined orientation error of less than 3-degrees. |
| **M.F.2** The system shall plan the trajectory of the robot arm based on the given surgical plan | **M.P.2** The system will plan the trajectory of the robot arm based on the given surgical plan with a latency less than or equal to 150 ms | Total latency of the system should be less than 500 ms. |
| **M.F.3** The system shall execute surgical plan by reaming along the generated trajectory | **M.P.3.1** The system will execute surgical plan by reaming along the generated trajectory with an position error of less than 1 mm | Survey sent to surgeons and literature review suggest a desired position error of less than 2 mm. Combining **M.P.1.2.1** and **M.P.3.1** will result in a combined position error of less than 2 mm. |
| | **M.P.3.2** The system will execute surgical plan by reaming along the generated trajectory with an orientation error of less than 1.5-degrees | Survey sent to surgeons and literature review suggest a desired orientation error of less than 3-degrees. Combining **M.P.1.2.2** and **M.P.3.2** will result in a combined orientation error of less than 3-degrees. |
| **M.F.4** The system shall compute error and interpret the movement of the pelvis during reaming | **M.P.4.1** The system will compute error and interpret the movement of the pelvis during reaming with a latency less than or equal to 50 ms | Latency similar to localization |
| | **M.P.4.2** The system will generate a new trajectory if the interpreted position and orientation errors are greater than 1 mm or greater than 1.5-degrees. | Survey sent to surgeons and literature review suggest a desired position and orientation error of less than 2 mm and 3-degrees. Therefore, the thresholds for compensating for these errors should be less than these desired errors. |
| **M.F.5** The system shall adapt and compensate for movement by generating a new trajectory | **M.P.5** The system will adapt and compensate for movement by generating a new trajectory with a latency less than or equal to 150 ms | Latency similar to trajectory planning |
| **M.F.6** The system shall allow the surgeon to place the robot arm at an initial position | **M.P.6** The system will allow the surgeon to place the robot arm to an initial position by back-driving the robotic arm | Reduce system complexity by keeping path to be planned short |
| **M.F.7** The system shall provide the surgeon with visual feedback | **M.P.7** The system will provide the surgeon with visual feedback with a latency less than or equal to 150 ms | From literature on tele-surgery, latency 150 ms is found to be noticeable to surgeons, and degrades performance of surgeon-performed tasks |
| **M.F.8** The system shall allow the surgeon to e-stop | **M.P.8** The system will allow the surgeon to e-stop the system, stopping the system within 500 ms | Competitor systems have similar quantification |

### 5.2 Non-functional Requirements

**M.N.1** The system will produce forces low enough for it to be safe around humans.

**M.N.2** The system will provide a minimal and easy-to-interpret user interface design for surgeons.

**M.N.3** The system will autonomously detect malfunctions and errors and notify user accordingly.

**D.N.1** The system will allow for numerous successful surgeries, without the need for servicing and calibration.

**D.N.2** The system will have a cost comparable to similar systems on the market.

**D.N.3** The system will adhere to all relevant ISO standards pertaining to medical robotic systems.

**D.N.4** The system will be of a size and dimension that is ergonomic.

**D.N.5** The system will be designed such that it can be serviced easily.

**D.N.6** The system will be designed to be easily sterilizable or sterile in the sterile field.