# Conceptual Design Review Report

## Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Team C:

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

December 13, 2021

Carnegie Mellon University
The Robotics Institute

# Contents

# 1 Project Description

Total Hip Arthroplasty (THA) Surgery as performed in present-day involves many steps, including cutting the femoral head, drilling into the femur, placing the femoral stem into the femur, reaming the acetabulum, and placing an acetabular cup into the reamed acetabulum. One of the most crucial factors in determining a successful surgery is the accuracy of acetabular cup position and orientation, as the correct position and orientation would decrease likelihood future dislocation of the hip joint and increase patient comfort [1]. Therefore, it is imperative that surgeons know exactly what depth they are reaming the acetabulum to and the what orientation the acetabular cup is placed at.

However, most surgeons cannot see the site of surgery well during surgery and do not use the proper tools to obtain accurate results, leading to malpositioned cups. In fact, it is estimated that less than 50% of THA outcomes are within surgical safe zones (such as the Lewinnek Safe Zone, a constraint that increases surgery success) [2]. This is a result of "intraoperative pelvic tilt, distorted anatomical landmarks, and limited accuracy and reproducibility of the alignment guides." Other factors of malpositioned cups are "minimally invasive surgical approaches, low surgeon volume, and obesity" [1]. While there are modern robotic systems that can help mitigate this problem, all of them increase the time of the surgery takes and lack robustness.

"More recently, patient-specific safe zones based on preoperative assessments of pelvic kinematics have gathered momentum as a route for improving stability and reducing complications in THA" [3]. With the manual surgery, these plans would be hard to execute with high accuracy. However, autonomous robotic technology can provide a way to execute patient-specific surgical plans to meet these patient-specific safe zones with high accuracy [1]. To meet this need, our team is proposing the ARTHuR Robot (Autonomous Reaming for Total Hip Replacement Robot), which will maximize the accuracy of acetabular reaming, while remaining robust during surgery.

# 2 Use Case

## 2.1 Use Case Narrative

Dr. Williams is an orthopedic surgeon and needs to perform a total hip replacement/arthroplasty on a patient suffering from osteoarthritis in their hip joint. Prior to the beginning of the surgery, Dr. Williams takes a CT scan of the patient's pelvis in order to reconstruct the 3-dimensional geometry of the patient's pelvis, specifically their acetabulum. With this 3D geometry, Dr. Williams is able to use his intuition to choose an acetabular cup specifically suited for the patient, and using a 3-dimensional surgical planner, plan the exact location and orientation of the acetabular cup, such that the resulting prosthesis will be in the Lewinnek Safe Zone.

Dr. Williams then begins the surgery by orienting the patient to their side such that the hip which is being operated on is facing upwards. Dr. Williams, using typical surgical technique for hip replacement surgeries, cuts the femoral stem to expose the acetabulum and cleans the desired amount of soft tissue from the pelvis. With the acetabulum exposed, the surgeon then drills a reflective marker array into the pelvis which is used to locate the pelvis in 3 dimensional space with a tracking camera. Using a registration probe with a reflective marker array attached to it, the surgeon registers the position of 3-5 points of interest on the pelvis and then generates a point cloud that represents the anatomy of the patient's pelvis. This point cloud and the points of interest are then used to localize the hip in 3-dimensional space, fitting the 3D geometry from the CT scan to the patient's pelvis on the operating table. Given this localization and the desired acetabular cup location, a cutting trajectory is then generated which a robotic arm will follow in order to ream the acetabulum as desired. Dr. Williams then fixes a reamer onto the end-effector of the robotic arm. From here, the doctor manipulates the robotic arm such that the end-effector is near where the reamer would begin cutting into the acetabulum. Dr. Williams then starts the robotic arm using the surgeon's user interface, which begins to autonomously track the trajectory and ream the acetabulum according to the surgical plan as can be seen in the fourth image in Figure 1. During the reaming, the robot arm detects and compensates for movements in the patient's pelvis that occur as a result of the applied forces, maintaining the surgical plan. Furthermore, Dr. Williams is provided with visual feedback on a monitor, which demonstrates the current progress in reaming the acetabulum. He also has access to an emergency stop button which can cut power to the robot.

Once the robot arm has completed the planned trajectory, it stops and allows for Dr. Williams to remove it from the surgical site so they can analyze the resulting reamed acetabulum. Once the reaming operation is deemed a success, Dr. Williams can then proceed with the rest of the operation by first hammering the selected acetabular cup into the reamed acetabulum. Dr. Williams then cuts into the femur and implants the femoral stem of the hip implant into it, before inserting the femoral head into the acetabular cup to complete the prosthetic hip joint. With the femoral stem and acetabular cup connected, Dr. Williams is free to stitch the patient up and send them on their way home, happy and pain-free.

### 2.2 Use Case Graphical Representation

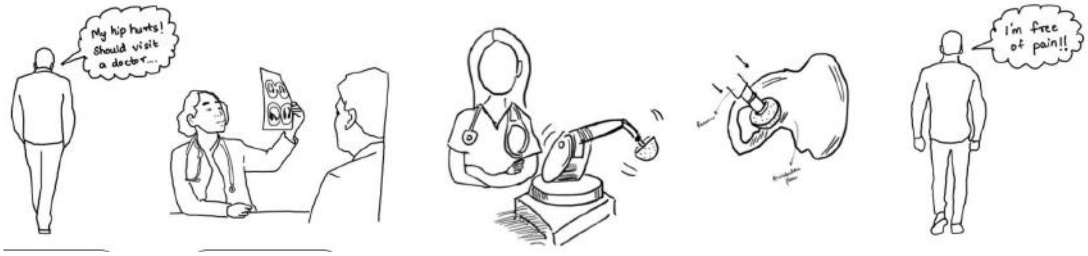


Figure 1: Use Case Graphical Representation

## 3 System-Level Requirements

The following requirements were elicited from discussions with our sponsor and with surgeons, while also factoring in the given resources and time.

### 3.1 Mandatory System-Level Requirements

The mandatory requirements are further split into Functional & Performance Requirements (Table 1) and Non-Functional Requirements.

### 3.1.1 Mandatory Functional & Performance Requirements

**Table 1: Mandatory Functional & 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 [4]. 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 [4]. 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 [4]. 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 [4]. 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 [4]. 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 |

### 3.1.2 Mandatory 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.

## 3.2 Desired System-Level Requirements

The desired requirements are further split into Functional & Performance Requirements (Table 2) and Non-Functional Requirements.

### 3.2.1 Desired Functional & Performance Requirements

**Table 2: Desired Functional & Performance Requirements**

| Functional Requirement | Performance Requirement | Justification |
|---|---|---|
| **D.F.1** The system shall allow surgeon to change end-effector/tool | **D.P.1** The system will allow surgeon to change end-effector/tool in 3 steps or less | During surgery, some surgeons would prefer to step up reamers sizes instead of using the final size. Allowing them to do this in 3 steps or will allow the surgeon to quickly and change tools during surgery. |
| **D.F.2** The system shall position acetabular cup for surgeon to impact into reamed acetabulum | **D.P.2** The system will position acetabular cup for surgeon to impact into reamed acetabulum with less than 3-degree error. | According to surgeon surveys and literature review, 3-degree error is the maximum allowable error. Since the reaming determines positional error, positional error is not a concern for impaction. |

### 3.2.2 Desired Non-Functional Requirements

**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.

# 4 Functional Architecture

Figures 2 and 3 below show the initial and revised versions of our functional architecture.

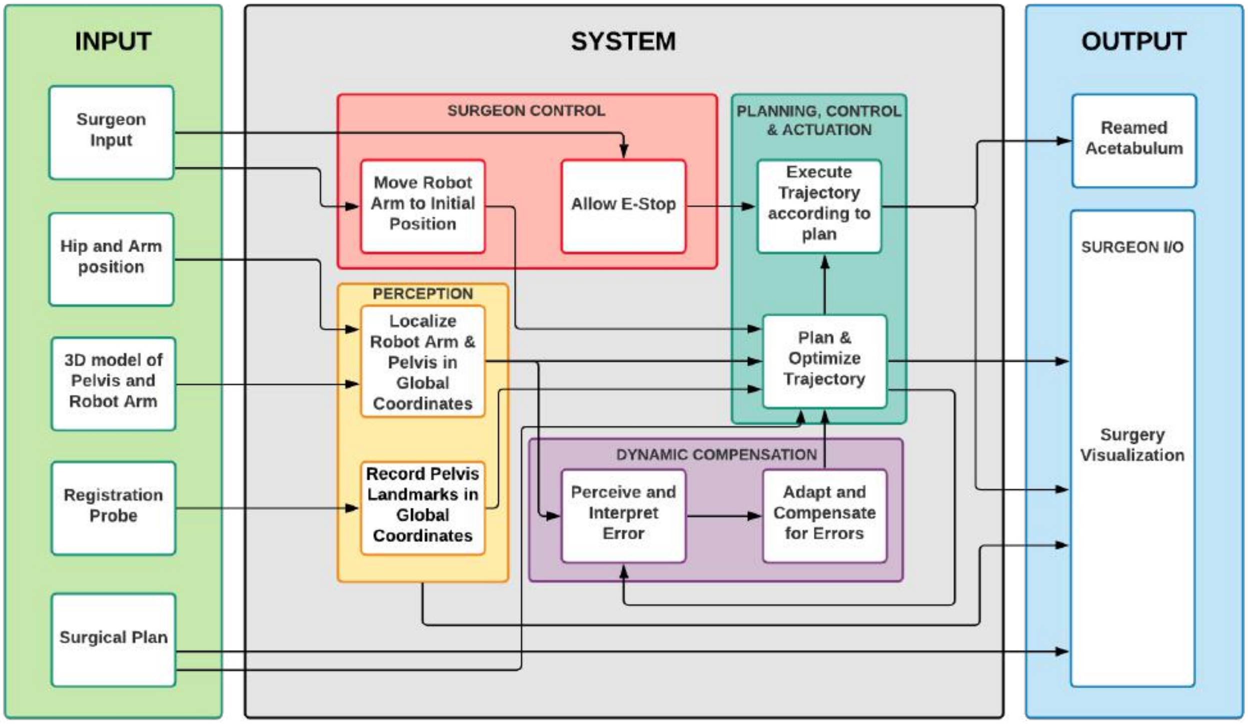

INPUT
SYSTEM
OUTPUT
Surgeon Input
Hip and Arm position
3D model of Pelvis and Robot Arm
Registration Probe
Surgical Plan
SURGEON CONTROL
Move Robot Arm to Initial Position
Allow E-Stop
PLANNING, CONTROL & ACTUATION
Execute Trajectory according to plan
Plan & Optimize Trajectory
PERCEPTION
Localize Robot Arm & Pelvis in Global Coordinates
Record Pelvis Landmarks in Global Coordinates
DYNAMIC COMPENSATION
Perceive and Interpret Error
Adapt and Compensate for Errors
Reamed Acetabulum
SURGEON I/O
Surgery Visualization

**Figure 2: Initial Draft of the Functional Architecture**

The functional architecture embedded within a process flow diagram is as shown in Figure 3. This revision was made so as to highlight the temporal components and dependencies in our processes effectively, which wasn't as evident in the initial draft. In the revised functional architecture, the horizontal axis represents the time at which each process occurs. Broadly, the diagram is divided into three sections that represent the inputs, the system and the outputs. Their spatial position along the x-axis represents the timing of their occurrence. Two processes placed one below the other indicates that both happen in tandem. The **inputs** are as follows:

1. The hip and arm positions inferred by the marker positions using the camera.
2. The 3D model of the the pelvis of the patient obtained pre-operatively, such as using a CT-Scan.
3. 3D model of the robot arm.
4. The surgical plan provided in the form of the pose of the acetabular implant in the pelvis.
5. The registration probe used by the surgeon to match key landmarks of the patient's pelvis to the 3D model obtained pre-operatively.

6. The surgeon's input to initialize the position of the robot arm, and control the e-stop button.

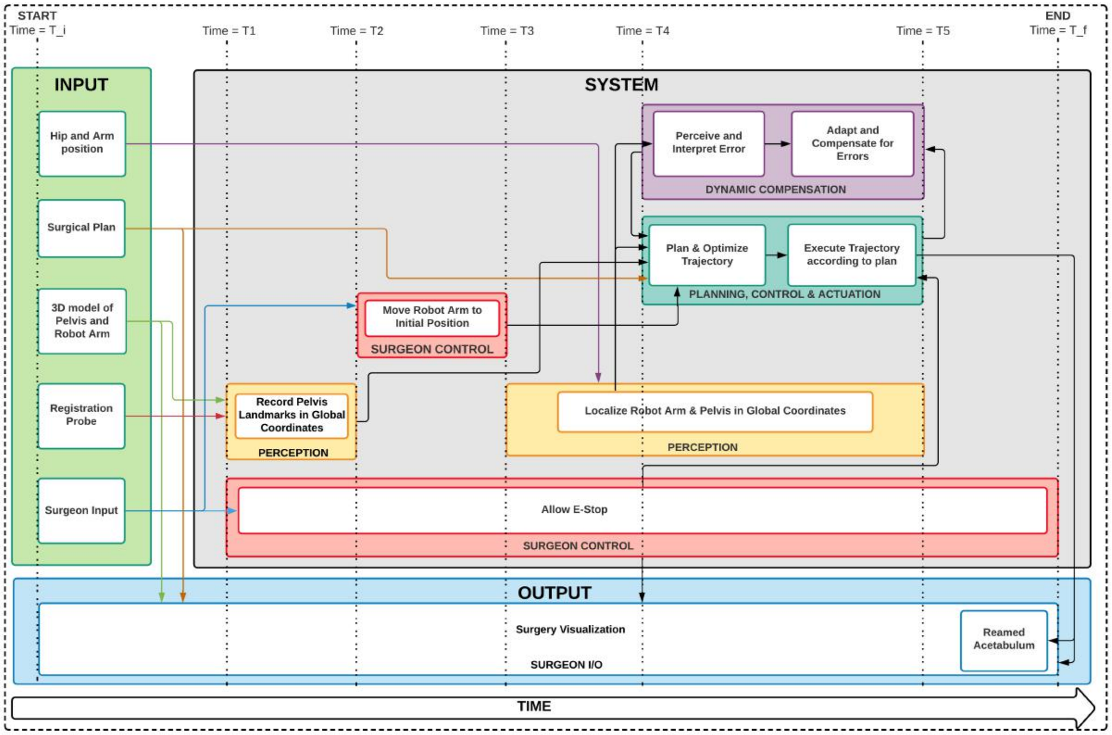

START
Time = T_i
Time = T1
Time = T2
Time = T3
Time = T4
Time = T5
END
Time = T_f
INPUT
Hip and Arm position
Surgical Plan
3D model of Pelvis and Robot Arm
Registration Probe
Surgeon Input
SYSTEM
Perceive and Interpret Error
Adapt and Compensate for Errors
DYNAMIC COMPENSATION
Plan & Optimize Trajectory
Execute Trajectory according to plan
PLANNING, CONTROL & ACTUATION
Move Robot Arm to Initial Position
SURGEON CONTROL
Record Pelvis Landmarks in Global Coordinates
PERCEPTION
Localize Robot Arm & Pelvis in Global Coordinates
PERCEPTION
Allow E-Stop
SURGEON CONTROL
OUTPUT
Surgery Visualization
SURGEON I/O
Reamed Acetabulum
TIME

**Figure 3: Revised Functional Architecture Embedded in a Process Flow Diagram**

In the **system** block, a registration probe is used by the surgeon to record the pelvis landmarks as the first step. Next, the arm is moved by the surgeon to the start position where the acetabulum has been exposed for the reaming process. Once moved, the arm is localized by the perception system in the global coordinates and continues to be localized throughout the reaming operation. The planning, control and actuation as well as the dynamic compensation blocks are placed one below the other, indicating that both processes are happening simultaneously and depend on one another. Based on the start and end points computed by the planning block, an optimized trajectory is determined and executed. Simultaneously, the dynamic compensation module ensures that the end point is recomputed in the event that the pelvis moves from its initial position upon being impinged by the reaming tool.

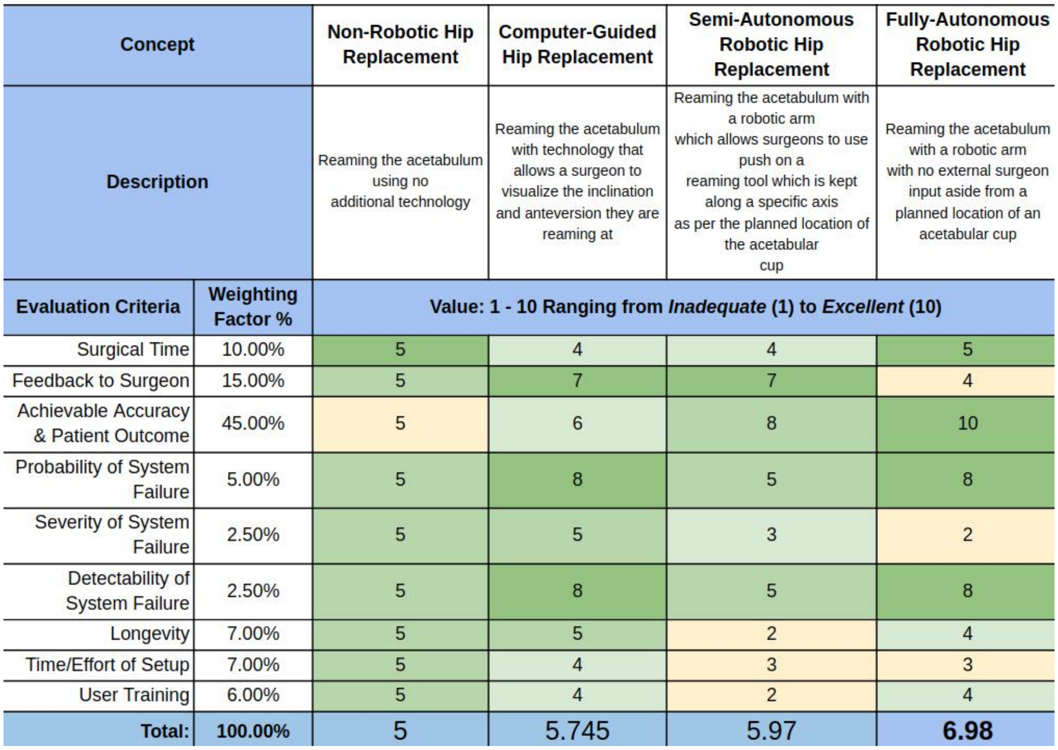
Finally, the **output** is the reamed acetabulum as per the accuracy defined by our performance requirements. Throughout this process, the various steps and current status of the surgery can be visualized on a monitor which formed the surgeon I/O component of our system. An enlarged version of this functional architecture can be found in the appendix in figure 17.

## 5 System and Subsystem-Level Trade Studies

We conducted a primary system-level trade study and two sub-system level trade studies to decide the overall functionalities of our system. The performance of each alternative in the trade study was valued on a scale of 1-10. Each candidate was evaluated based on the expected performance matching the required performance/functional requirements and weights were assigned accordingly. The highest scoring candidate was chosen, which is highlighted in **bold**.

### 5.1 System-Level Trade Study: Investigating Levels of Autonomy

The goal of robotics and automation in the medical industry is to make medical procedures safer and inexpensive for patients by assisting doctors to perform complex procedures with greater precision, flexibility and control. In this regard, we conducted a system-level trade study where we compared the performance and impact of varying levels of autonomy in modern day surgeries. In our study, we evaluated the pros and cons of four systems, namely: non-robotic hip replacement, computer guided hip replacement, semi-autonomous robotic hip replacement and fully-autonomous robotic hip replacement. The primary goal of our system is to improve the accuracy of acetabular reaming in total hip replacement while providing the surgeon with adequate feedback on the status of the procedure. Hence, these criteria were weighted higher in comparison to other criteria such as surgical time, system failure and effort of setup among others. The study, as seen in Figure 4, elucidated that a fully-autonomous robot would have the greatest impact on accuracy of reaming and fared better than the other alternatives. A higher resolution version of this trade study can be found in the appendix as Figure 19.



| **Concept** | | **Non-Robotic Hip Replacement** | **Computer-Guided Hip Replacement** | **Semi-Autonomous Robotic Hip Replacement** | **Fully-Autonomous Robotic Hip Replacement** |
|---|---|---|---|---|---|
| **Description** | | Reaming the acetabulum using no additional technology | Reaming the acetabulum with technology that allows a surgeon to visualize the inclination and anteversion they are reaming at | Reaming the acetabulum with a robotic arm which allows surgeons to use push on a reaming tool which is kept along a specific axis as per the planned location of the acetabular cup | Reaming the acetabulum with a robotic arm with no external surgeon input aside from a planned location of an acetabular cup |
| **Evaluation Criteria** | **Weighting Factor %** | **Value: 1 - 10 Ranging from *Inadequate* (1) to *Excellent* (10)** | | | |
| Surgical Time | 10.00% | 5 | 4 | 4 | 5 |
| Feedback to Surgeon | 15.00% | 5 | 7 | 7 | 4 |
| Achievable Accuracy & Patient Outcome | 45.00% | 5 | 6 | 8 | 10 |
| Probability of System Failure | 5.00% | 5 | 8 | 5 | 8 |
| Severity of System Failure | 2.50% | 5 | 5 | 3 | 2 |
| Detectability of System Failure | 2.50% | 5 | 8 | 5 | 8 |
| Longevity | 7.00% | 5 | 5 | 2 | 4 |
| Time/Effort of Setup | 7.00% | 5 | 4 | 3 | 3 |
| User Training | 6.00% | 5 | 4 | 2 | 4 |
| **Total:** | **100.00%** | 5 | 5.745 | 5.97 | **6.98** |

**Figure 4: System Level Trade Study: Levels of Autonomy**

## 5.2 Sub-system Level Trade Study: Inverse Kinematics Packages

This sub-system level trade study delves into the comparison between the various inverse kinematics packages we could implement in our system. The candidates chosen are as follows:

1. **IKFast**: It is a powerful inverse kinematics solver [5] provided within the OpenRAVE [6] motion planning software. Unlike most inverse kinematics solvers, IKFast can analytically solve the kinematics equations of any complex kinematics chain, and generate language-specific files (like C++) for later use.

2. **The Kinematics and Dynamics Library (KDL)**: Developed by OROCOS, KDL [7] is an application independent framework for modelling and computation of kinematic chains, such as robots, biomechanical human models, computer-animated figures, machine tools, etc.

3. **TRAC-IK**: TRAC-IK [8] is an alternative Inverse Kinematics solver to OROCOS' KDL. It varies from KDL in that it has two IK solver implementations - one having an improved convergence algorithm to KDL and the second used Sequential Quadratic Programming (SQP) nonlinear optimization approach. By default, the IK search returns immediately when either of these algorithms converges to an answer.

For the evaluation criteria, the robustness of the solver defined in terms of its average success rate (30%) and number of unique solutions (20%) form dominant weighting criteria. This is because having a failure-free path is mission-critical during the reaming operation. Similarly, another key element we needed to consider was latency and performance for the dynamic compensation to work effectively. So, the speed of solver is given a 30% weightage. The other two factors are the level of integration with ROS and Moveit! (15%), as well as documentation and support available for implementation(5%). As with the system level trade study, the scores are weighted on a scale of 1-10, and scores were assigned based on literature reviews [8] that compared these alternatives. The alternative that fared much better than the others was the IKFast plugin which we will integrate into our system. The primary reason was that unlike the other two alternatives, the IKFast is an analytical solver. It also provides extremely stable solutions that can be found in a few microseconds on recent processors. This meets our project's criteria effectively. A higher resolution version of this trade study can be found in the appendix as Figure 20.

## 5.3 Sub-system Level Trade Study: Visualization Toolkits

As part of the Surgeon I/O subsystem, there is a requirement to choose a visualization toolkit to render the robot state and reaming process updates onto a GUI. For this trade study, we examine the possibility of using two of the most popular visualization toolkits. Both of these softwares are developed and maintained by KitWare.

1. **3D Slicer**: 3D Slicer ("Slicer") is an open source, extensible software platform for image visualization and analysis. Slicer has a large community of users in medical imaging and surgical navigation, and is also used in fields such as astronomy, paleontology, and 3D printing.

2. **Visualization ToolKit (VTK)**: The Visualization Toolkit (VTK) is open source software for manipulating and displaying scientific data. It comes with state-of-the-art tools for 3D rendering, a suite of widgets for 3D interaction, and extensive 2D plotting capability.

We have used a three point criteria to arrive at the optimal choice of the visualization software. Our first and most important criteria was how well they integrate with PLUSToolKit and ROS. This criteria receives the highest weightage of 35%. A higher score is given to 3D Slicer since it integrates directly with PLUSToolKit through SlicerIGT and with ROS through ROS-IGTL-Bridge. However, VTK needs to be integrated with Insights ToolKit (ITK) before it can talk to PLUSToolKit and ROS-IGTL-Bridge. The next important criteria was the generality of the software. 3D Slicer gets a higher score since it uses both ITK and VTK. The former is used for image processing and image I/O while the latter is used for its 2D and 3D rendering pipelines, linear and non-linear transformations, segmentation infrastructure and more. However, VTK is limited in its capabilities without integrating with ITK. The Final criteria was to consider the depth of documentation and support availability. Since both the tools are well-documented and supported by the developers, they receive an equal score.

In conclusion, 3D Slicer appears to be an optimal choice for the visualization software. Albeit, it is important to note that the scope of this trade study is limited by the prior experience of the team members. We will continue reaching out to industry experts to expand our knowledge on these softwares and compare them in further detail. A higher resolution version of this trade study can be found in the appendix in Figure 21.

### 5.4 Other Potential Trade Studies

A large majority of our project has been predetermined as a result of our sponsorship. Whereas if this was an academically focused project we would have had to make more hardware decisions, our sponsor has decided to fund us and provide us with the hardware to complete our project as makes most sense for them. As a result, where we might have had a large trade study to determine whether to go with a Kuka, a Kinova, or a Universal Robotics robotic arm for our project, our sponsor is providing us with a Kinova Link-6 arm to work with, negating the need to conduct a trade study. In a similar vein, we have been provided with an Atracsys Sprytrack 300 to use as our primary camera, and will be provided with a surgical reamer to be attached to the end-effector of the Kinova Link-6 arm. As a result, the trade studies we had to conduct were mostly software package decisions. One potential future trade study could be studying the benefits of various control methods.

## 6 Cyberphysical Architecture

Figure 5 below shows the cyberphysical architecture for our system. The major components of the cyberphysical architecture are derived from the functional architecture. On a high-level, the various block are inputs, sensing, perception, planning, control actuation and surgeon I/O. We also have the power-supply for each module, and a watch-dog that ensures that the various sub-modules are functioning as expected. Any block here with a dashed border such as those seen in the input block do not directly translate to a work-package, and those with a solid border will contribute to a work-package in the work breakdown structure. The blue arrows indicate mechanical linkages, the black arrows indicate information flow and the red arrows are power flow. An enlarged version of this architecture can be found in the appendix.

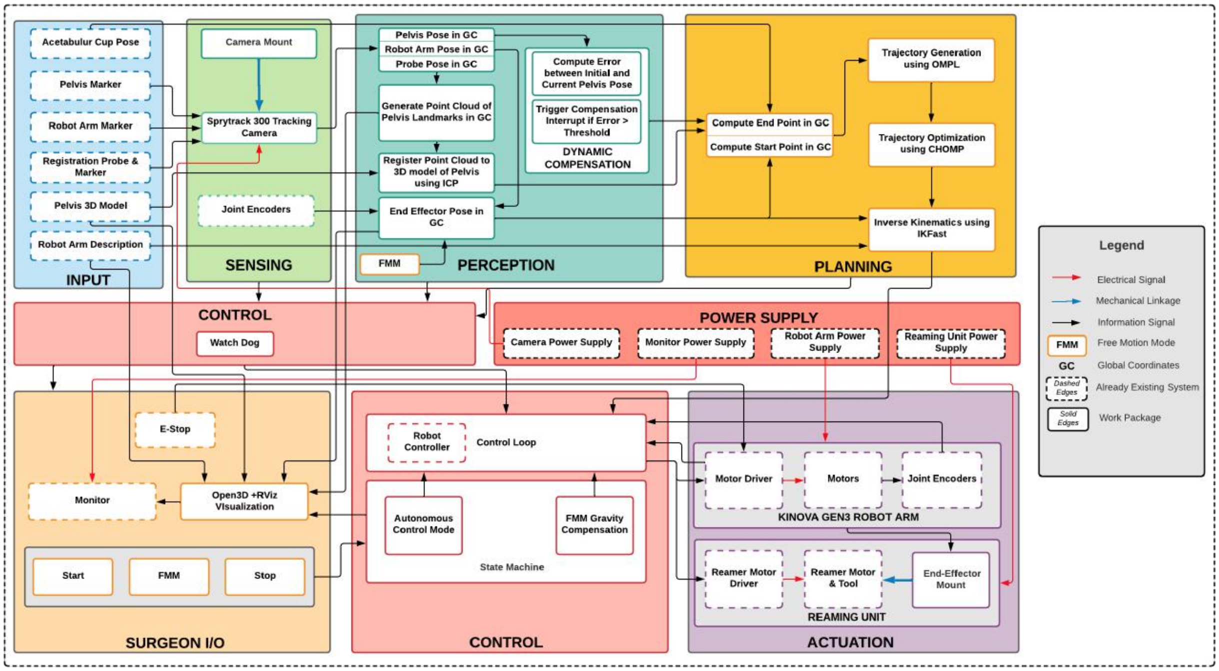


**Figure 5: Cyberphysical Architecture**

The following are brief descriptions of each sub-system involved in the cyberphysical architecture:

1. **Input**: There are six distinct inputs in the input block which have been detailed from the input block in our functional architecture. Three of these are derived from the reflective markers attached on the pelvis, robot arm and the registration probe respectively. The surgical plan in the form of the acetabular cup pose is used to compute the reaming trajectory. The final inputs are the pelvis and robot arm descriptions.

2. **Sensing**: The sensing module is primarily composed of the joint encoders used to compute the joint positions of each joint of the arm. The marker positions are determined using the Sprytrack 300 camera, which is our primary sensing component. This will be securely mounted to externally overlook the arm and the pelvis during the surgery.

3. **Perception**: Within the perception block, the sensor data from the Sprytrack camera is used to determine the pose of the patient's pelvis, robot arm and registration marker in a global frame of reference. The pose of the end effector pose is updated using the robot arm marker's pose combined with encoder data from the robot's joints. A one time process of landmark acquisition happens at the beginning of each surgery where key landmarks from the patients pelvis are captured to form a point cloud. This resulting point cloud is then fit to the 3D

model of the pelvis obtained pre-operatively as an input into the system. The dynamic compensation module is used to continuously monitor the error between the current pelvis pose and the initial pelvis pose. Should this error exceed a pre-defined threshold, an interrupt is triggered to ensure that the planning module compensates for this movement.

4. **Planning**: The start and end points of reaming is computed based on the surgeon input and the surgical plan. Using this, the reaming trajectory to be followed is generated using the Open Motion Planning Library (OMPL), and optimized using the Covariant Hamilton Optimization for Motion Planning (CHOMP). The optimized trajectory is then used an input to IKFast which returns the desired joint angles for the joint to execute. If a compensation interrupt is triggered from the perception block, the end point of reaming is re-computed to compensate for the motion of the pelvis during surgery.

5. **Control**: The control module is composed of two modes which will be set by the Surgeon I/O. A state machine is used switch between Autonomous Control Mode or Free Motion Mode (FMM). If the state machine is in Autonomous Control Mode, then the control loop will take inputs from the planned trajectory and use those inputs in the control loop. If the state machine is in Free Motion Mode, then the control loop will be using gravity compensation to keep the robot in place unless moved manually by the surgeon. A watch-dog module continuously monitors and takes overriding controls actions should any sub-system level malfunctions.

6. **Actuation**: The control loop sends current signals to the motor drivers, which in turn will be used to drive the motors. The joint positions from the encoders will be constantly monitored as feedback to the control loop. The reaming unit in the actuation block will be controlling the spindle rate and torque of the reamer during the reaming operation. The reamer will be mounted onto the end-effector of the robot arm mechanically.

7. **Surgeon I/O**: The surgeon would be provided with a visualization that they can monitor throughout the procedure. A control interface was also included to allow the surgeon to control the operating mode of the robot through the state machine in the control module. An E-Stop will be available to them at all times for stopping the robot arm at any time during the procedure.

# 7 Subsystem Descriptions

## 7.1 Inputs

There are six distinct inputs in the input block. The first is the 3D model of the pelvis, typically obtained from a CT scan pre-operatively. The second is the robot arm description which is used later by Moveit! for inverse kinematics and motion planning. Three sets of passive fiducial markers form the next three inputs. Each marker has an associated geometry [9] which is known beforehand. An example of a reflective marker used is as shown in Figure 6. One of these markers is drilled into the patient's pelvis by the surgeon. Another marker is attached to the base of the robot arm using a mount. The final marker is used to capture key landmarks on the patient's pelvis and construct a point cloud which will later be used for registration with the pre-operatively built 3D model of the pelvis. The final input into the system is the surgeon's input, which will control the sequence of operations using the surgeon IO and also exercise the use of a emergency stop button.

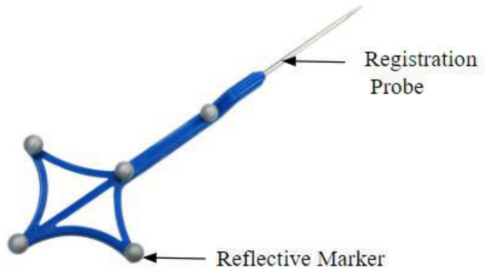


**Figure 6: Reflective Marker [9]**

The final input is the surgical plan in the form of the 6-dimensional acetabular cup pose which the surgeon decides prior to the surgery.

## 7.2 Perception & Sensing

The Sprytrack 300 camera [10], as shown in Figure 7, forms the primary sensing component. Its operation is summarized in a flow diagram as shown in Figure 8. The pose of the end effector pose is updated using the robot arm marker's pose combined with encoder data from the robot's joints. The Sprytrack 300 camera is composed of two cameras designed to detect and track 6D pose of fiducials with sub-millimeter accuracy in real time video streams. The IR Illuminators first emit IR light, which are reflected by passive markers. Sprytrack 300 then detects the reflected light and transmits this data to the host computer. The 3D position of each fiducial is computed and matched to the marker geometry and the the rigid body 6D pose is computed for user post-processing.

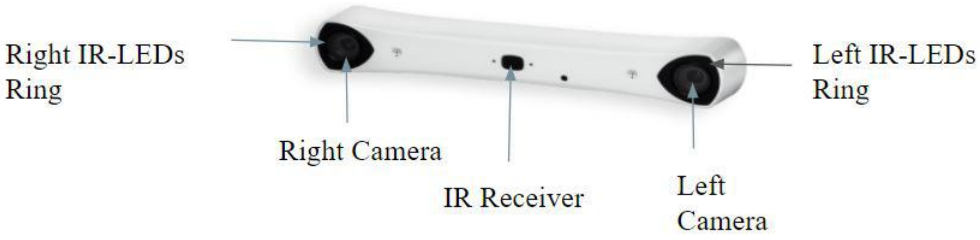


**Figure 7: Sprytrack 300 Camera [10]**

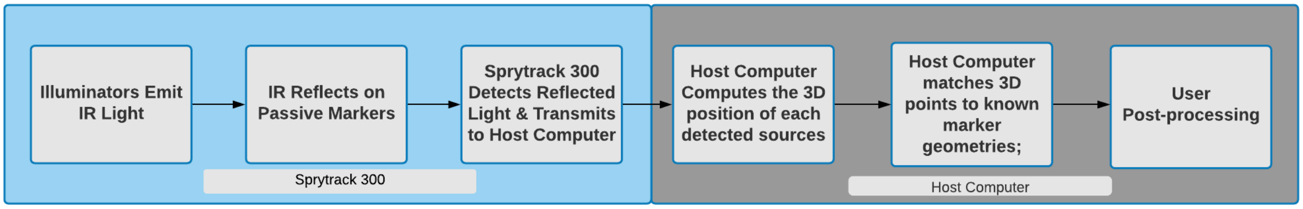

Illuminators Emit IR Light
IR Reflects on Passive Markers
Sprytrack 300 Detects Reflected Light & Transmits to Host Computer
Sprytrack 300
Host Computer Computes the 3D position of each detected sources
Host Computer matches 3D points to known marker geometries;
User Post-processing
Host Computer

**Figure 8: Functional Flow Diagram of Sprytrack 300 Camera**

**Registration** is the process where the surgeon captures key landmarks from the surface of the acetabulum using a registration probe shown in Figure 6, and associates to the landmarks on the pelvis model obtained pre-operatively as an input. The host computer is used to construct a point cloud using this data. Finally, the captured point-cloud is registered to the 3D model of the patient's pelvis obtained pre-operatively using a CT Scan. The capabilities of the Insight Toolkit (ITK) will be used to perform the registration process. Figure 9 shows this process taking place on commercial systems [11].

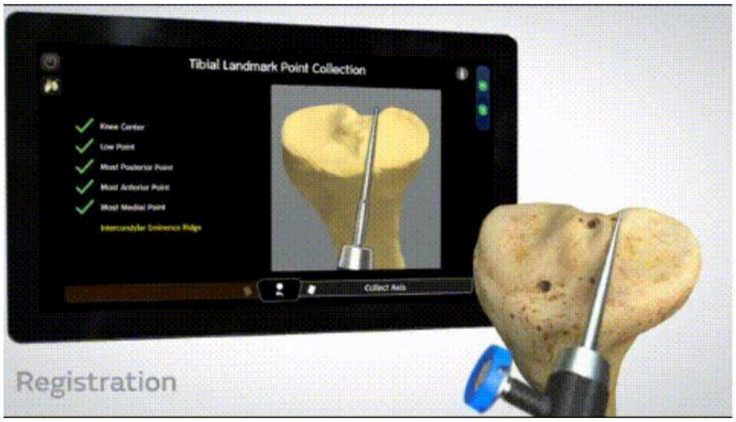


**Figure 9: Registration Process taking place on commercial systems**

**Dynamic compensation** is the task of adapting the planned trajectory based on any motion of the patient and consequently the pelvis from its initial position. During total hip replacement surgery, the forces acting on the patient while reaming are high, causing the patient to move. This moving of the patient leads to inaccuracies while reaming based on the surgical plan. To tackle this problem, our autonomous solution would be constantly checking for any movement of the patient above a certain threshold and would re-plan the trajectory of reaming if that threshold were to be crossed. This threshold would be calculated based on the maximum allowable error as defined by the performance requirements. The flow of decisions for dynamic compensation is seen in Figure 10.

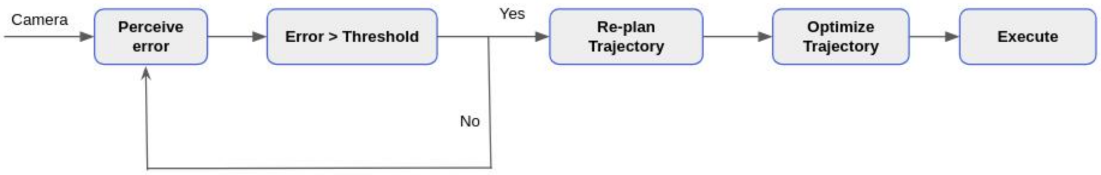


**Figure 10: Dynamic Compensation process flow**

### 7.3 Motion Planning

A trajectory for reaming is generated between a starting point, which is defined by where the robot arm is left in space and the end point which is defined by the desired depth of reaming. These points are fed to ROS for trajectory generation using OMPL, which is The Open Motion Planning Library, and trajectory optimization using CHOMP, which is Covariant Hamilton Optimization for Motion Planning. The optimized trajectory is then used an input to IKFast which returns the desired joint angles.

The planning block as a whole works with the MoveIt motion planning framework within ROS. The MoveIt framework acts as a nexus between the perception, planning, control and actuation blocks as seen in Figure 11. Information about the robot arm such as robot model and joint states along with the marker pose as seen by the camera are fed into MoveIt. Using this, MoveIt would initialize the planning node for trajectory planning and optimization using OMPL and CHOMP. This data is sent back to MoveIt which then initializes the Inverse Kinematics and Collision detection plugins within the control block. Finally, MoveIt would interface with the actuation block to control the actuators of the robot arm.

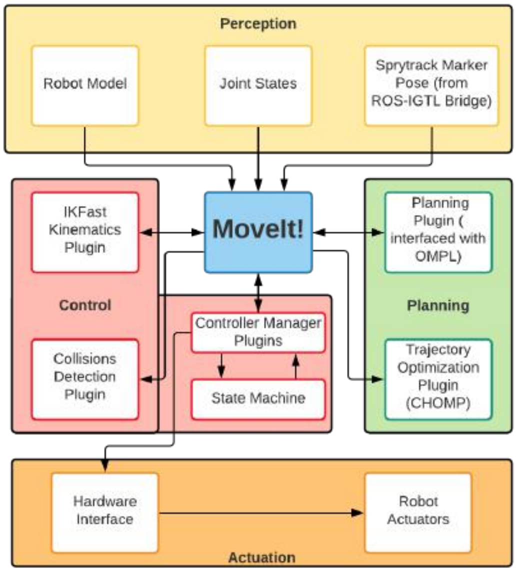

Perception
Robot Model
Joint States
Sprytrack Marker Pose (from ROS-IGTL Bridge)
IKFast Kinematics Plugin
MoveIt!
Planning Plugin ( interfaced with OMPL)
Control
Planning
Controller Manager Plugins
Collisions Detection Plugin
State Machine
Trajectory Optimization Plugin (CHOMP)
Hardware Interface
Robot Actuators
Actuation

**Figure 11: ROS architecture for Planning using MoveIT**

## 7.4 Control & Actuation

The Control subsystem of the ARTHuR system will be determined by several different inputs. At the highest level, the Control subsystem is the controlled by the state machine set by the Surgeon I/O. As shown in the Cyberphysical Architecture (Figure 5), the Surgeon I/O has the FMM (Free Motion Mode) button, which switches the state of the state machine between Autonomous Control Mode or Free Motion Mode (FMM). The state of the state machine determines what inputs the control loop takes.

When the state machine is in **Autonomous Control Mode**, the control loop will take inputs from the planned trajectory in the form of desired joint angles. The control loop will then use force-based control to move the arm along the trajectory. This use of force-based control will allow the actuation subsystem to have adjustable feed rate to avoid high forces on the acetabulum to limit the risk of fractures, and to satisfy requirement **M.N.1** to produce low enough forces be safe around humans. Previous surgical robotic systems have used force-based control for the aforementioned reasons [12]. The control loop will then receive positional and velocity feedback from the joint encoders of the robotic arm, and current feedback to derive torque feedback for each joint from the motor drivers.

When the state machine is in **Free Motion Mode**, the control loop will not receive inputs from the planned trajectory, and will instead receive feedback only through the joint encoders from the robotic arm. Based on the joint angles, desired torques will be calculated, informing the desired currents that the motor drivers will send to the joint motors. In this mode, the surgeon will be able to set the initial position of the robot arm, greatly reducing the complexity of trajectory planning because the robot arm will not have to plan from moving outside the body to the acetabulum, satisfying requirement. **M.F.6**. Furthermore, in this mode, the robot arm will stay in place unless the surgeon moves it by hand. A flow diagram of Free Motion Mode is shown in Figure 12.

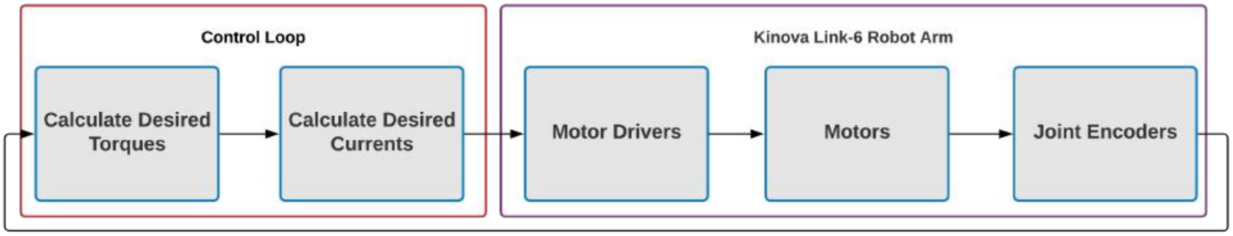


**Figure 12: Free Motion Mode Flow Diagram**

The Actuation subsystem will contain two discrete units: the Kinova Link-6 Robotic Arm and the Reaming Unit. At the time of this report submission, our sponsor has not shared the specifications of the Link-6 Robotic Arm, but it is known that the arm will be a 6-DoF arm with joint encoders. The reaming unit will be attached to the end-effector of the arm mechanically and will be the tool cutting the acetabulum.

The Watch Dog module within the Control subsystem is a module that checks the system for malfunctions and errors, and reports them to the Surgeon I/O module, satisfying requirement **M.N.1**. In addition, in case of error, the watchdog module will inform the control loop to react accordingly based on the error. In case the Watch Dog module fails to detect a system failure, the surgeon will have an E-Stop available to them to cut power to the arm at any time.

### 7.5 Surgeon I/O

The Surgeon I/O subsystem is used by the doctor to visualize the current state of the robot and stay updated about the reaming process. It allows two major visualizations. First is an interactive one made using 3D Slicer. The interactive GUI will be developed using 3D Slicer that would enable the surgeon to register landmarks, manage the state of the robot and view the current states of the pelvis and robot in 3D in real time. The second is made through RViz, which displays the state of the robot and its parameters in detail. This is mainly for development purposes. Figure 13 demonstrates a general idea of what the resulting user interface would look like for a surgeon.

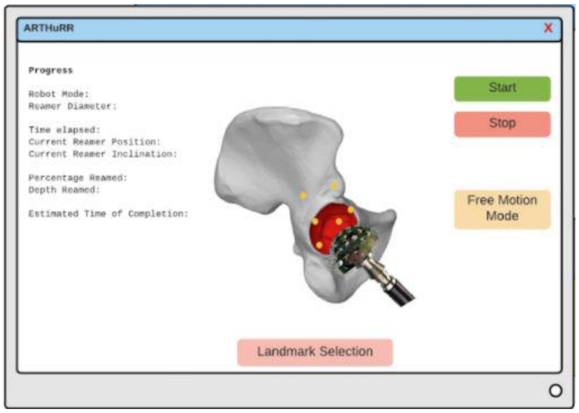


**Figure 13: Surgeon I/O Visualization**

## 8 Project Management

### 8.1 Work Plan and Tasks

Figure 14 depicts the high level Work Breakdown Structure required to execute and build the ARTHuR system successfully. These work packages were further broken down into lower level work packages for the schedule in Figure 16 for the sake of defining what work would need to be done every week. Generally the work is broken down into 8 categories: Hardware Setup, Simulation Setup, Perception & Sensing, Control & Actuation, Planning, Surgeon I/O, System Integration, and Management. Hardware Setup would largely involve getting the camera and robot arm setup for use while Simulation Setup would involve getting our ROS environment and packages configured and consistent across all our devices. Perception & Sensing, Control & Actuation, Planning, and Surgeon I/O would all comprise of developing those subsystems as they exist in the Cyberphysical Architecture in Figure 5 and would include the development of these subsystems first in simulation and then in reality on the physical arm. Systems integration would then be a large work package which we anticipate taking the majority of early Fall 2022, which would involve integrating all developed subsystems together to develop a full working system on the robot arm. Finally, we anticipate Management to be a large work package as well which would involve consistent work by our Project Manager who would consistently check our progress against the work we against our schedule and analyze potential risks and mitigation actions.

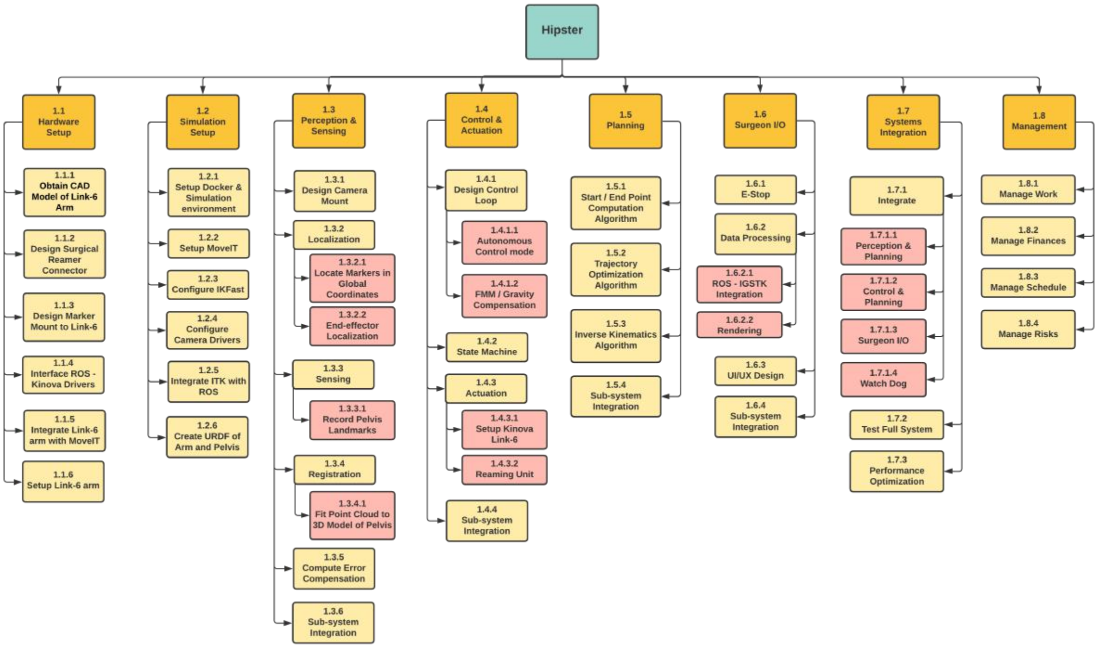

Hipster
1.1 Hardware Setup
1.1.1 Obtain CAD Model of Link-6 Arm
1.1.2 Design Surgical Reamer Connector
1.1.3 Design Marker Mount to Link-6
1.1.4 Interface ROS - Kinova Drivers
1.1.5 Integrate Link-6 arm with MoveIT
1.1.6 Setup Link-6 arm
1.2 Simulation Setup
1.2.1 Setup Docker & Simulation environment
1.2.2 Setup MoveIT
1.2.3 Configure IKFast
1.2.4 Configure Camera Drivers
1.2.5 Integrate ITK with ROS
1.2.6 Create URDF of Arm and Pelvis
1.3 Perception & Sensing
1.3.1 Design Camera Mount
1.3.2 Localization
1.3.2.1 Locate Markers in Global Coordinates
1.3.2.2 End-effector Localization
1.3.3 Sensing
1.3.3.1 Record Pelvis Landmarks
1.3.4 Registration
1.3.4.1 Fit Point Cloud to 3D Model of Pelvis
1.3.5 Compute Error Compensation
1.3.6 Sub-system Integration
1.4 Control & Actuation
1.4.1 Design Control Loop
1.4.1.1 Autonomous Control mode
1.4.1.2 FMM / Gravity Compensation
1.4.2 State Machine
1.4.3 Actuation
1.4.3.1 Setup Kinova Link-6
1.4.3.2 Reaming Unit
1.4.4 Sub-system Integration
1.5 Planning
1.5.1 Start / End Point Computation Algorithm
1.5.2 Trajectory Optimization Algorithm
1.5.3 Inverse Kinematics Algorithm
1.5.4 Sub-system Integration
1.6 Surgeon I/O
1.6.1 E-Stop
1.6.2 Data Processing
1.6.2.1 ROS - IGSTK Integration
1.6.2.2 Rendering
1.6.3 UI/UX Design
1.6.4 Sub-system Integration
1.7 Systems Integration
1.7.1 Integrate
1.7.1.1 Perception & Planning
1.7.1.2 Control & Planning
1.7.1.3 Surgeon I/O
1.7.1.4 Watch Dog
1.7.2 Test Full System
1.7.3 Performance Optimization
1.8 Management
1.8.1 Manage Work
1.8.2 Manage Finances
1.8.3 Manage Schedule
1.8.4 Manage Risks

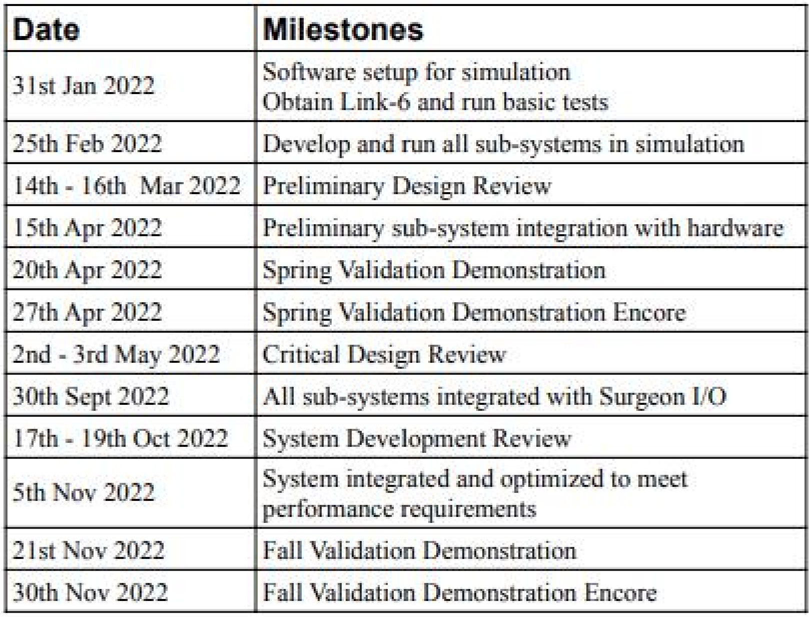
**Figure 14: Work Breakdown Structure**

## 8.2 Schedule and Key Milestones

Milestones have been set for the spring semester and high-level milestones have been set for fall semester as shown in Table 3. The project is planned to be completed in a sequential manner going from building each sub-system on simulation, incorporating them individually on the arm, integrating all of them to work in tandem with the arm, optimizing the system performance to match requirements, and finally validate the system before demonstration. The external milestones for both the semesters are the design reviews and validation demonstrations. We have three internal milestones for the duration of the spring semester in January, February and April. The first internal milestone is to complete the software setup for simulation, obtain the arm from sponsors and evaluate the basic functionalities of the arm.

For the first progress review we will have all sub-systems running in simulation before integrating them with the arm. At the end of the spring semester we would have all sub-systems except 'Surgeon I/O' running on the arm. We have two internal milestones in the fall semester in which we would complete system integration with hardware and optimize the system for better performance respectively. At the end of the fall semester, we would demonstrate autonomous reaming of the acetabulum for total hip replacement surgery. All sub-systems are broken down into detailed tasks in the project schedule for both spring and fall semesters as seen in Figure 15 and Figure 16. The schedules are also attached in the appendix(23, 24) for better clarity.

Table 3 : Key milestones for spring and fall semesters



| Date | Milestones |
|---|---|
| 31st Jan 2022 | Software setup for simulation<br>Obtain Link-6 and run basic tests |
| 25th Feb 2022 | Develop and run all sub-systems in simulation |
| 14th - 16th Mar 2022 | Preliminary Design Review |
| 15th Apr 2022 | Preliminary sub-system integration with hardware |
| 20th Apr 2022 | Spring Validation Demonstration |
| 27th Apr 2022 | Spring Validation Demonstration Encore |
| 2nd - 3rd May 2022 | Critical Design Review |
| 30th Sept 2022 | All sub-systems integrated with Surgeon I/O |
| 17th - 19th Oct 2022 | System Development Review |
| 5th Nov 2022 | System integrated and optimized to meet performance requirements |
| 21st Nov 2022 | Fall Validation Demonstration |
| 30th Nov 2022 | Fall Validation Demonstration Encore |

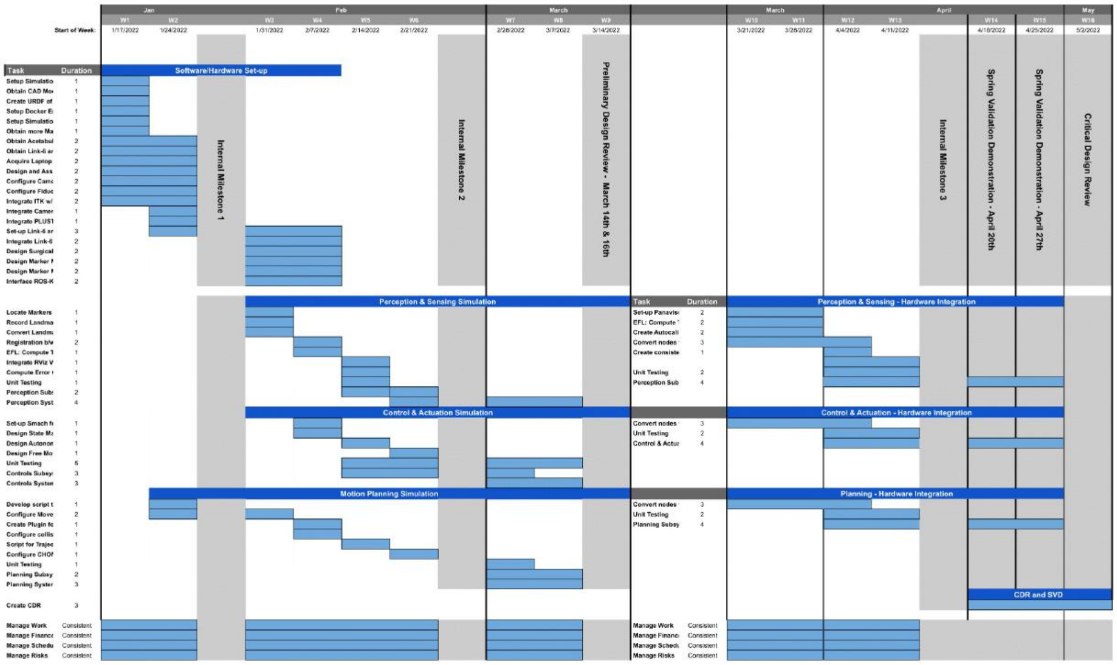


Figure 15: Project schedule for spring semester

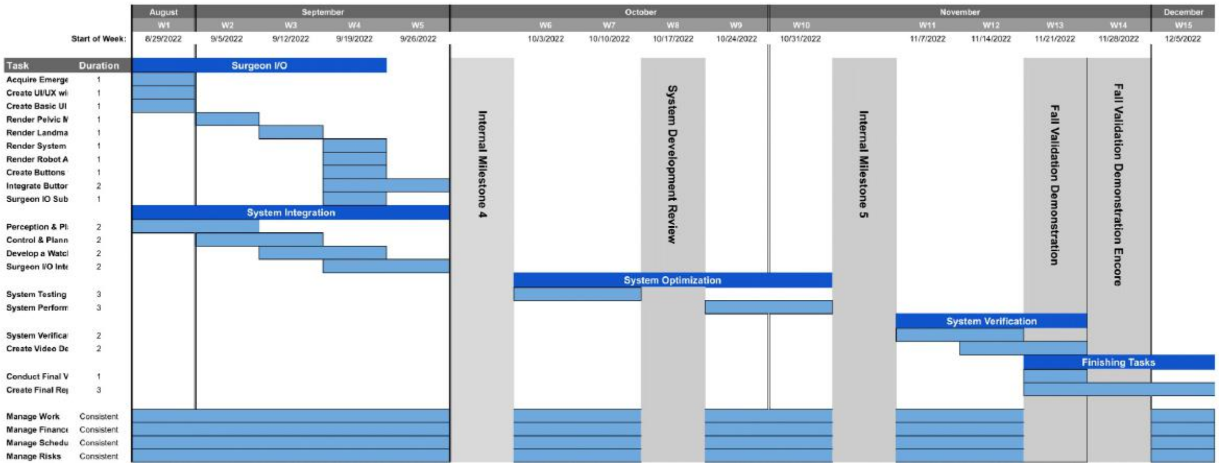


Figure 16: Project schedule for fall semester

## 8.3 System Validation Experiments

The goal of the Spring Validation Demonstration is to showcase our subsystems working individually on the robot arm, with some relaxed performance requirements. Our efforts towards integrating the subsystems and optimizing their performance to achieve the latency and performance requirements will be carried out during Fall. The validation experiments for each subsystem will look as follows.

- **Location:** B512, Newell-Simon Hall
- **Equipment:** Kinnova Link-6 Robot Arm, Pelvic Sawbones, Test Jig, Atracsys SpryTrack 300 Camera, Reflective Markers, and Monitor.

### 8.3.1 Spring Validation Experiment

1. **Perception and Sensing Validation**:

   (a) **Requirements**:

   - **M.P.1.1-3** The system shall localize the robot arm in real-time with respect to the pelvis before and during surgery with a latency $\leq$ 500ms, position error of $\leq$ 3mm and orientation error $\leq$ 3 degrees.
   - **M.P.4** The system shall compute error and interpret the movement of the pelvis during reaming with a latency $\leq$ 500 ms.

   (b) **Procedure**:

   - Place a marker on the robot's end-effector.
   - Record the robot's end-effector pose through encoder values and transformation matrices. Record time to get end-effector pose.
   - Record the end-effector marker's pose from the camera.
   - Place a marker on a plane and record its initial position.

- Move the marker to 3 new positions and record the time needed to get new marker positions.
- Record computed error for the 3 new marker positions.

(c) **Success Criteria**:

- The 2 recorded poses must match with an error $\leq$ 3mm and orientation error $\leq$ 3 degrees.
- The robot must record new marker positions with a latency $\leq$ 500ms.
- The robot must compute errors for each new marker position.

2. **Motion Planning and Controls/Actuation Validation**:

(a) **Requirements**:

- **M.P.2.** The system shall plan the trajectory of the robot arm based on the given surgical plan with a latency $\leq$ 500ms.
- **M.P.3.1, M.P.3.2.** The system shall execute a surgical plan by reaming along the generated trajectory with a position error of $\leq$ 3mm and orientation error $\leq$ 3 degrees.
- **M.P.5.** The system shall adapt and compensate for movement by generating a new trajectory with a latency $\leq$ 500ms.

(b) **Procedure**:

- Command the robot to go to an end-point.
- Record the time taken for the robot to generate trajectory.
- Run the Quantitative Trajectory Evaluator and examine the results folder

(c) **Success Criteria**:

- The robot end-effector must generate a new trajectory within 500ms.
- The robot end-effector reaches the end point within a threshold without moving through singularities.
- The maximum error at any point during the trajectory must be within a certain error threshold.

3. **Free Motion Mode Validation**:

(a) **Requirements**:

- **M.P.6.** The system shall allow the surgeon to place the robot arm to an initial position by back-driving the robotic arm.

(b) **Procedure**:

- Select 3 points marked as “initial” points in the physical world.
- Gently pull the end-effector’s tool point center to the marked “initial” points.

(c) **Success Criteria**:

- The robot arm back-drives to the marked “initial” points.

### 8.3.2 Fall Validation Experiment

The Fall Validation Experiments will validate the entire system consisting of the fully-integrated subsystems with optimized performance. The experiment will be set up with a test jig consisting of a pelvis that emulates the patient's movement during a surgery. To begin the test, we place the robot arm near the center of the acetabular floor using the FMM Mode. Successful initial alignment validates **M.P.6**. We would then press the "start" button and allow for the robot to localize itself. We would then record the latency of the initial pose localization. The robot must determine its pose with respect to the pelvis within 50ms, within a pose and orientation error of 1mm and 1.5 degrees respectively. This would validate **M.P.1.1**, **M.P.1.2.1**, **M.P.1.2.2**, and **M.P.4.2**. We would also record the time taken for the motion plan to be generated. The latency must be within 150ms. This would validate **M.P.2**. Now, the robot would execute the trajectory by reaming the Sawbones acetabulum. The trajectory must be followed within the defined root mean square error threshold. We would obtain these errors using the Quantitative Trajectory Evaluator. This would validate **M.P.3.1** and **M.P.3.2**. In the case that the center of the acetabulum or initially aligned point moves, we would record the latency of the robot indicating an error value between the two points and generating a new plan. This latency in computing error must be within 50ms and must generate a new plan within 150ms. This would validate **M.P.4.1** and **M.P.5**. The robot and the pelvis, in their current states, must be visible on the surgeon I/O, with a latency within 150 ms. This would validate **M.P.7**. During the reaming process, the robot will be asked to e-stop, it must stop the system within 500 ms.

## 8.4 Team Member Responsibilities

The responsibilities to complete the functions of the system has been divided between team members based on expertise and interests of individuals as shown in Table 4. Each sub-system has a primary and secondary owner to reduce the risks of a team member not being able to complete his/her responsibilities.

**Table 4: Team Member Responsibilities**

| Name | Primary Role | Secondary Role |
|---|---|---|
| Kaushik Balasundar | Perception and Sensing Lead | Software Engineer |
| Parker Hill | Mechanical Systems Engineering Lead | Perception and Sensing Engineer |
| Anthony Kyu | Controls and Actuation Lead | Mechanical Systems Engineer |
| Sundaram Seivur | Trajectory Planning Lead | Controls and Actuation Engineer |
| Gunjan Sethi | Software Engineering Lead | Perception and Sensing Engineer |

## 8.5 Parts List and Budget

The parts and budget list for the Spring semester is listed in Table 5. Although, all the necessary equipment to complete the project is provided to us by our sponsors, we are in need to buy model bones for testing and other accessories to complement the hardware. We are trying to retain a majority of the project funds for risk mitigation strategies involving hardware failure such as camera failure and robot arm malfunctioning.

**Table 5: Parts List and Budget**

| Item | Unit Cost($) | Quantity | Total Cost |
|---|---|---|---|
| Hemi Pelvis 48mm - Solid foam | 27.10 | 5 | 135.5 |
| Hemi Pelvis 56mm with vise attachment - Solid foam | 27.10 | 5 | 135.5 |
| Monitor Desk Mount Vesa | 42.40 | 1 | 42.4 |
| Panavise Mount for holding sawbones | 96.30 | 1 | 96.3 |
| Acetabulum Reamer Shaft | 700 | 1 | 700 |
| Acetabulum Reamer Shells | 100 | 1 | 100 |
| 24in x 48 in aluminium sheet | 59 | 2 | 118 |
| Reamer drill | 500 | 1 | 500 |
| Software Training | 60 | 5 | 300 |
| Total | | | $2127.7 |

## 8.6 Risk Management

There are many technical, schedule, and programmatic risks that could adversely effect the team's capability to complete the project successfully, all of which are summarized in Figure 25. However, there are three risks which, if not mitigated properly, could have a much more disastrous effect on the outlook of the project than the other risks. Further risk information can be found in the appendix in Figure 26, 27, 28, 29, 30.

1. **Robot arm does not arrive on time**: This is the first big risk that could have a big impact on the schedule of the project should it be realized. It is incredibly important that the Kinova Link-6 arrives on time so that there is enough time to focus on tuning the system from working in simulation to working in reality. To mitigate this risk, the schedule was devised such that the focus of the project is on simulation for the first half of the Spring semester, so that any delays take place while there is no need for the arm. Furthermore, through discussions with our sponsors, we have asked for the arm to be ordered as soon as possible, and it should hopefully arrive on or before January $20^{th}$ 2022.

2. **Robot arm breaks**: This is the second big risk that could either impact the schedule or the entire project as a whole should it be realized. If the robot arm breaks in such a way that is is fixable, the rest of the project would be delayed until the robot arm was completely fixed. However, if the robot arm breaks and is deemed unusable, then the entire project would need to be changed to simulation only. To mitigate this, we have decided to implement a rule in which code is only run on the robot arm after it has been verified in simulation. Furthermore, we have secured a safe work environment for the arm and plan on speaking with other professors to determine if we can use their arm in a worst case scenario.

3. **Performance requirements not met**: This is the final big risk which represents a programmatic risk to our entire project. Should our performance requirements not be met, the project could not be considered acceptable to our sponsors or surgeons, thus leading to it making little impact in the medical industry. The biggest mitigation action to avoid this risk involved conducting research into what our requirements reasonably should be, which led us to changing our requirements to be latency as opposed to frequency based. Furthermore, we plan on revisiting and updating our performance requirements every end-of-sprint meeting.

## References

[1] B. Kayani, S. Konan, A. Ayuob, S. Ayyad, and F. S. Haddad, "The current role of robotics in total hip arthroplasty," *EFORT Open Reviews*, vol. 4, no. 11, p. 618–625, 2019.

[2] L. J. L. T. R. C. C. L. . Z. J. R. Lewinnek, G. E., "Dislocations after total hip-replacement arthroplasties," *The Journal of Bone  Joint Surgery*, vol. 60, no. 2, p. 217–220, 1978.

[3] L. S. D. L. V. D. S. C. C. J. . M.-A. S. Rivière, C., "Spine–hip relations in patients with hip osteoarthritis," *EFORT Open Reviews*, vol. 3, no. 2, p. 39–44, 2018.

[4] Stryker, "Mako tha surgical technique." Available at https://www.strykermeded.com/media/2042/mako-tha-surgical-technique.pdf.

[5] OROCOS, "IKFAST kinematics solver." Available at http://docs.ros.org/en/kinetic/api/moveit_tutorials/html/doc/ikfast/ikfast_tutorial.html.

[6] R. Diankov, *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.

[7] OROCOS, "KDL overview — OROCOS documentation." Available at https://docs.orocos.org/kdl/overview.html/.

[8] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 928–935, IEEE, 2015.

[9] "IZI-medical disposable passive blunt probe." Available at https://izimed.com/products/disposable-passive-blunt-probe.

[10] "Sprytrack 300 product specifcations." Available at https://www.atracsys-measurement.com/products/sprytrack-180/.

[11] "CORI surgical system." Available at https://www.smith-nephew.com/professional/products/robotics/cori-surgical-system/.

[12] T. Haidegger, B. Benyó, L. Kovács, and Z. Benyó, "Force sensing and force control for surgical robots," *IFAC Proceedings Volumes*, vol. 42, no. 12, pp. 401–406, 2009. 7th IFAC Symposium on Modelling and Control in Biomedical Systems.

# 9 Appendix

### Full Size Functional Architecture

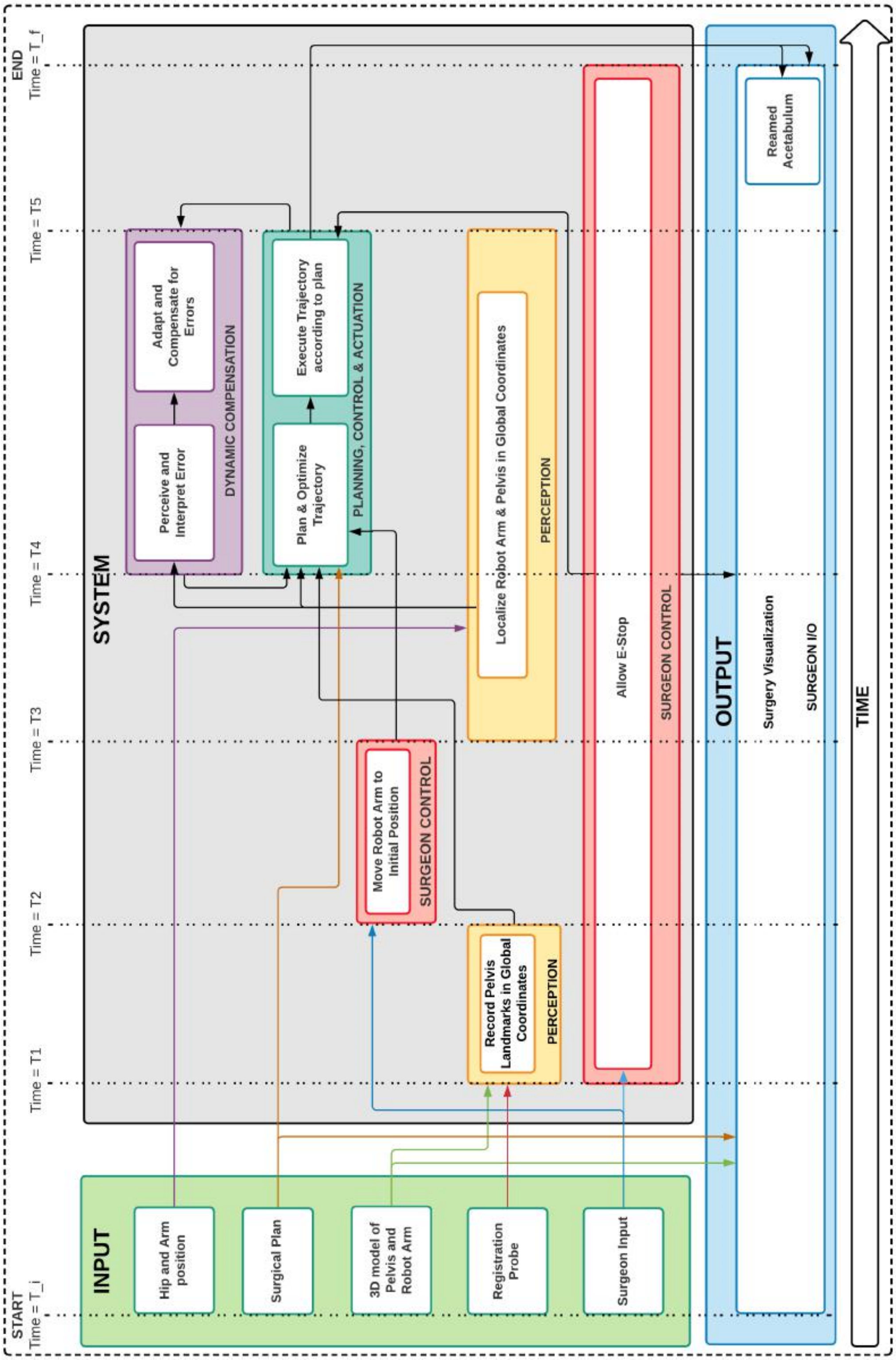

START
Time = T_i
Time = T1
Time = T2
Time = T3
Time = T4
Time = T5
END
Time = T_f
INPUT
Hip and Arm position
Surgical Plan
3D model of Pelvis and Robot Arm
Registration Probe
Surgeon Input
SYSTEM
Record Pelvis Landmarks in Global Coordinates
PERCEPTION
Move Robot Arm to Initial Position
SURGEON CONTROL
Localize Robot Arm & Pelvis in Global Coordinates
PERCEPTION
Perceive and Interpret Error
Adapt and Compensate for Errors
DYNAMIC COMPENSATION
Plan & Optimize Trajectory
Execute Trajectory according to plan
PLANNING, CONTROL & ACTUATION
Allow E-Stop
SURGEON CONTROL
OUTPUT
Surgery Visualization
SURGEON I/O
Reamed Acetabulum
TIME

Figure 17: Full Size Functional Architecture : Revised

## Full Size Cyberphysical Architecture

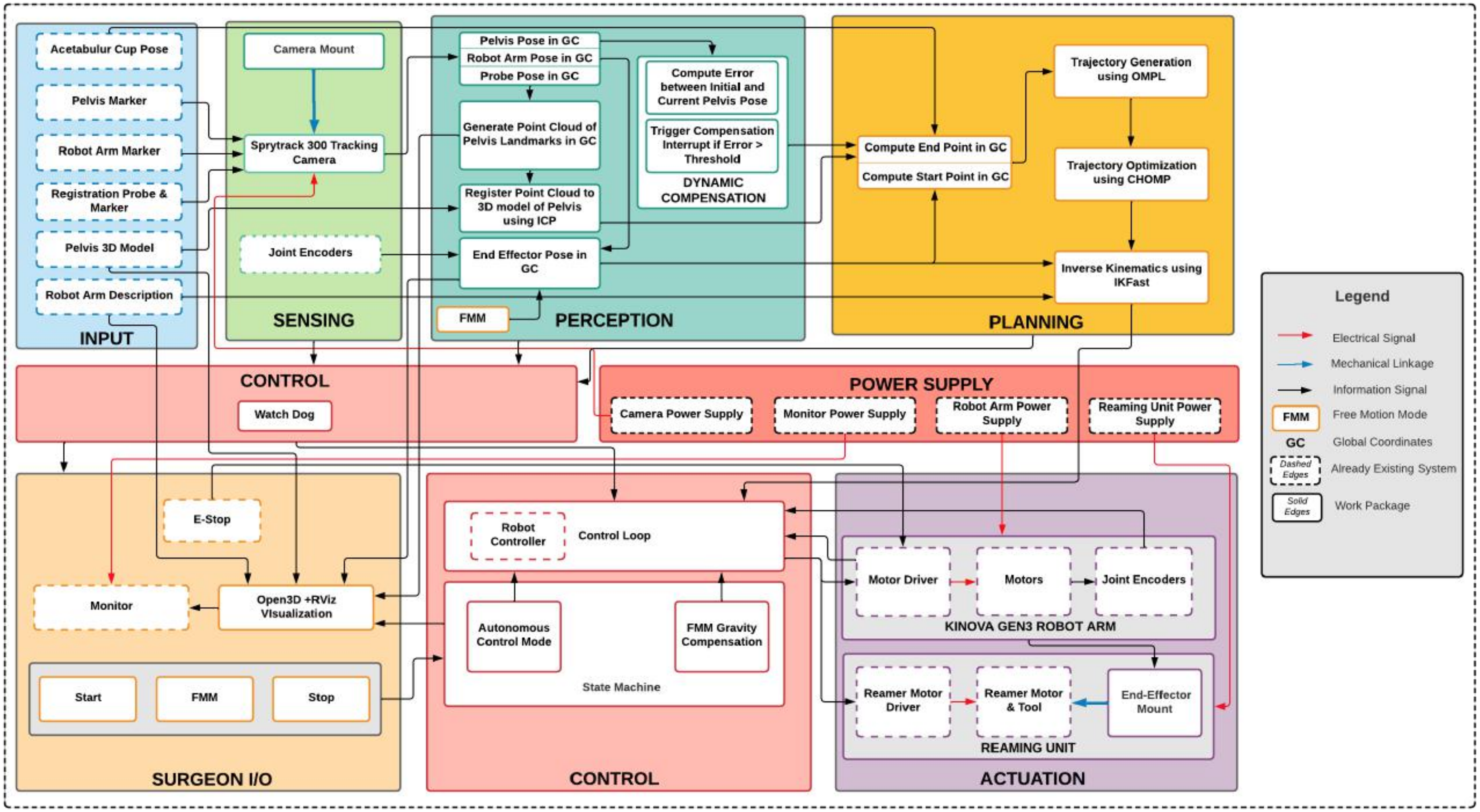
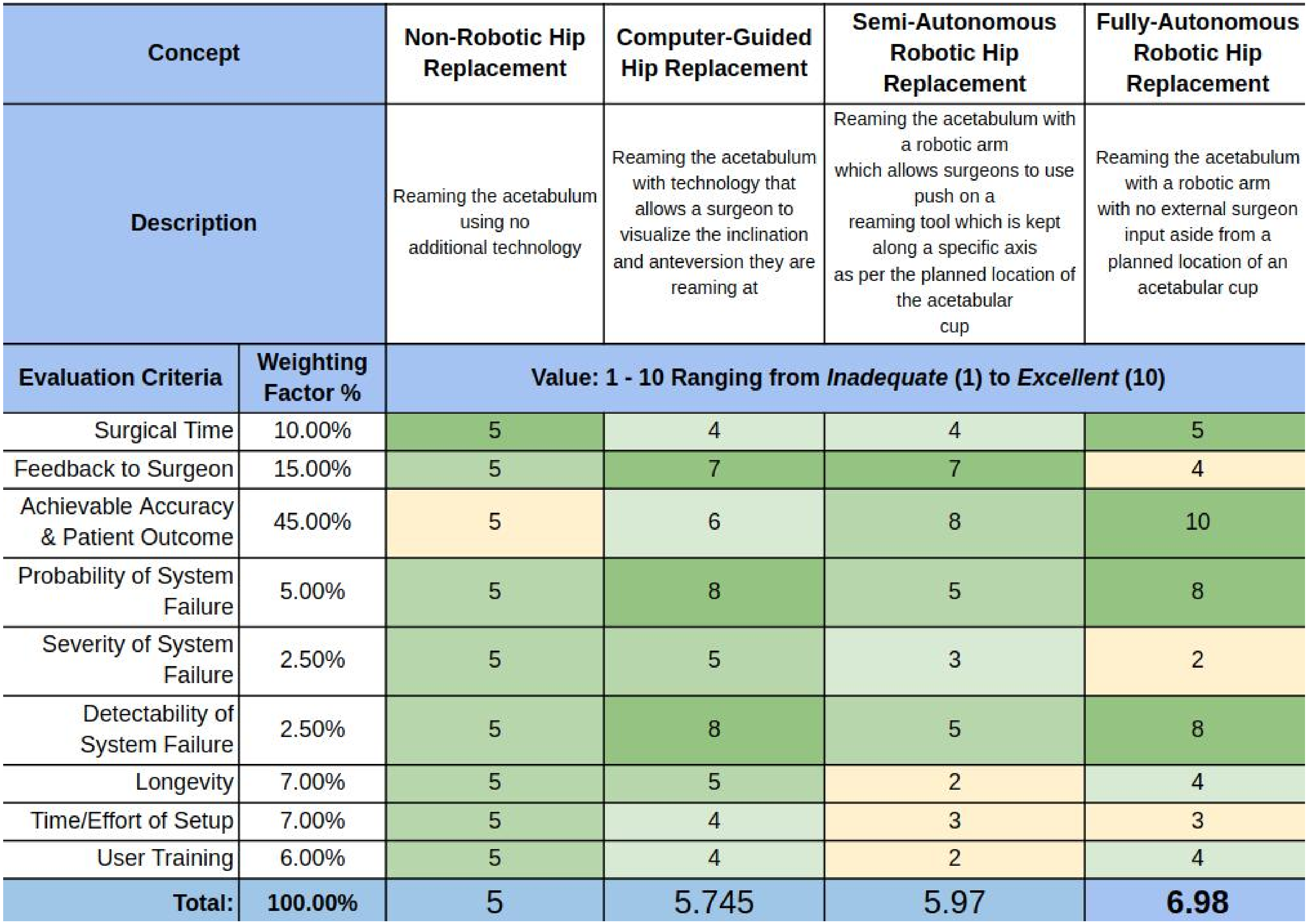


Figure 18: Full Size Cyberphysical Architecture

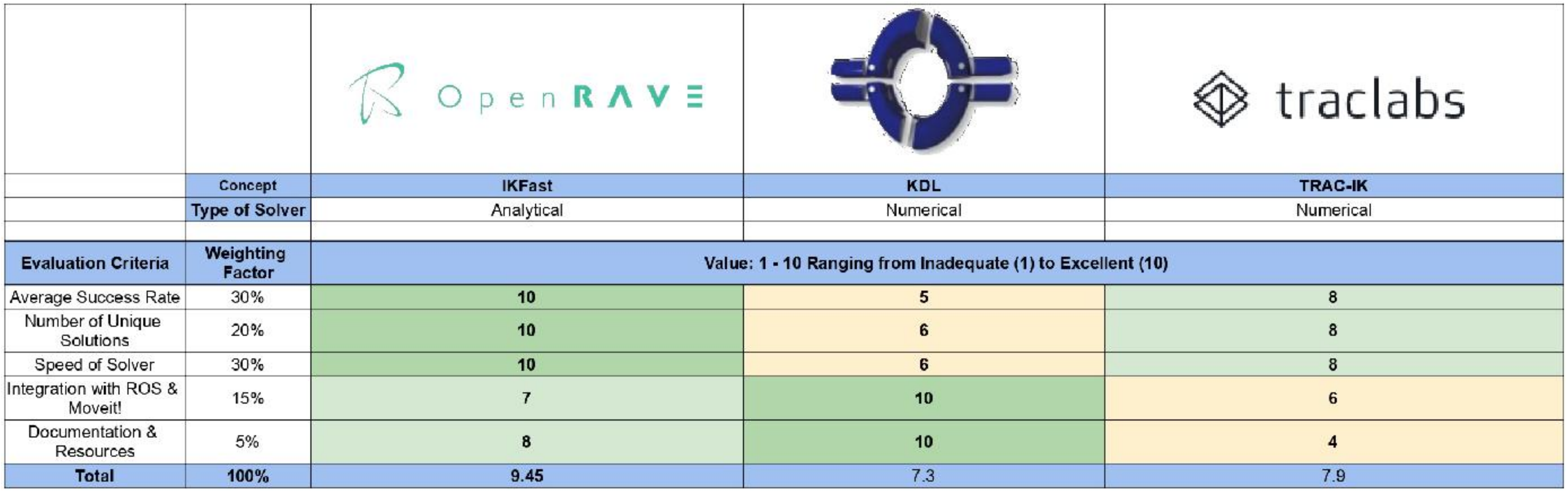
### Full Size System Level Trade Study



| Concept | | Non-Robotic Hip Replacement | Computer-Guided Hip Replacement | Semi-Autonomous Robotic Hip Replacement | Fully-Autonomous Robotic Hip Replacement |
|---|---|---|---|---|---|
| Description | | Reaming the acetabulum using no additional technology | Reaming the acetabulum with technology that allows a surgeon to visualize the inclination and anteversion they are reaming at | Reaming the acetabulum with a robotic arm which allows surgeons to use push on a reaming tool which is kept along a specific axis as per the planned location of the acetabular cup | Reaming the acetabulum with a robotic arm with no external surgeon input aside from a planned location of an acetabular cup |
| **Evaluation Criteria** | **Weighting Factor %** | **Value: 1 - 10 Ranging from *Inadequate* (1) to *Excellent* (10)** | | | |
| Surgical Time | 10.00% | 5 | 4 | 4 | 5 |
| Feedback to Surgeon | 15.00% | 5 | 7 | 7 | 4 |
| Achievable Accuracy & Patient Outcome | 45.00% | 5 | 6 | 8 | 10 |
| Probability of System Failure | 5.00% | 5 | 8 | 5 | 8 |
| Severity of System Failure | 2.50% | 5 | 5 | 3 | 2 |
| Detectability of System Failure | 2.50% | 5 | 8 | 5 | 8 |
| Longevity | 7.00% | 5 | 5 | 2 | 4 |
| Time/Effort of Setup | 7.00% | 5 | 4 | 3 | 3 |
| User Training | 6.00% | 5 | 4 | 2 | 4 |
| **Total:** | **100.00%** | 5 | 5.745 | 5.97 | **6.98** |

Figure 19: Full Size System Level Trade Study

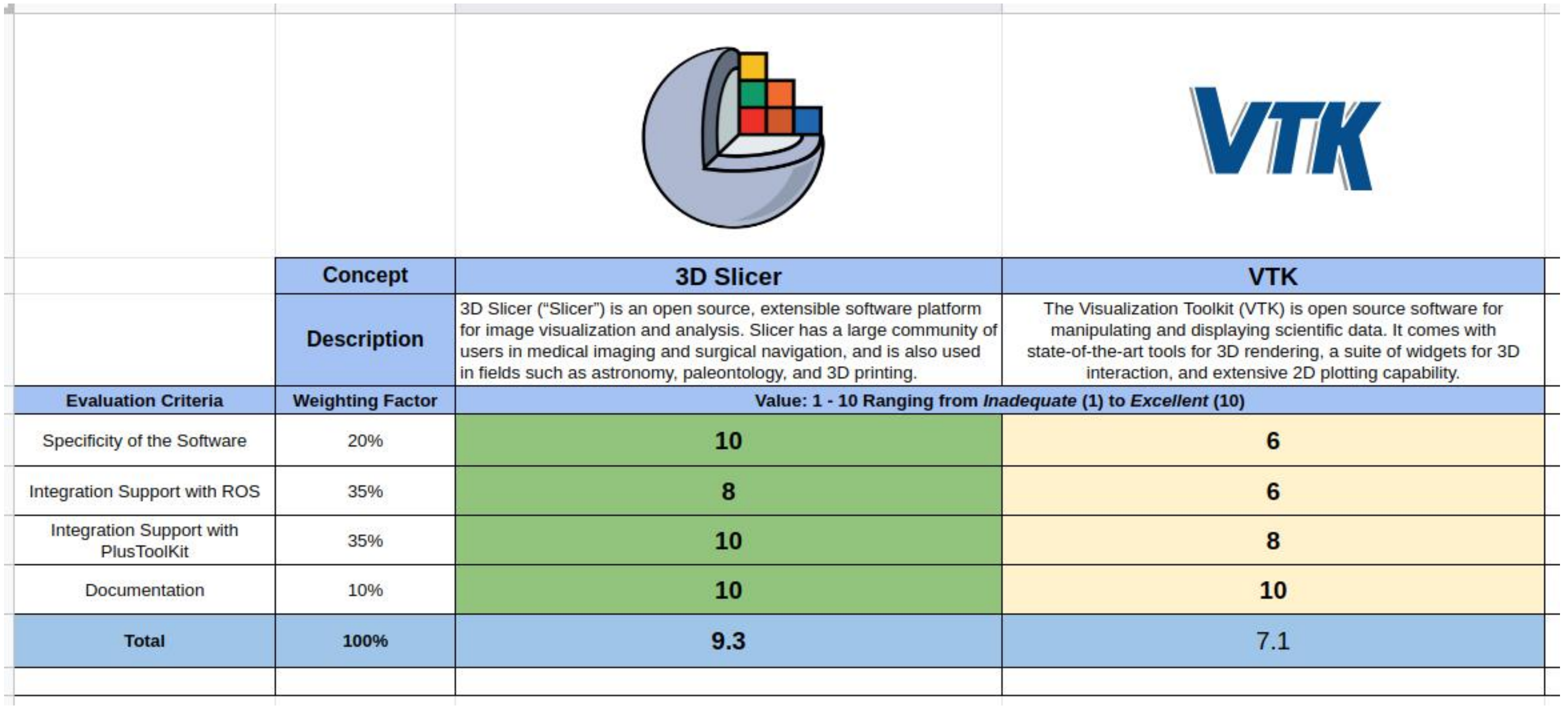
### Full Size Sub-System Level Trade Study - Inverse Kinematics Packages



|  |  | OpenRAVE |  | traclabs |
|---|---|---|---|---|
|  | Concept | IKFast | KDL | TRAC-IK |
|  | Type of Solver | Analytical | Numerical | Numerical |
| Evaluation Criteria | Weighting Factor | Value: 1 - 10 Ranging from Inadequate (1) to Excellent (10) |  |  |
| Average Success Rate | 30% | 10 | 5 | 8 |
| Number of Unique Solutions | 20% | 10 | 6 | 8 |
| Speed of Solver | 30% | 10 | 6 | 8 |
| Integration with ROS & Moveit! | 15% | 7 | 10 | 6 |
| Documentation & Resources | 5% | 8 | 10 | 4 |
| Total | 100% | 9.45 | 7.3 | 7.9 |

Figure 20: Full Size Sub-System Level Trade Study - 1

### Full Size Sub-System Level Trade Study - Visualization ToolKits



| | Concept | 3D Slicer | VTK |
|---|---|---|---|
| | Description | 3D Slicer ("Slicer") is an open source, extensible software platform for image visualization and analysis. Slicer has a large community of users in medical imaging and surgical navigation, and is also used in fields such as astronomy, paleontology, and 3D printing. | The Visualization Toolkit (VTK) is open source software for manipulating and displaying scientific data. It comes with state-of-the-art tools for 3D rendering, a suite of widgets for 3D interaction, and extensive 2D plotting capability. |
| **Evaluation Criteria** | **Weighting Factor** | **Value: 1 - 10 Ranging from *Inadequate* (1) to *Excellent* (10)** | |
| Specificity of the Software | 20% | **10** | **6** |
| Integration Support with ROS | 35% | **8** | **6** |
| Integration Support with PlusToolKit | 35% | **10** | **8** |
| Documentation | 10% | **10** | **10** |
| **Total** | **100%** | **9.3** | 7.1 |

Figure 21: Full Size Sub-System Level Trade Study - 2

## Full Size Work Breakdown Structure

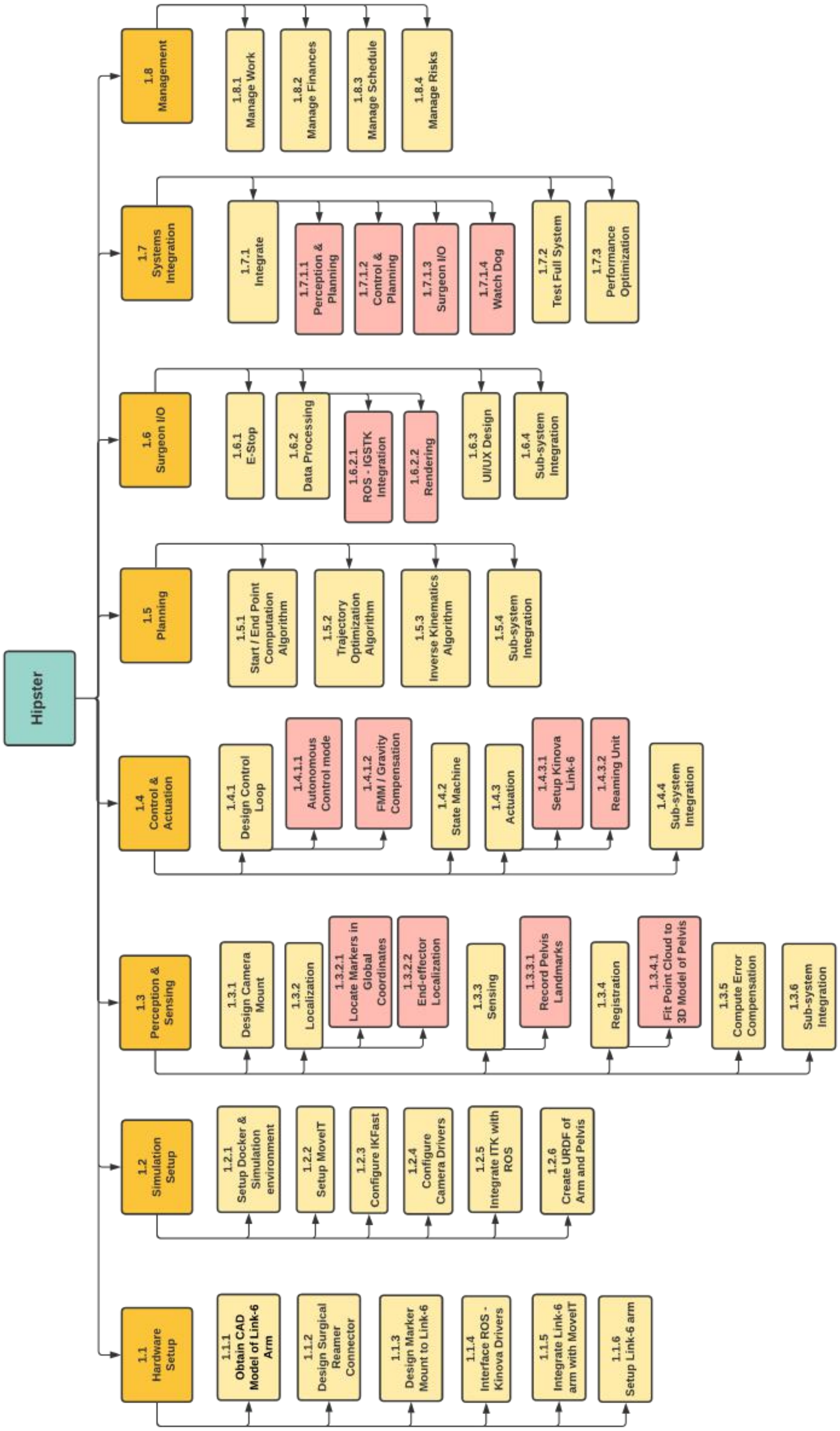

Hipster
1.1 Hardware Setup
1.1.1 Obtain CAD Model of Link-6 Arm
1.1.2 Design Surgical Reamer Connector
1.1.3 Design Marker Mount to Link-6
1.1.4 Interface ROS - Kinova Drivers
1.1.5 Integrate Link-6 arm with MoveIT
1.1.6 Setup Link-6 arm
1.2 Simulation Setup
1.2.1 Setup Docker & Simulation environment
1.2.2 Setup MoveIT
1.2.3 Configure IKFast
1.2.4 Configure Camera Drivers
1.2.5 Integrate ITK with ROS
1.2.6 Create URDF of Arm and Pelvis
1.3 Perception & Sensing
1.3.1 Design Camera Mount
1.3.2 Localization
1.3.2.1 Locate Markers in Global Coordinates
1.3.2.2 End-effector Localization
1.3.3 Sensing
1.3.3.1 Record Pelvis Landmarks
1.3.4 Registration
1.3.4.1 Fit Point Cloud to 3D Model of Pelvis
1.3.5 Compute Error Compensation
1.3.6 Sub-system Integration
1.4 Control & Actuation
1.4.1 Design Control Loop
1.4.1.1 Autonomous Control mode
1.4.1.2 FMM / Gravity Compensation
1.4.2 State Machine
1.4.3 Actuation
1.4.3.1 Setup Kinova Link-6
1.4.3.2 Reaming Unit
1.4.4 Sub-system Integration
1.5 Planning
1.5.1 Start / End Point Computation Algorithm
1.5.2 Trajectory Optimization Algorithm
1.5.3 Inverse Kinematics Algorithm
1.5.4 Sub-system Integration
1.6 Surgeon I/O
1.6.1 E-Stop
1.6.2 Data Processing
1.6.2.1 ROS - IGSTK Integration
1.6.2.2 Rendering
1.6.3 UI/UX Design
1.6.4 Sub-system Integration
1.7 Systems Integration
1.7.1 Integrate
1.7.1.1 Perception & Planning
1.7.1.2 Control & Planning
1.7.1.3 Surgeon I/O
1.7.1.4 Watch Dog
1.7.2 Test Full System
1.7.3 Performance Optimization
1.8 Management
1.8.1 Manage Work
1.8.2 Manage Finances
1.8.3 Manage Schedule
1.8.4 Manage Risks

Figure 22: Full Size Work Breakdown Structure

## Full Size Project Spring Schedule

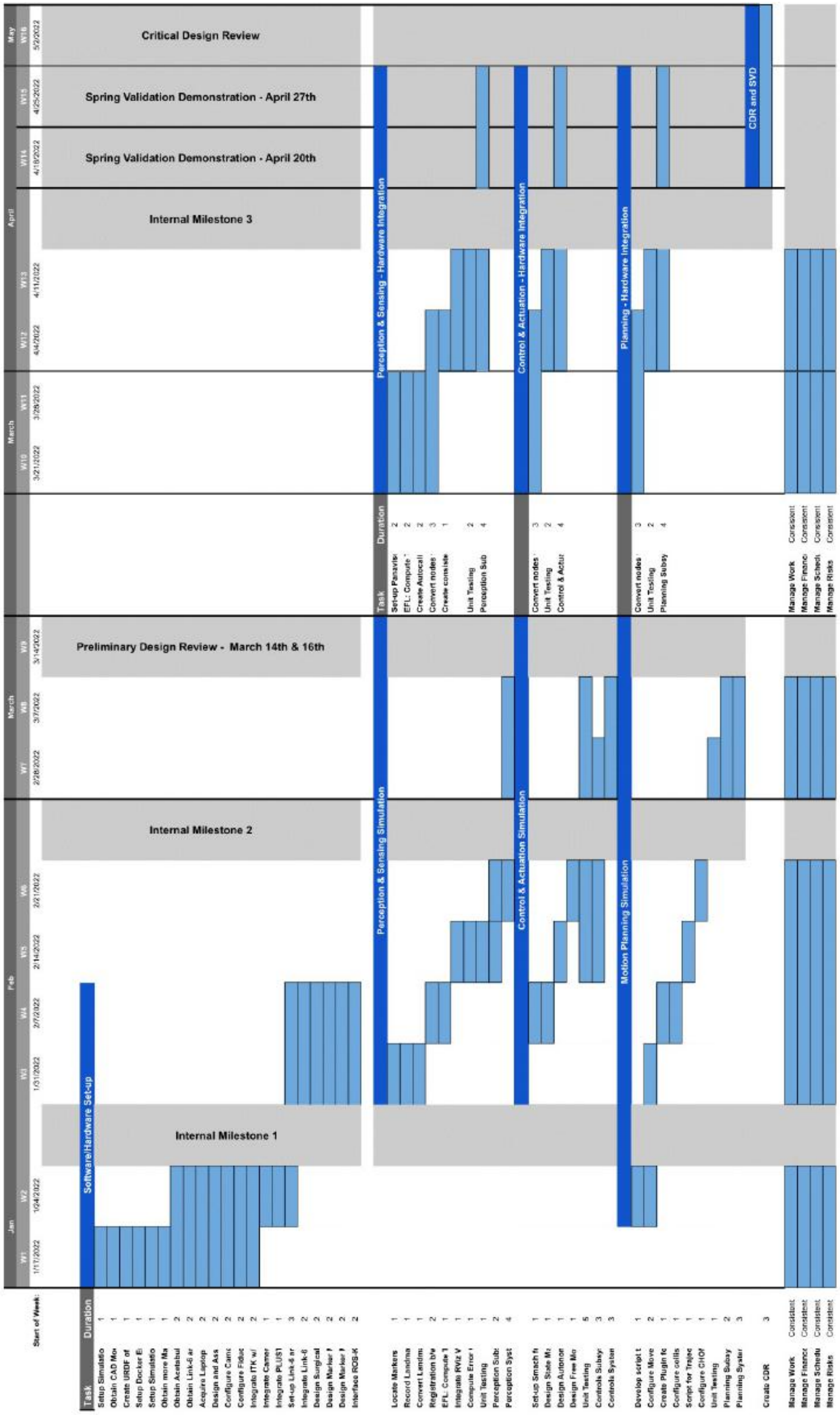


Figure 23: Full Size Project Schedule

## Full Size Project Fall Schedule

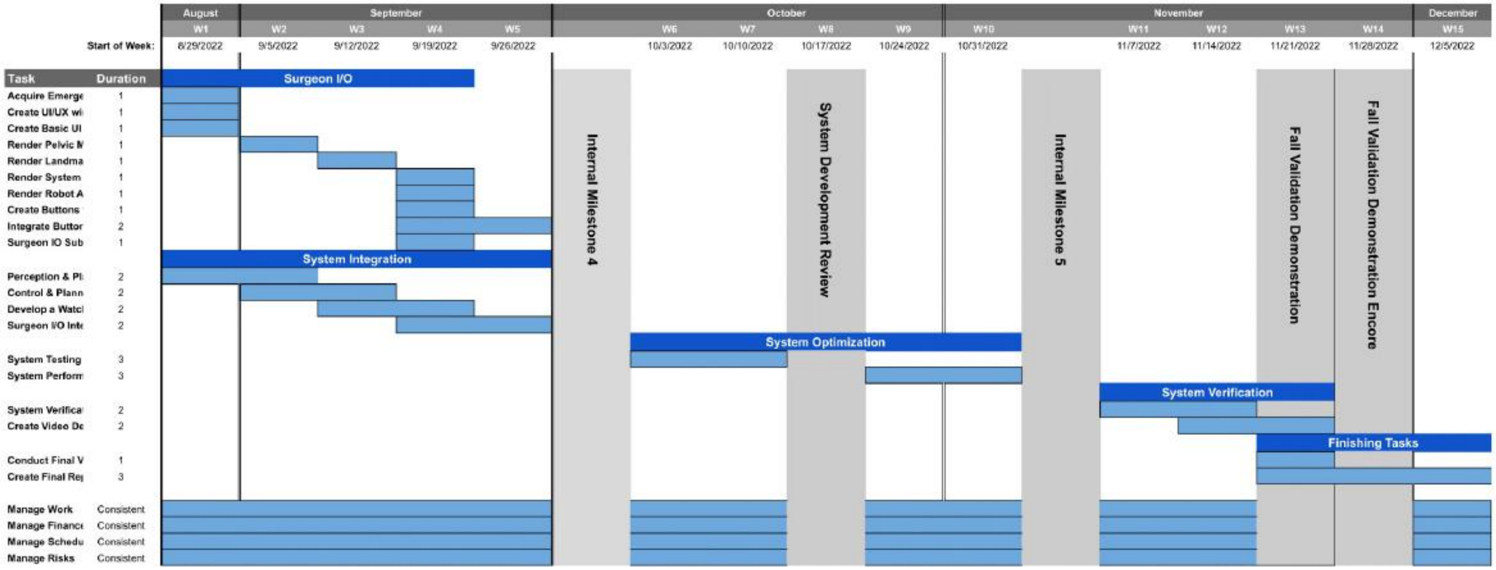
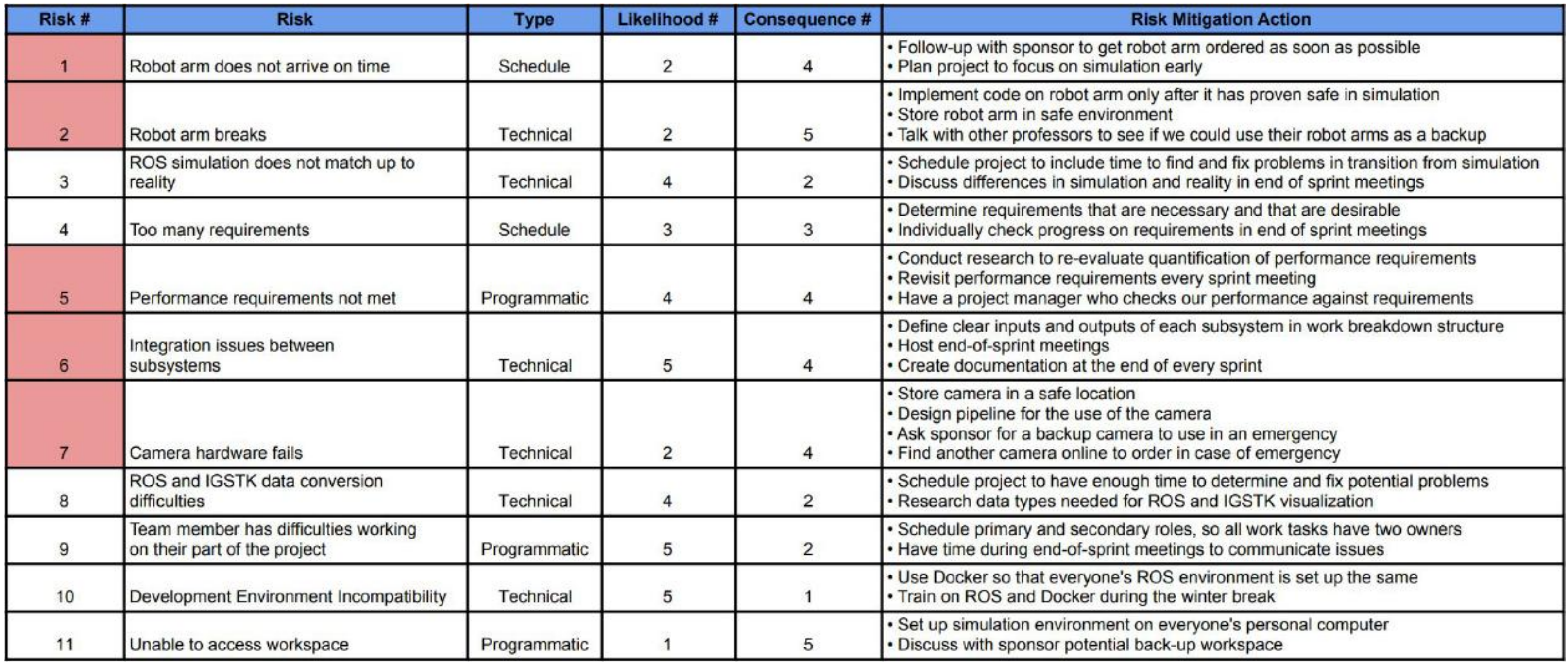


Figure 24: Full Size Project Schedule

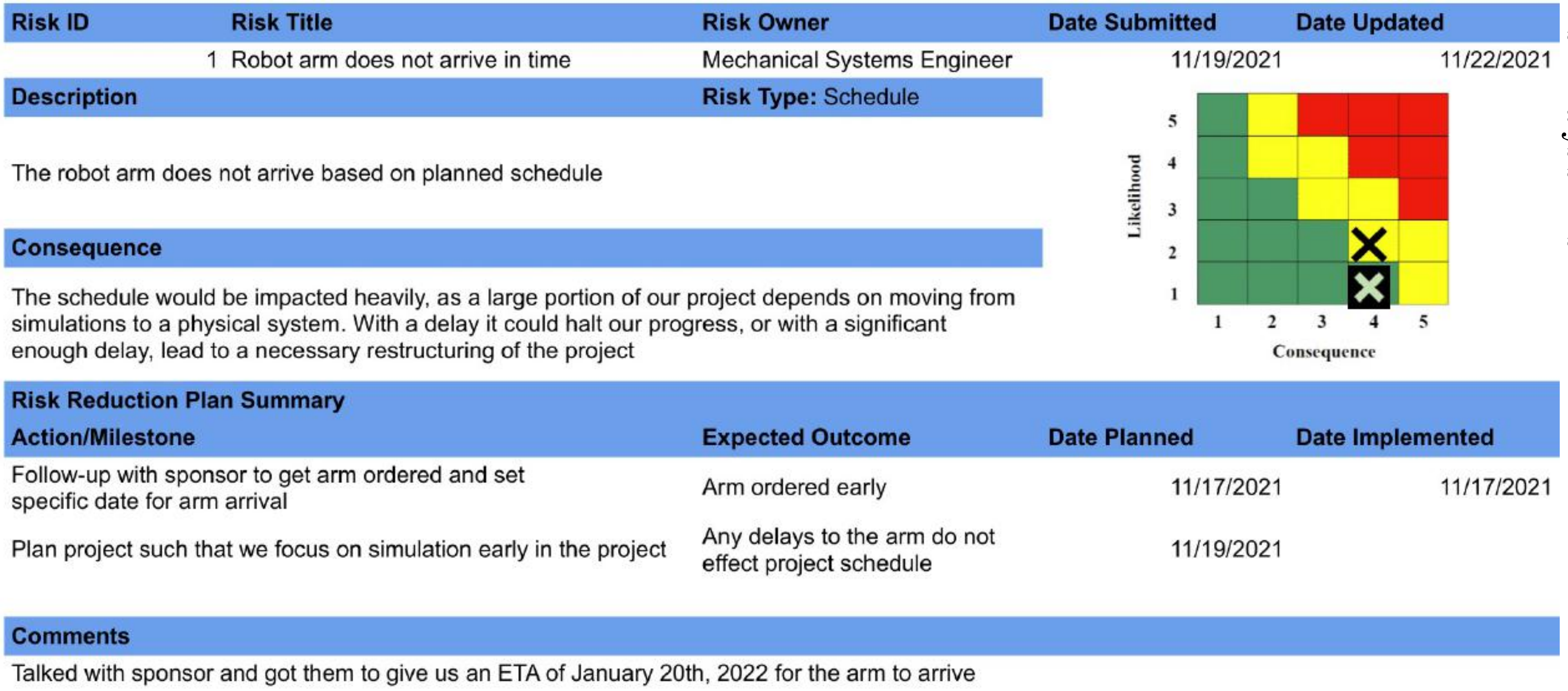
## Full Size Project Risks Summary



| Risk # | Risk | Type | Likelihood # | Consequence # | Risk Mitigation Action |
|---|---|---|---|---|---|
| 1 | Robot arm does not arrive on time | Schedule | 2 | 4 | • Follow-up with sponsor to get robot arm ordered as soon as possible<br>• Plan project to focus on simulation early |
| 2 | Robot arm breaks | Technical | 2 | 5 | • Implement code on robot arm only after it has proven safe in simulation<br>• Store robot arm in safe environment<br>• Talk with other professors to see if we could use their robot arms as a backup |
| 3 | ROS simulation does not match up to reality | Technical | 4 | 2 | • Schedule project to include time to find and fix problems in transition from simulation<br>• Discuss differences in simulation and reality in end of sprint meetings |
| 4 | Too many requirements | Schedule | 3 | 3 | • Determine requirements that are necessary and that are desirable<br>• Individually check progress on requirements in end of sprint meetings |
| 5 | Performance requirements not met | Programmatic | 4 | 4 | • Conduct research to re-evaluate quantification of performance requirements<br>• Revisit performance requirements every sprint meeting<br>• Have a project manager who checks our performance against requirements |
| 6 | Integration issues between subsystems | Technical | 5 | 4 | • Define clear inputs and outputs of each subsystem in work breakdown structure<br>• Host end-of-sprint meetings<br>• Create documentation at the end of every sprint |
| 7 | Camera hardware fails | Technical | 2 | 4 | • Store camera in a safe location<br>• Design pipeline for the use of the camera<br>• Ask sponsor for a backup camera to use in an emergency<br>• Find another camera online to order in case of emergency |
| 8 | ROS and IGSTK data conversion difficulties | Technical | 4 | 2 | • Schedule project to have enough time to determine and fix potential problems<br>• Research data types needed for ROS and IGSTK visualization |
| 9 | Team member has difficulties working on their part of the project | Programmatic | 5 | 2 | • Schedule primary and secondary roles, so all work tasks have two owners<br>• Have time during end-of-sprint meetings to communicate issues |
| 10 | Development Environment Incompatibility | Technical | 5 | 1 | • Use Docker so that everyone's ROS environment is set up the same<br>• Train on ROS and Docker during the winter break |
| 11 | Unable to access workspace | Programmatic | 1 | 5 | • Set up simulation environment on everyone's personal computer<br>• Discuss with sponsor potential back-up workspace |

Figure 25: Full Size Project Risks Summary

## Risk Analysis - Risk 1



| Risk ID | Risk Title | Risk Owner | Date Submitted | Date Updated |
|---|---|---|---|---|
| 1 | Robot arm does not arrive in time | Mechanical Systems Engineer | 11/19/2021 | 11/22/2021 |

**Description** **Risk Type:** Schedule

The robot arm does not arrive based on planned schedule

**Consequence**

The schedule would be impacted heavily, as a large portion of our project depends on moving from simulations to a physical system. With a delay it could halt our progress, or with a significant enough delay, lead to a necessary restructuring of the project

**Risk Reduction Plan Summary**

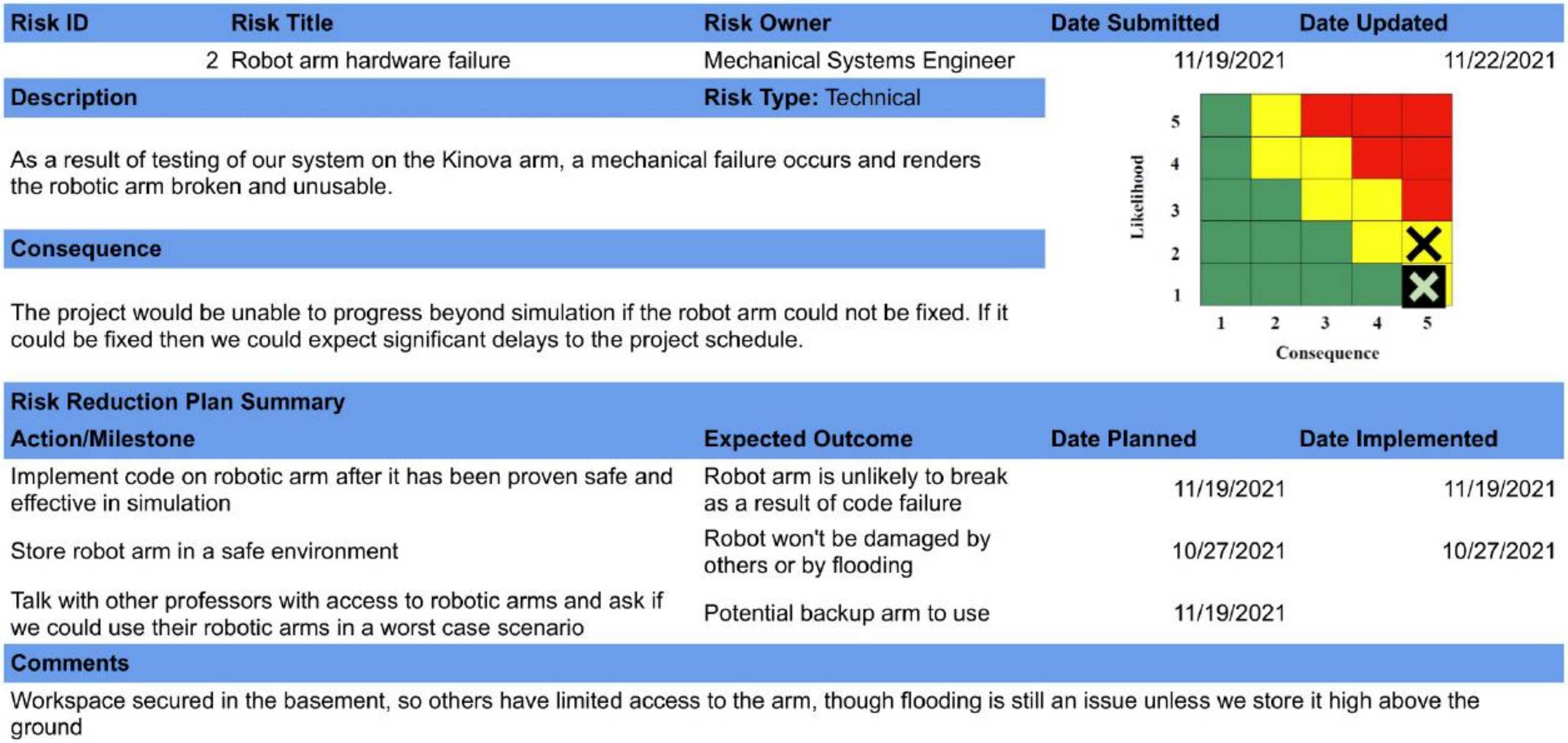
| Action/Milestone | Expected Outcome | Date Planned | Date Implemented |
|---|---|---|---|
| Follow-up with sponsor to get arm ordered and set specific date for arm arrival | Arm ordered early | 11/17/2021 | 11/17/2021 |
| Plan project such that we focus on simulation early in the project | Any delays to the arm do not effect project schedule | 11/19/2021 | |

**Comments**

Talked with sponsor and got them to give us an ETA of January 20th, 2022 for the arm to arrive

Figure 26: Risk Analysis for Risk 1

## Risk Analysis - Risk 2



| Risk ID | Risk Title | Risk Owner | Date Submitted | Date Updated |
|---|---|---|---|---|
| 2 | Robot arm hardware failure | Mechanical Systems Engineer | 11/19/2021 | 11/22/2021 |

**Description** | **Risk Type:** Technical

As a result of testing of our system on the Kinova arm, a mechanical failure occurs and renders the robotic arm broken and unusable.

**Consequence**

The project would be unable to progress beyond simulation if the robot arm could not be fixed. If it could be fixed then we could expect significant delays to the project schedule.

**Risk Reduction Plan Summary**

| Action/Milestone | Expected Outcome | Date Planned | Date Implemented |
|---|---|---|---|
| Implement code on robotic arm after it has been proven safe and effective in simulation | Robot arm is unlikely to break as a result of code failure | 11/19/2021 | 11/19/2021 |
| Store robot arm in a safe environment | Robot won't be damaged by others or by flooding | 10/27/2021 | 10/27/2021 |
| Talk with other professors with access to robotic arms and ask if we could use their robotic arms in a worst case scenario | Potential backup arm to use | 11/19/2021 | |

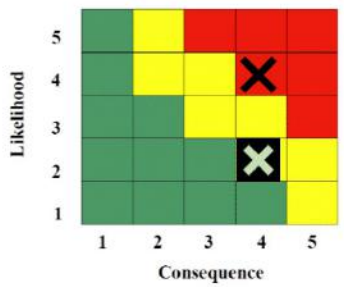
**Comments**

Workspace secured in the basement, so others have limited access to the arm, though flooding is still an issue unless we store it high above the ground

Figure 27: Risk Analysis for Risk 2

## Risk Analysis - Risk 5

| Risk ID | Risk Title | Risk Owner | Date Submitted | Date Updated |
|---|---|---|---|---|
| 5 | Performance requirements not met | Project Manager | 11/19/2021 | 11/22/2021 |
| **Description** | | **Risk Type:** Programmatic | | |
| Some or all of the performance requirements we set for the system have not been met | | | | |
| **Consequence** | | | | |
| If the performance is bad enough the project could be considered a failure and not novel enough to make an impact in the medical industry after the conclusion of the project | | | | |



| Risk Reduction Plan Summary | | | |
|---|---|---|---|
| **Action/Milestone** | **Expected Outcome** | **Date Planned** | **Date Implemented** |
| Conduct research on performance requirements to determine whether frequency or latency is better, and what the system would need to do as a minimum viable product | Accurate performance requirements that can feasibly be met | 11/20/2021 | 11/22/2021 |
| Consistently revisit and update performance requirements | Updated requirements as realities of project set in | 11/20/2021 | |
| Have a project manager role to keep on task | If project is completed in a timely manner then we have a lot of time to tune the system | 11/20/2021 | 11/20/2021 |

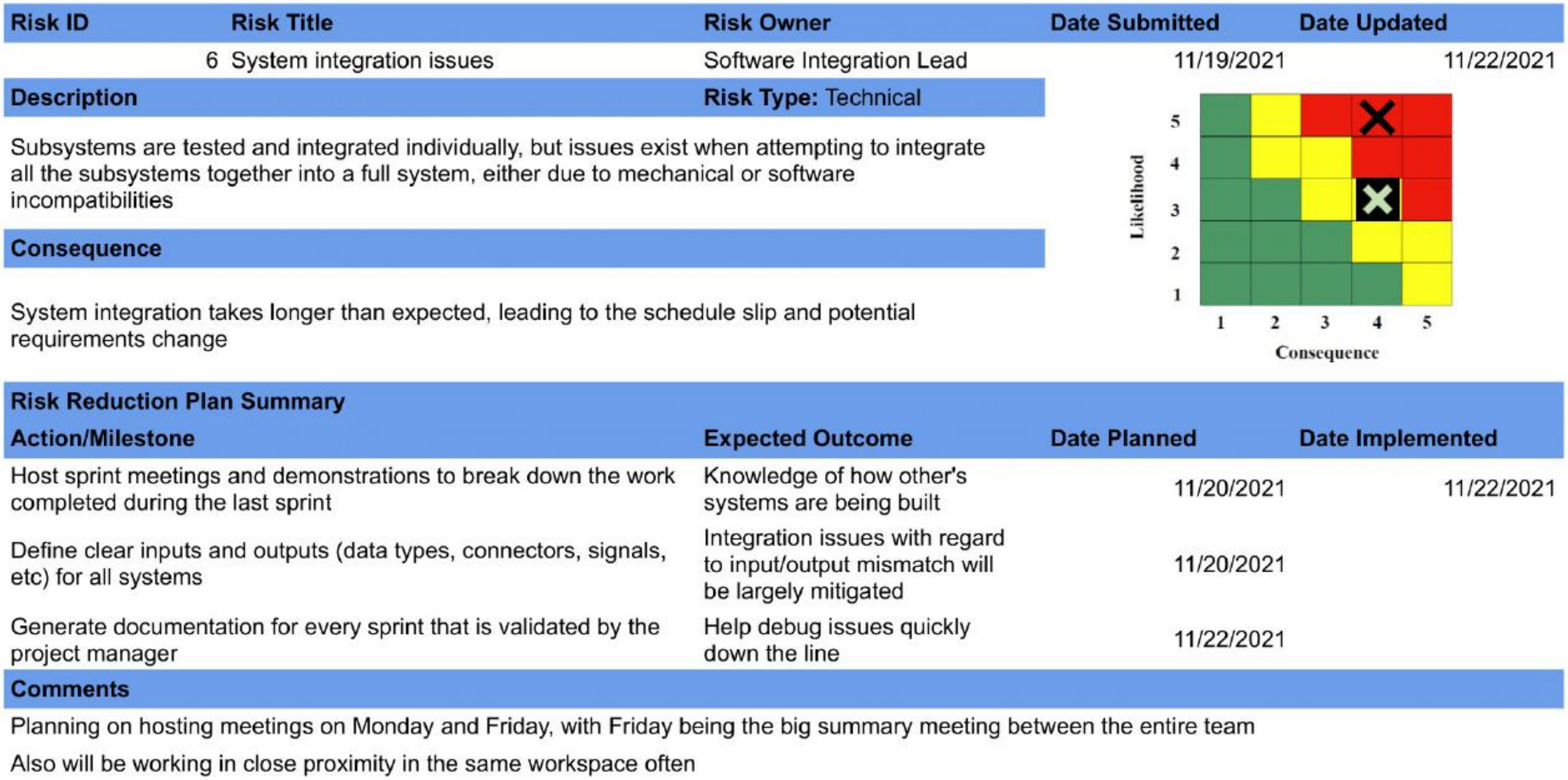
**Comments**

Discussion with sponsor led us to re-analyze our performance requirements to be latency based

Plan on discussing performance requirements and feasibility every sprint meeting

**Figure 28: Risk Analysis for Risk 5**

## Risk Analysis - Risk 6



| Risk ID | Risk Title | Risk Owner | Date Submitted | Date Updated |
|---|---|---|---|---|
| 6 | System integration issues | Software Integration Lead | 11/19/2021 | 11/22/2021 |

**Description** **Risk Type:** Technical

Subsystems are tested and integrated individually, but issues exist when attempting to integrate all the subsystems together into a full system, either due to mechanical or software incompatibilities

**Consequence**

System integration takes longer than expected, leading to the schedule slip and potential requirements change

**Risk Reduction Plan Summary**

| Action/Milestone | Expected Outcome | Date Planned | Date Implemented |
|---|---|---|---|
| Host sprint meetings and demonstrations to break down the work completed during the last sprint | Knowledge of how other's systems are being built | 11/20/2021 | 11/22/2021 |
| Define clear inputs and outputs (data types, connectors, signals, etc) for all systems | Integration issues with regard to input/output mismatch will be largely mitigated | 11/20/2021 | |
| Generate documentation for every sprint that is validated by the project manager | Help debug issues quickly down the line | 11/22/2021 | |

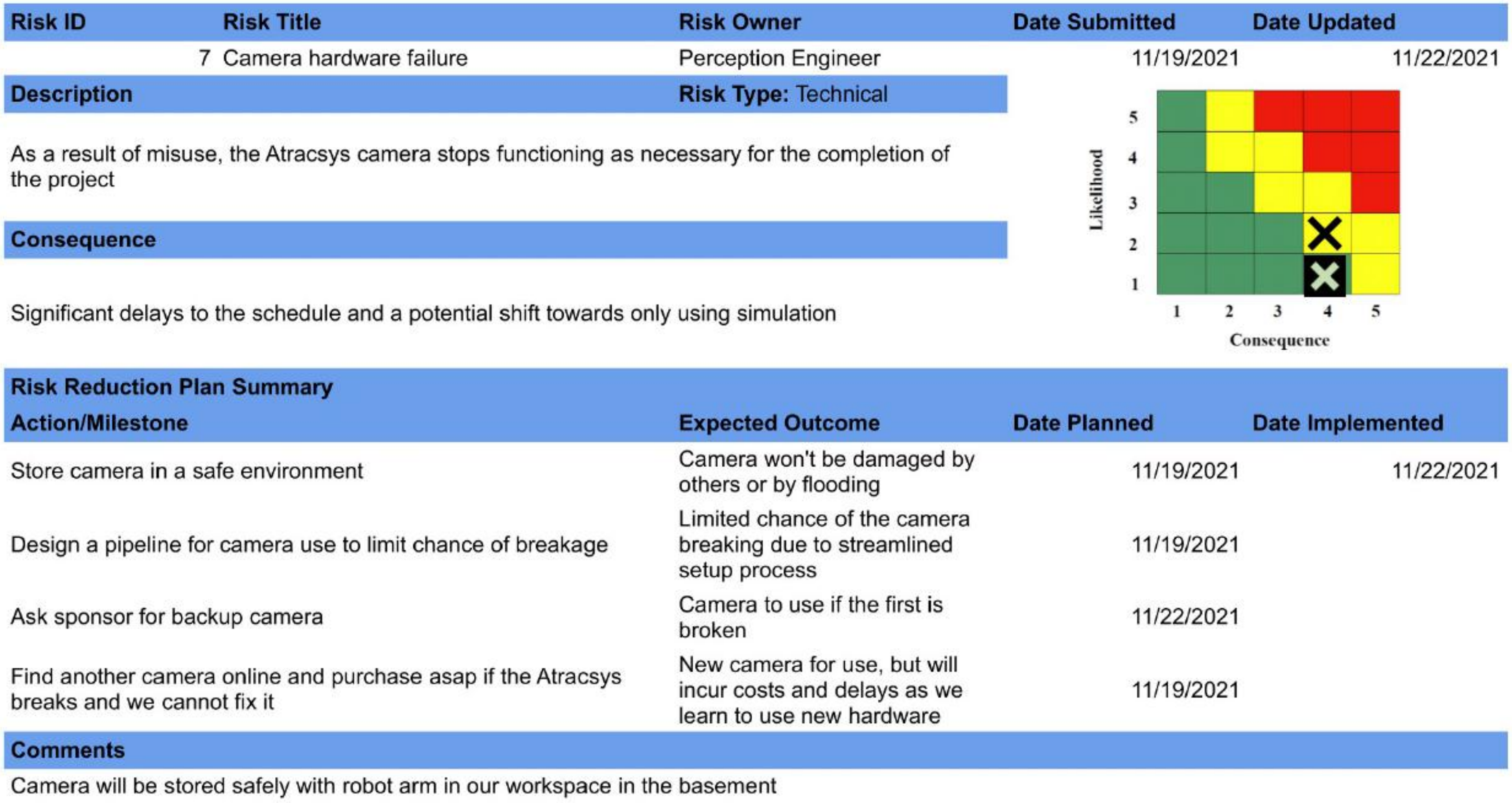
**Comments**

Planning on hosting meetings on Monday and Friday, with Friday being the big summary meeting between the entire team

Also will be working in close proximity in the same workspace often

Figure 29: Risk Analysis for Risk 6

## Risk Analysis - Risk 7



| Risk ID | Risk Title | Risk Owner | Date Submitted | Date Updated |
|---|---|---|---|---|
| 7 | Camera hardware failure | Perception Engineer | 11/19/2021 | 11/22/2021 |

**Description** — **Risk Type:** Technical

As a result of misuse, the Atracsys camera stops functioning as necessary for the completion of the project

**Consequence**

Significant delays to the schedule and a potential shift towards only using simulation

**Risk Reduction Plan Summary**

| Action/Milestone | Expected Outcome | Date Planned | Date Implemented |
|---|---|---|---|
| Store camera in a safe environment | Camera won't be damaged by others or by flooding | 11/19/2021 | 11/22/2021 |
| Design a pipeline for camera use to limit chance of breakage | Limited chance of the camera breaking due to streamlined setup process | 11/19/2021 | |
| Ask sponsor for backup camera | Camera to use if the first is broken | 11/22/2021 | |
| Find another camera online and purchase asap if the Atracsys breaks and we cannot fix it | New camera for use, but will incur costs and delays as we learn to use new hardware | 11/19/2021 | |

**Comments**

Camera will be stored safely with robot arm in our workspace in the basement

Figure 30: Risk Analysis for Risk 7