May 3rd, 2022

HIPSTER

Critical Design Review

Autonomous Reaming for Total Hip Replacement
(ARTHuR)

The Team

Kaushik Balasundar

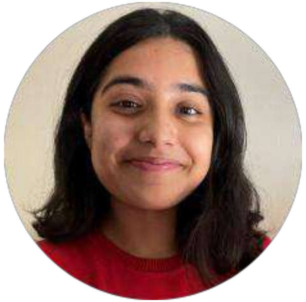
Perception and
Sensing Lead

Parker Hill

Mechanical
Systems
Engineering Lead

Anthony Kyu

Controls and
Actuation Lead



Gunjan Sethi

Software
Engineering Lead

Sundaram Seivur

Trajectory Planning
Lead

Contents

Motivation

A doctor may recommend hip replacement if there exists **significant pain, inflammation and damage to the hip joint.**

When is the surgery successful?
Acetabular implant is within **Lewinnek Safe Zone**.

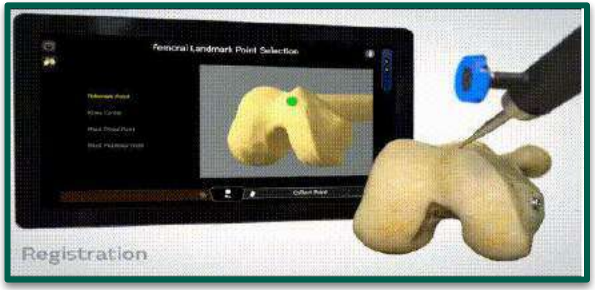
What is the current success rate?
Less than 50% of manual surgeries are within the Lewinnek safe zone

Why?
Surgeons **cannot see site of surgery** very well. Lots of **forces** involved.

Project Description

A **fully autonomous** robotic arm aimed at performing acetabular reaming with **high accuracy**, eliminating the need of surgeons to use intuition to correctly position/angle the reamer.

Use Case



System Level Requirements

Mandatory Functional Requirements

The system will

M.P.1.1 **Localize** the robot arm with a **latency less than or equal to 50 ms**

M.P.1.2.1 **Localize** the robot arm with respect to the pelvis with a **position error of less than 1 mm**

M.P.1.2.2 **Localize** the robot arm with respect to the pelvis with an **orientation error less than 1.5 degrees**

M.P.2 **Plan** the trajectory based on the given surgical plan with a **latency less than or equal to 150 ms**

M.P.3.1 **Execute** surgical plan by reaming along the trajectory with an **position error of less than ~~1 mm~~ 3 mm**

M.P.3.2 **Execute** surgical plan by reaming along the trajectory with an **orientation error of less than ~~1.5 degrees~~ 3-degrees**

M.P.4.1 **Compute** error and interpret the movement of the pelvis with a **latency less than or equal to 50 ms**

M.P.4.2 **Generate** a new trajectory **if the errors are greater than 1 mm or greater than 1.5 degrees**

M.P.5 **Adapt and compensate** for movement by generating a new trajectory with **latency less than or equal to 150 ms**

M.P.6 **Allow** the surgeon **to place the robot arm** to an initial position by **back-driving the robotic arm**

M.P.7 **Provide** the surgeon with **visual feedback** with a **latency less than or equal to 150 ms**

M.P.8 **Allow** the surgeon to **e-stop** the system, stopping the system **within 500 ms**

Justification: Kinova Gen3 Arm Accuracy Limitation

System Level Requirements

Mandatory Non-Functional Requirements

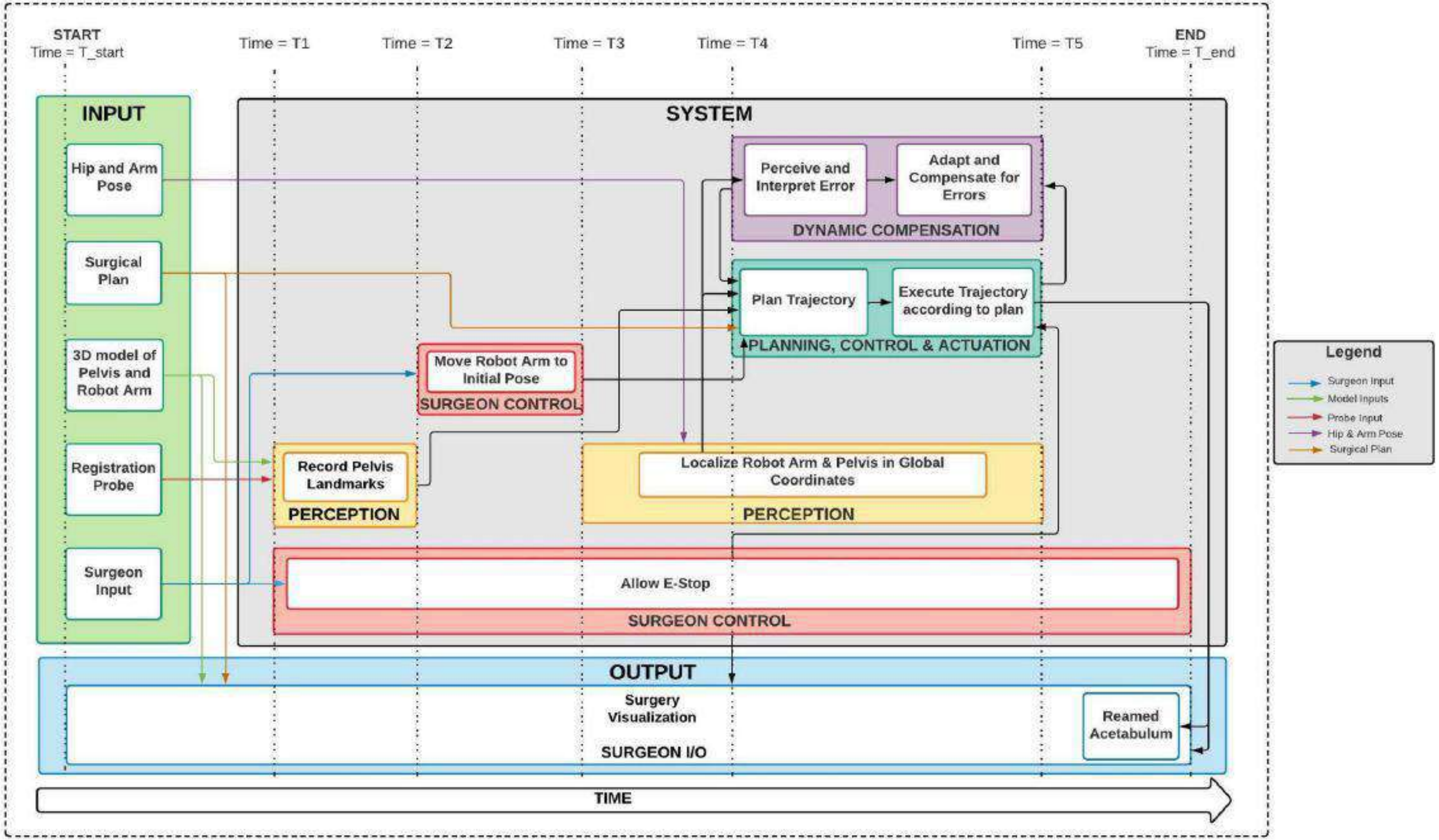
The system will

M.N.1 **Produce forces low enough** for it to be safe around humans

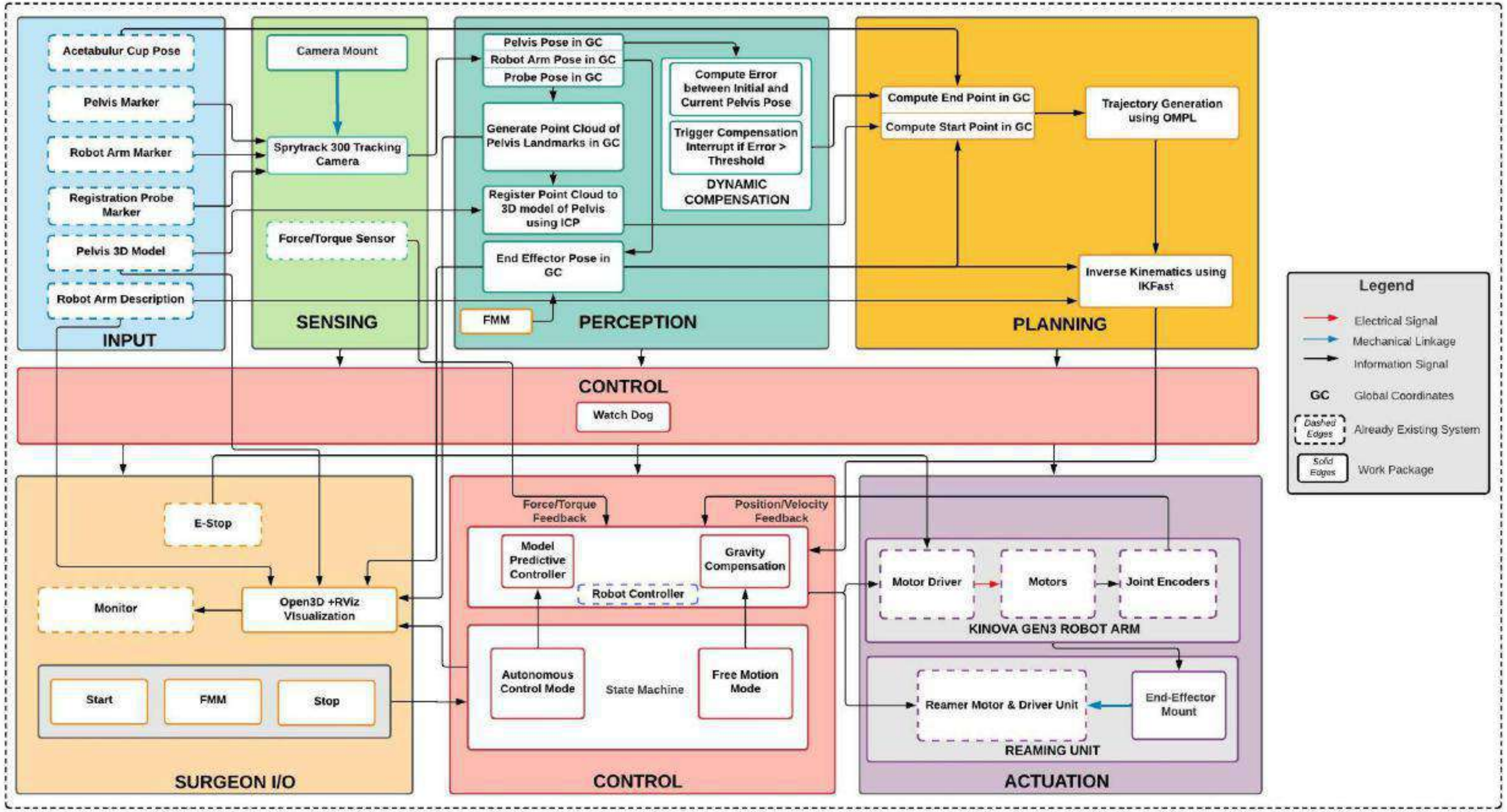
M.N.2 **Provide** a minimal and easy-to-interpret **user interface** design for surgeons

M.N.3 **Autonomously detect malfunctions** and errors and notify user accordingly

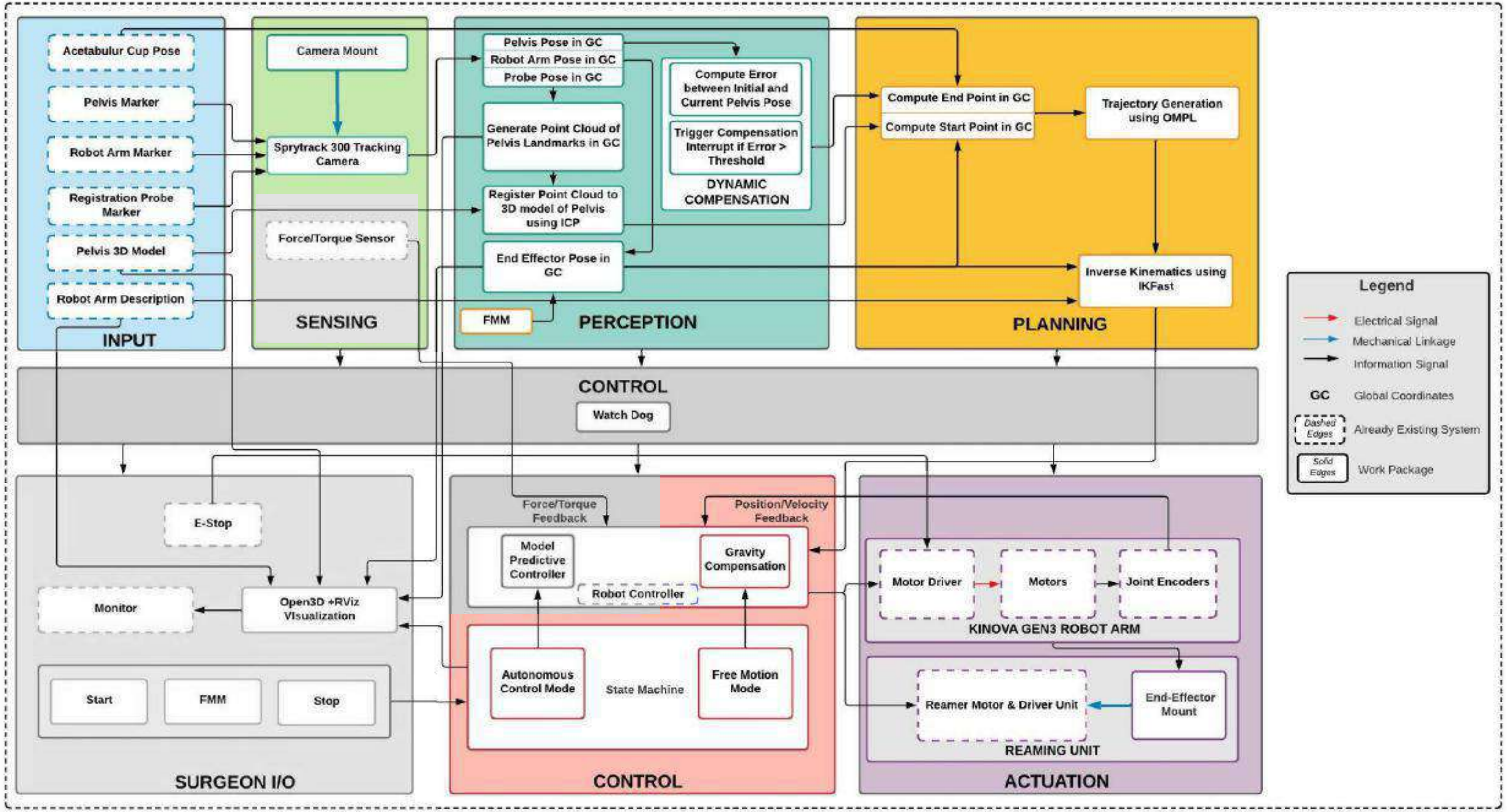
Functional Architecture



Cyberphysical Architecture



Current Progress



Current System Status

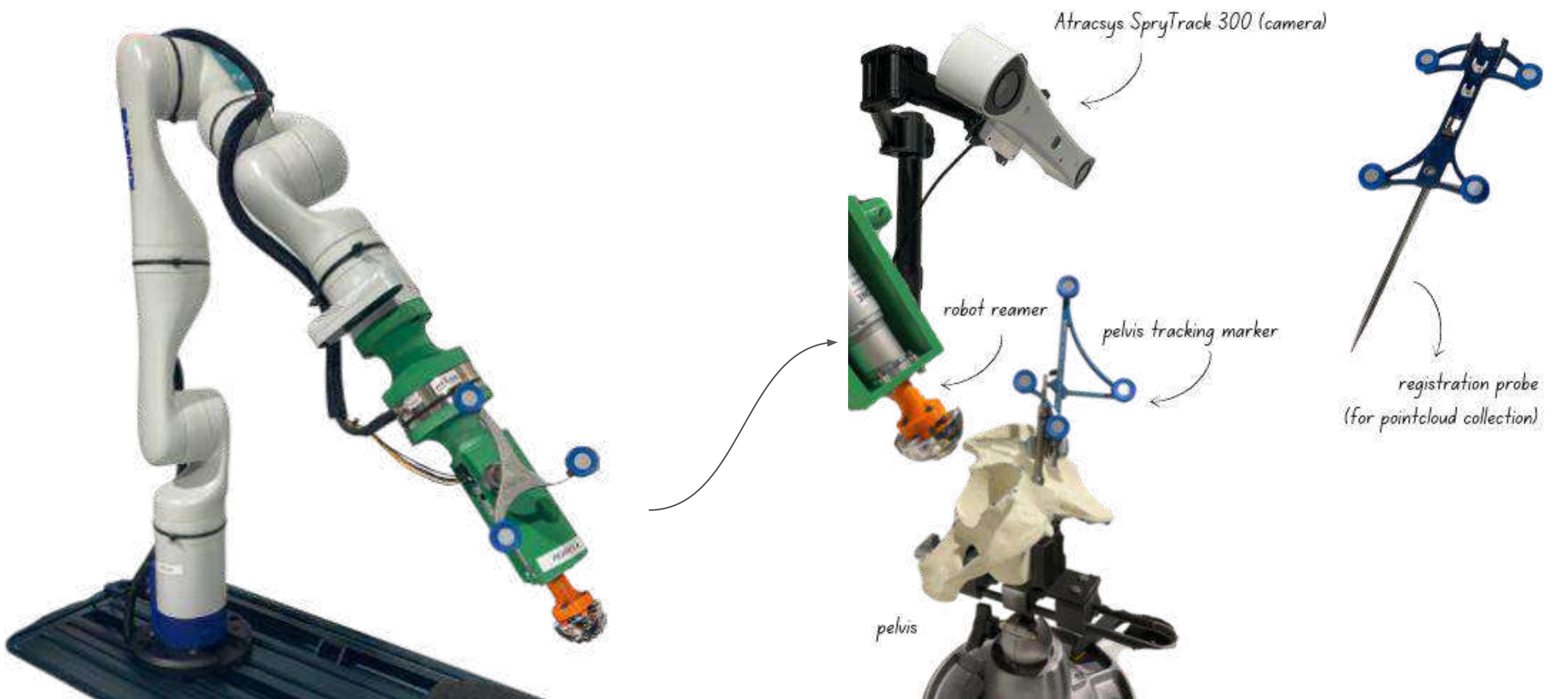
- Targeted Requirements
- Overall System Depiction
- Subsystem Descriptions and Analysis
- SVD Performance Evaluation
- Video Excerpt
- Strong/Weak Points

Targeted Requirements

#	Status	Requirement	Subsystem
M.P.1.1, M.P.1.2.1, M.P.1.2.2	✓	The system shall localize the robot arm in real-time with respect to the pelvis before and during surgery with a ***latency ≤ 500ms***.	Perception and Sensing
M.P.4.1 - 3	✓	The system shall compute error and interpret the movement of the pelvis during reaming with a ***latency ≤ 500 ms***, and detect changes with a ***position error of ≤ 3mm*** and ***orientation error ≤ 3 degrees***.	

#	Status	Requirement	Subsystem
M.P.2.	✓	The system shall plan the trajectory of the robot arm based on the given surgical plan with a ***latency ≤ 500ms***.	Motion Planning and Controls
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 ≤ 3mm*** and ***orientation error ≤ 3 degrees.***	
M.P.5.	✓	The system shall adapt and compensate for movement by generating a new trajectory with a ***latency ≤ 500ms***.	Motion Planning and Controls + Perception

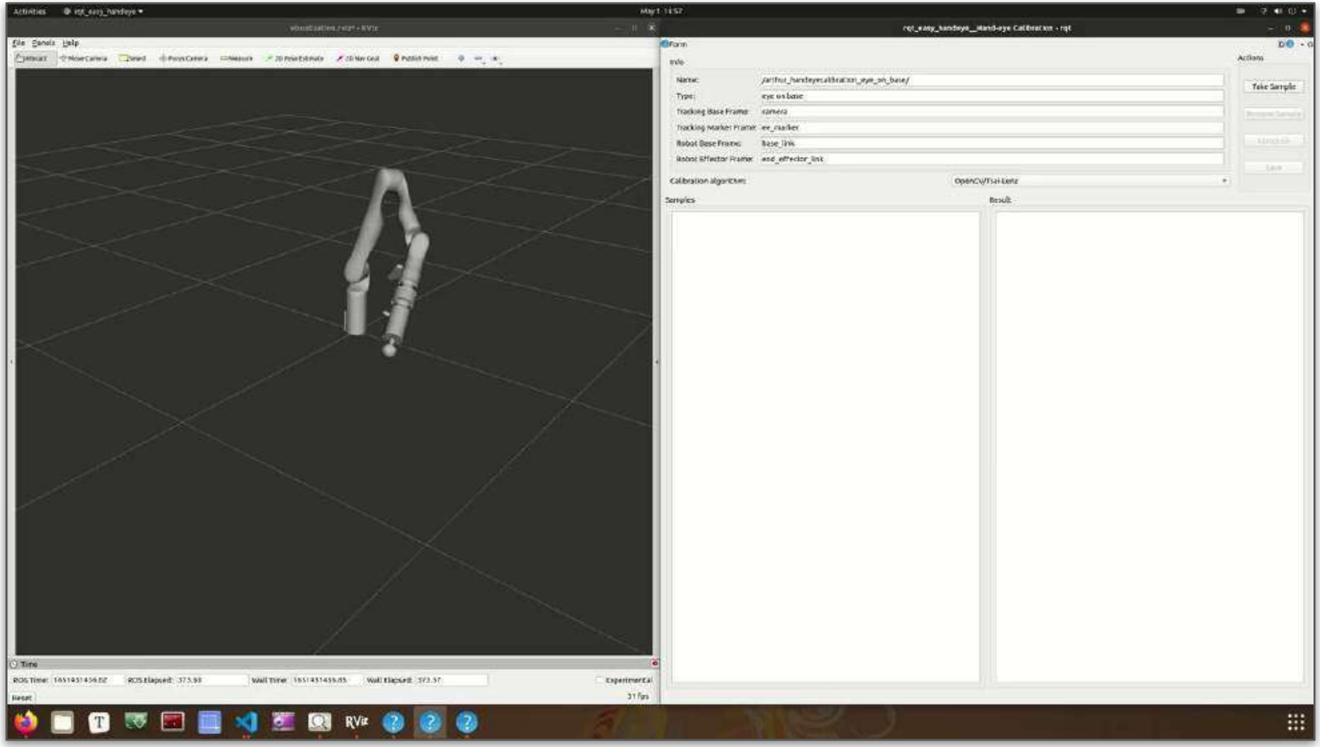
Overall System Depiction



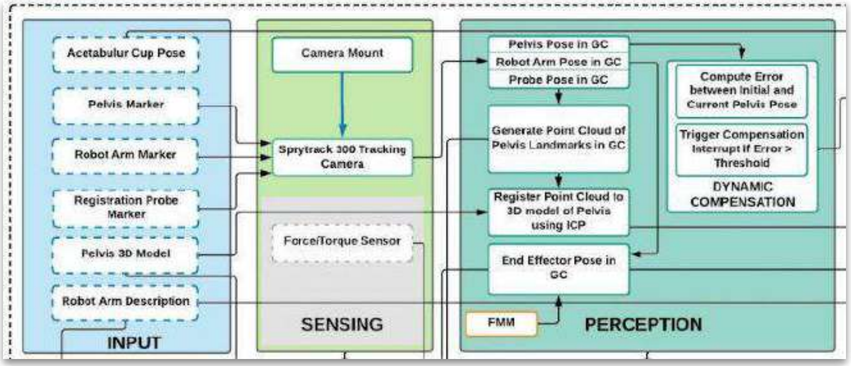
Subsystem Depictions and Progress

- Hand-Eye Calibration
- Perception and Sensing
- Motion Planning and Controls
- Hardware

Hand-Eye Calibration

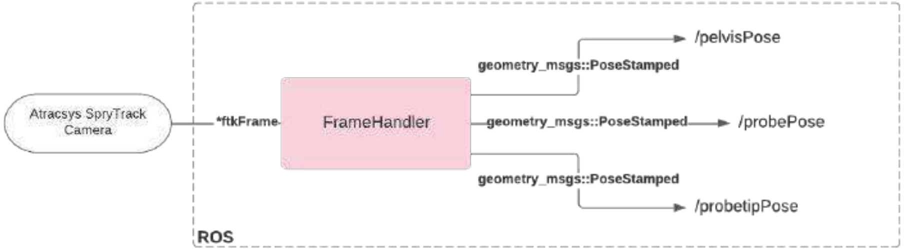


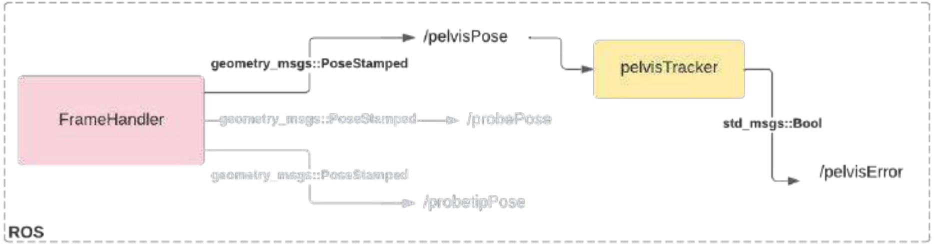
Perception and Sensing



#	Status	Requirement	Achieved Performance	Features
M.P.1.1, M.P.1.2.1, M.P.1.2.2	✓	The system shall localize the robot arm in real-time with respect to the pelvis before and during surgery with a ***latency ≤ 500ms***.	**~20 ms**	✓ Frame Handler
M.P.4.1 - 3	✓	The system shall compute error and interpret the movement of the pelvis during reaming with a ***latency ≤ 500 ms***, and detect changes with a ***position error of ≤ 3mm*** and ***orientation error ≤ 3 degrees***.	**<= 500ms** **<= 3mm position** **<= 3 degrees**	✓ Pointcloud Collector ✓ Registration ✓ Pelvis Tracker

Perception and Sensing





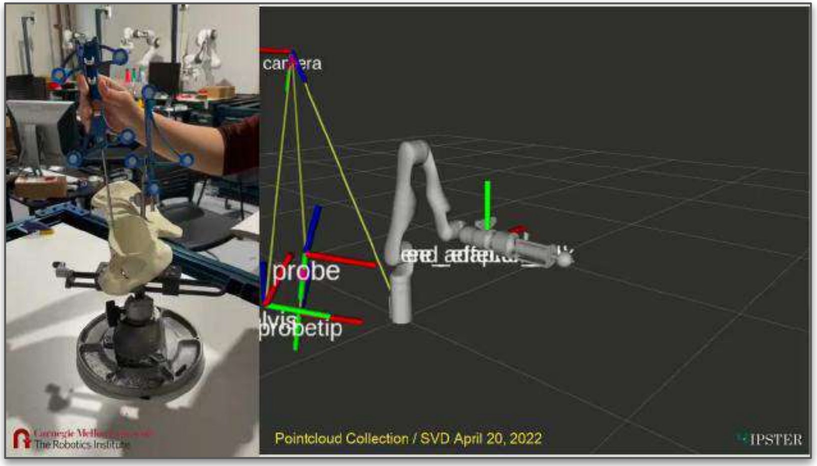
Frame Handler

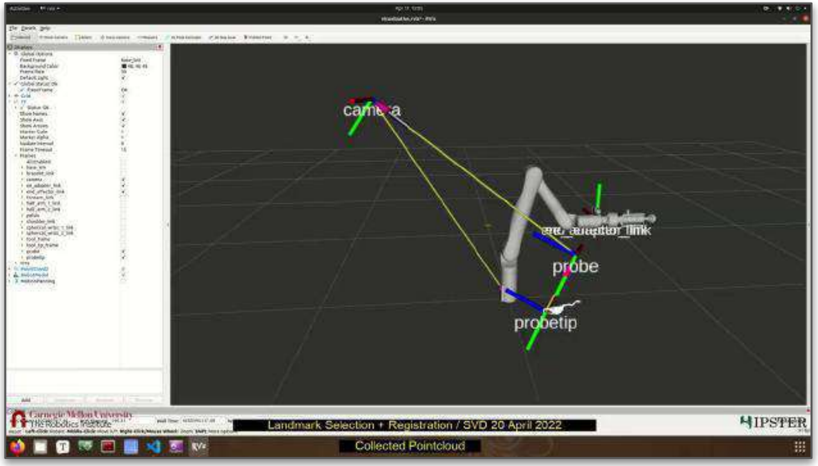
- ✓ Linked with Atracsys SDK C++ libraries
- ✓ Receives 6DOF positions in *ftkFrame type
- ✓ Typecastes all poses to standard ROS message types
- ✓ Publishes all poses to topics
- ✓ Broadcasts all poses as transforms

Pelvis Tracker

- ✓ Continuously subscribes to /pelvisPose
- ✓ Checks if changes in pelvis pose are beyond error thresholds
- ✓ Publishes a boolean error on topic if error thresholds are crossed

Perception and Sensing





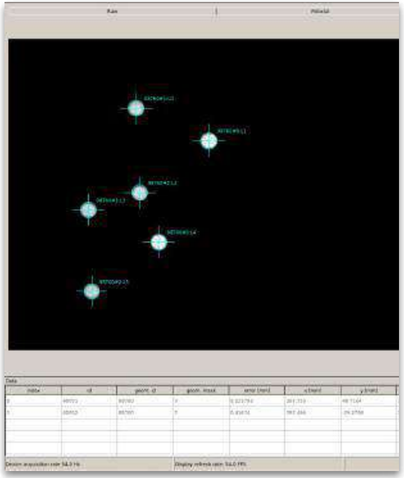
Pointcloud Collection

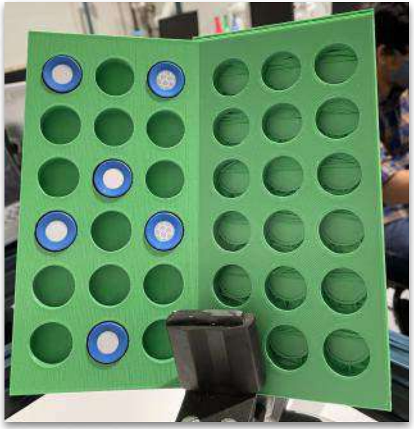
- ✓ Uses a registration probe to record a sparse or dense pointcloud
- ✓ Records points continuously as sensor_msgs/PointCloud2 message type.

Registration

- ✓ Performs initial manual landmark selection of 4-8 points
- ✓ Performs ICP-based registration between pre-scanned pelvis model and collected pointcloud
- ✓ Outputs the calculated transformation

Testing & Evaluation

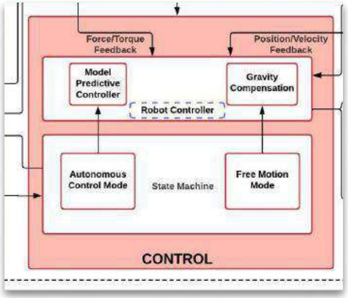


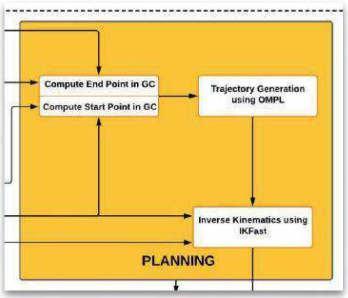


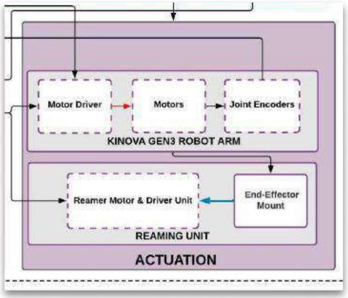
```
        min: 0.010s max: 0.026s std dev: 0.00097s window: 379
average rate: 54.033
        min: 0.010s max: 0.026s std dev: 0.00092s window: 433
average rate: 54.035
        min: 0.010s max: 0.026s std dev: 0.00088s window: 487
average rate: 54.032
        min: 0.010s max: 0.026s std dev: 0.00085s window: 541
average rate: 54.034
        min: 0.010s max: 0.027s std dev: 0.00095s window: 595
average rate: 54.036
        min: 0.010s max: 0.027s std dev: 0.00092s window: 649
```

Subsystem: Perception and Sensing		
Metric	**Required**	**Achieved**
Latency (ms)	< 500	~ 20
Position Error (mm)	<= 3	<= 2
Orientation Error (degress)	<= 3	<= 3

Motion Planning and Controls





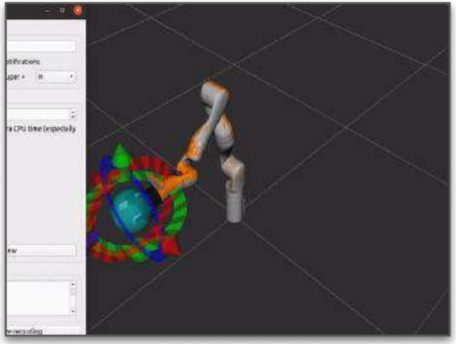


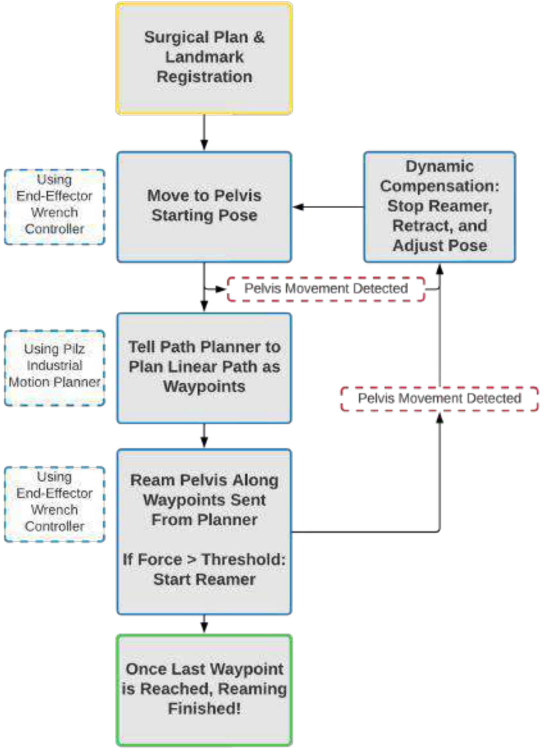
#	Status	Requirement	Achieved Performance	Features
M.P.2.	✓	The system shall plan the trajectory of the robot arm based on the given surgical plan with a ***latency*** ≤ ***500ms***.	Plans between **200ms - 700ms**. Highly dependent on configuration & hardware limitations.	✓ Arthur_Planning
M.P.3.1, M.P.3.2.	✓	The system shall execute a surgical plan by reaming along the generated trajectory with a ***position error of*** ≤ ***3mm*** and ***orientation error*** ≤ ***3 degrees.***	Max. position error of **2.4mm** and orientation error of **0.5 degrees**.	✓ Wrench_Controller ✓ Arthur_Planning
M.P.5.	✓	The system shall adapt and compensate for movement by generating a new trajectory with a ***latency*** ≤ ***500ms***.	Planning and controls within tolerance.	Integration (Perception + Planning + Controls)

Motion Planning and Controls

Arthur Planning

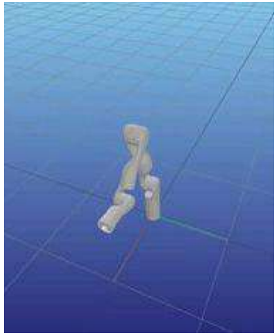
- ✓ Uses pilz industrial motion planner
- ✓ Takes reaming end point from perception system
- ✓ Plans linear path between current pose & reaming end point
- ✓ Takes input from controller node to plan new trajectory if dynamic compensation is triggered





Wrench Controller

- ✓ Uses Kinova Gen 3's Wrench Controller API
- ✓ Takes desired waypoint from Arthur Planning
- ✓ Calculates error in current vs desired pose
- ✓ Uses PID to calculate output wrench for wrench controller



MPC Preliminary Analysis

$$\min_{s_k, u_k, k\epsilon[1,H]} \quad \frac{1}{2}(s_H - s_d)^T Q_H (s_H - s_d) + \sum_{k=1}^{H-1} \frac{1}{2}(s_k - s_d)^T Q (s_k - s_d) + \frac{1}{2} u_k^T R u_k$$

$$\text{s.t.} \quad \begin{bmatrix} \dot{q} \\ \ddot{q} \\ \dot{x} \\ \ddot{x} \\ \dot{F}_{External} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M^{-1}(u - \tau_{External} - C\dot{q} - G) \\ J\dot{q} \\ \dot{J}\dot{q} + J\ddot{q} \\ B_{External}(\dot{J}\dot{q} + J\ddot{q}) \end{bmatrix}$$

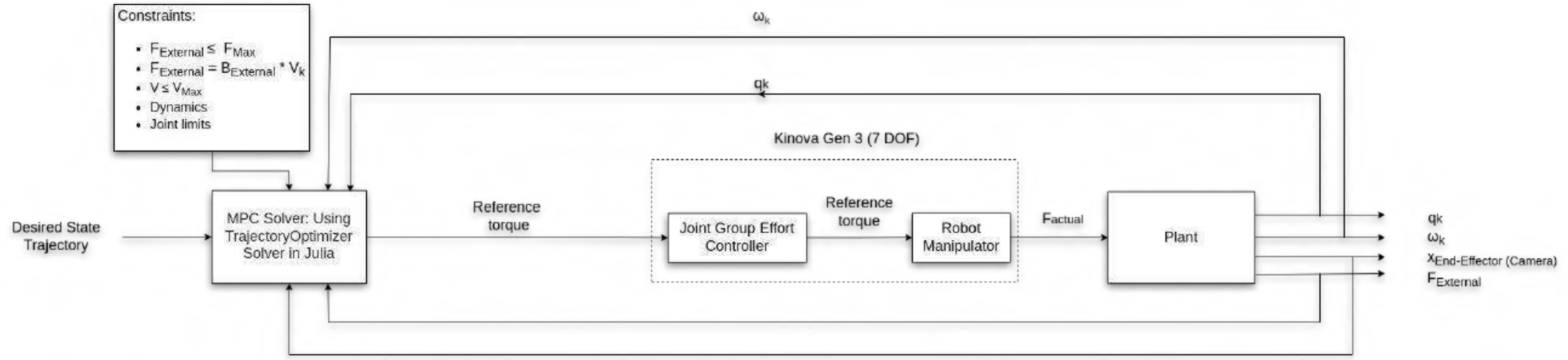
$$u = \tau_{Applied} \leq \tau_{Max}$$

$$||F_{External}|| = ||B_{External} J\dot{q}|| \leq F_{Max}$$

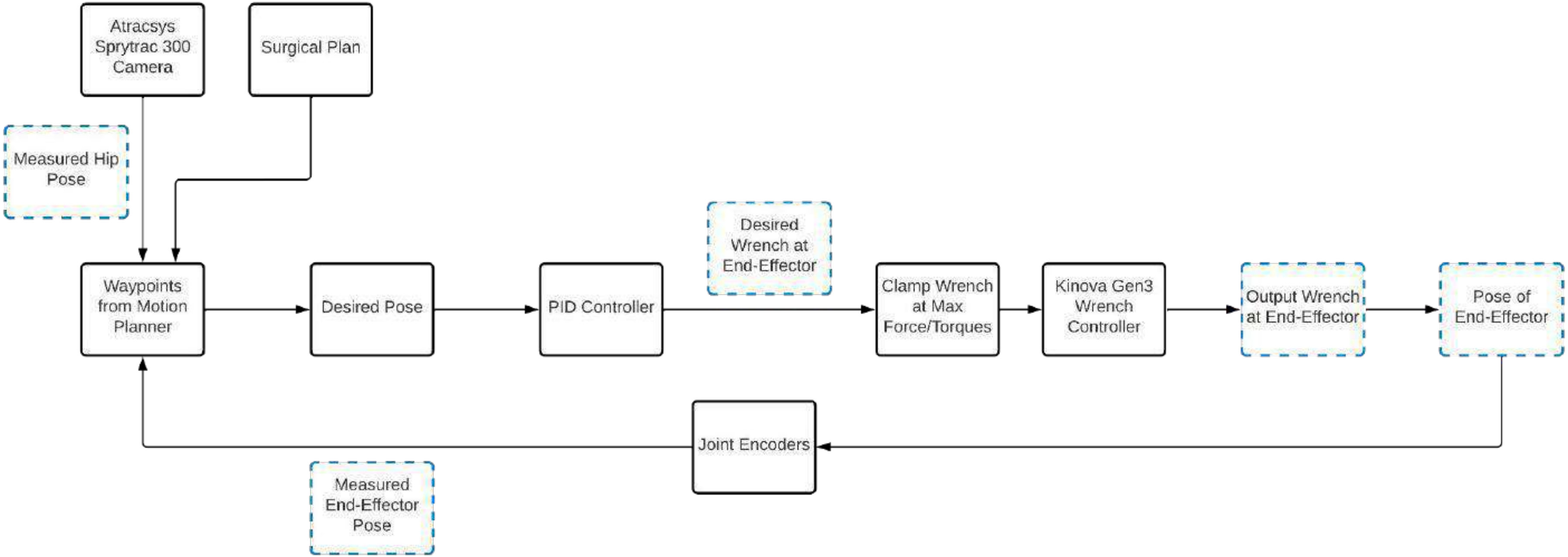
$$||\dot{x}|| = ||J\dot{q}|| \leq \dot{x}_{Max}$$

$$q \epsilon q_{Limits}$$

$$\dot{q} \epsilon \dot{q}_{Limits}$$



Wrench Controller Preliminary Analysis

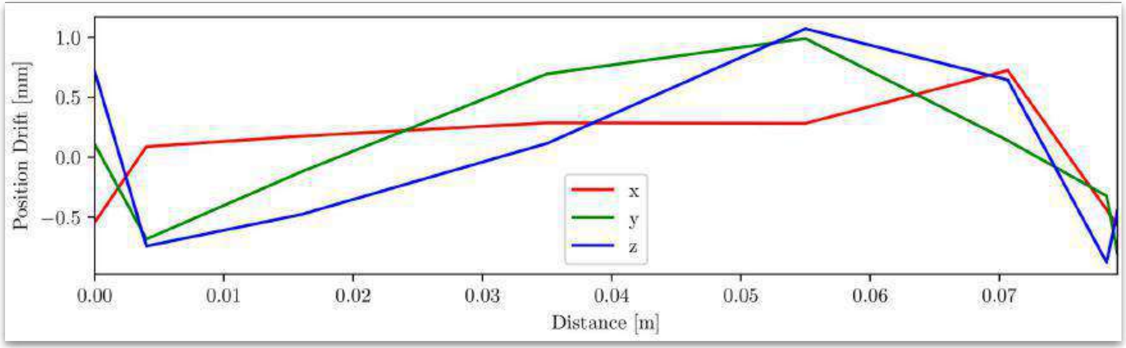


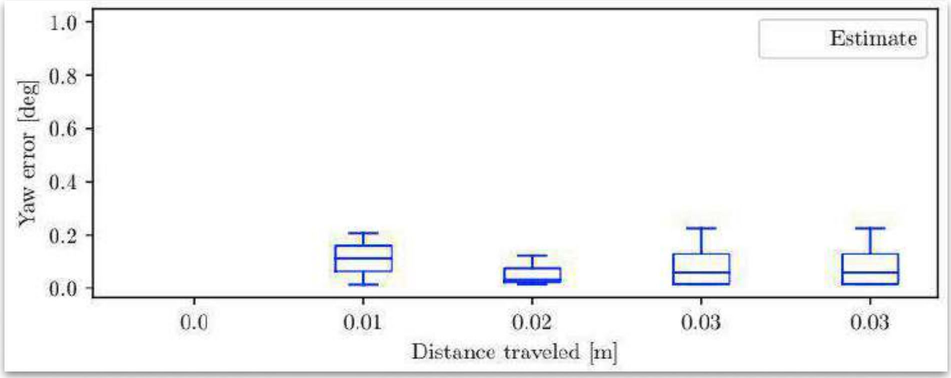
Perception + Planning + Controls

Dynamic Compensation

#	Status	Requirement	Achieved Performance	Features
M.P.5.	✓	The system shall adapt and compensate for movement by generating a new trajectory with a ***latency*** $\leq$ ***500ms***.	Planning and controls within tolerance.	Integration between Error Detection, Planning and Controls

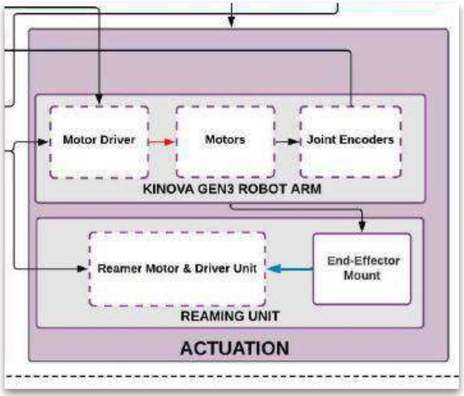
Testing & Evaluation





Subsystem: Planning and Controls		
Metric	**Required**	**Achieved**
Position Error (mm)	<= 3	2.61
Orientation Error (deg)	<= 3	0.241

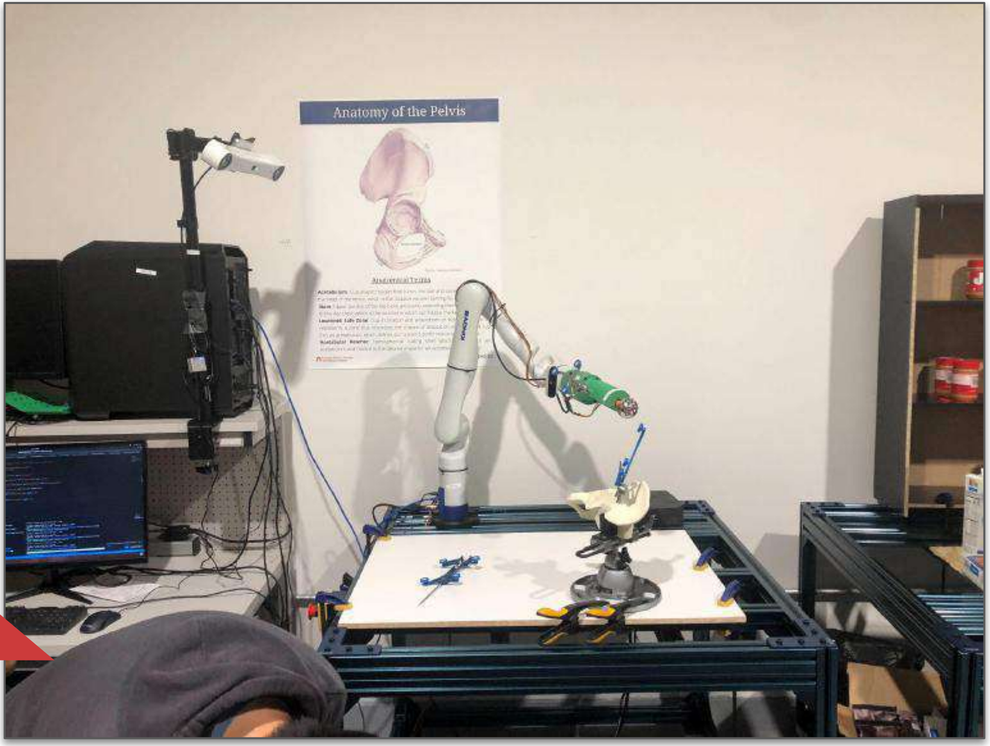
Hardware



#	Status	Requirement	Achieved Performance	Features
1	✓	The system end-effector shall be rigid and robust.	Minimal vibrations during movement	✓ 3D-Printed End-Effector Design
2	✓	The reamer should be able to maintain 0.5 Nm of torque while running at 400 rpm	Capable of maintaining 400 rpm at 0.5 Nm	✓ ServoCity Planetary Gear Motor
3	✓	The system PCB should be able to run the motor at a consistent rpm despite external torques	PID Controller run on arduino ensures consistent rpm	✓ Rosserial PID Controller (ROS Node) ✓ PCB and Power Supply

Hardware

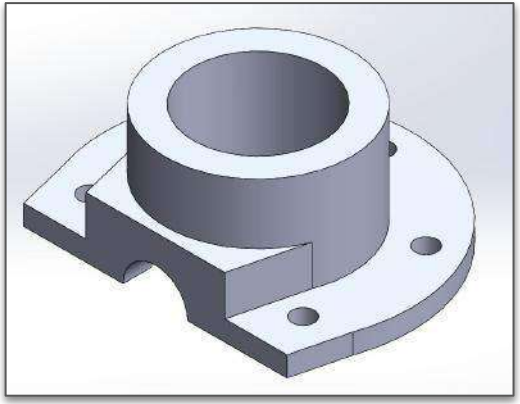
Workspace Setup

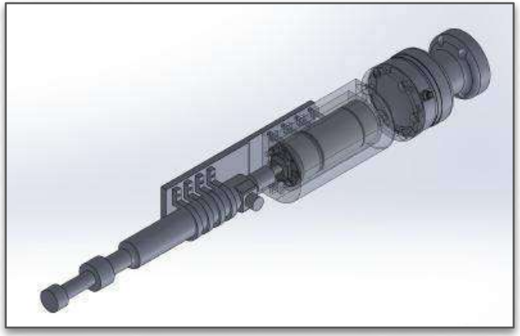


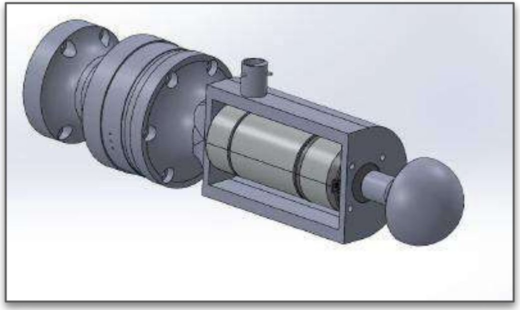
Anthony
(surgeon?)

Hardware

End-Effector Design Evolution

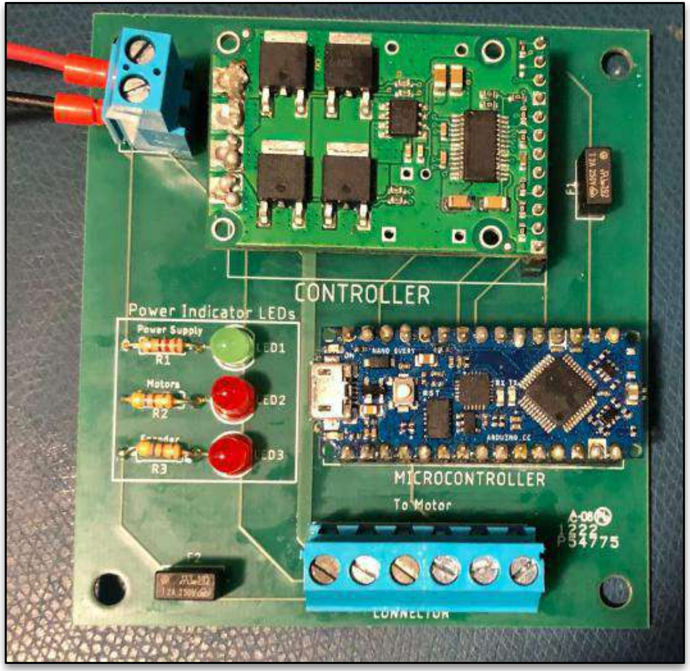


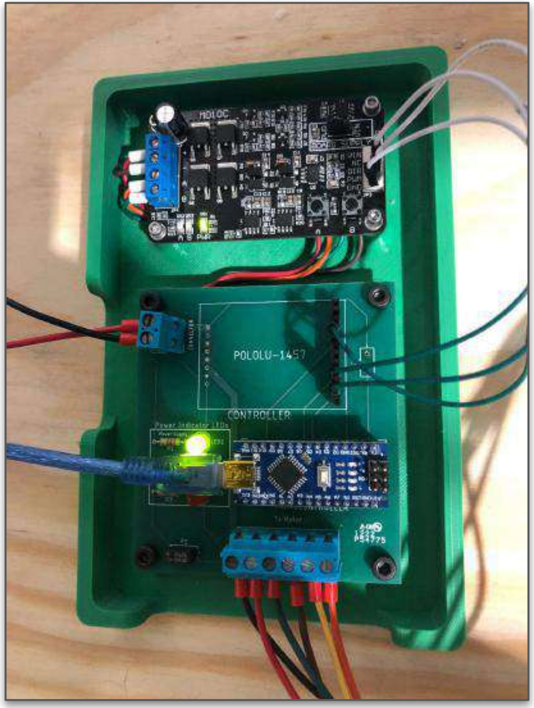




Hardware

PCB Design





Hardware

Force-Torque Sensor

ATI Axia80 Force Torque Sensor

Hardware

Potential Improvements

End-Effector:

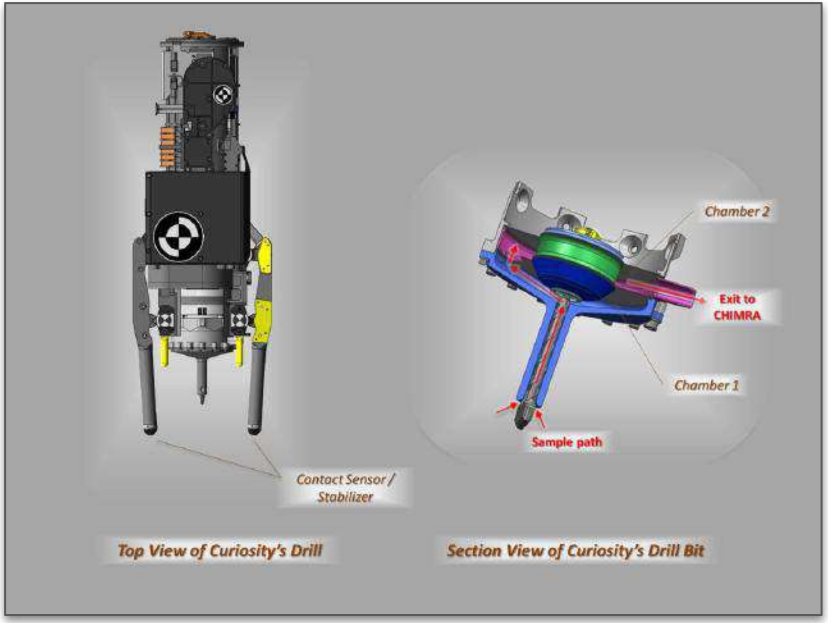
- Potential redesigns for more stability:
  - Turn end-effector 90°
  - Machine end-effector from aluminum or manufacture from plastics
  - Idea from Ben: Use similar design to Curiosity Rover’s Drill

PCB:

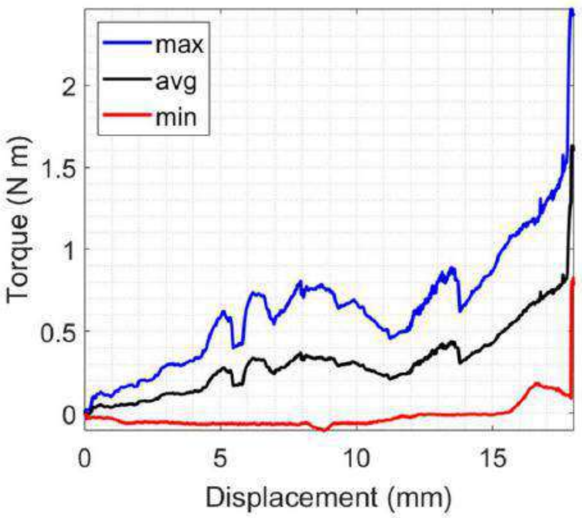
- Want to recreate PCB to get the entire motor controller on one board

FT Sensor:

- Use other communication protocols instead of Telnet
- Perform our own biasing rather than relying on the sensor



Hardware Analysis



Necessary Torque for Acetabular Reaming

From "A cadaver-based biomechanical model of acetabulum reaming for surgical virtual reality training simulators" by Pellicia

"In orthopedic surgery, reamers are driven by medical electric drill and the rotation speed is generally low, and no greater than 400 rpm"
From "Study on cutting force of reaming porcine bone and substitute bone" by Liu

No Load Speed: 612 rpm

Stall Torque: 1.6 Nm

Assuming a linear relationship between speed and torque, the motor should be able to maintain 400 rpm at 0.52 Nm.

SVD Performance Evaluation

Our personal thoughts:

- We think we did great and are super proud of how our demo turned out!
- Hope it was fun (if only a little bit scary to watch)!
- SVD went pretty much flawlessly and no errors showed themselves
- SVD-E had some bugs
    - Force-torque bias issue led to motor turning on too early
    - Trajectory issue where it got stuck at a waypoint
- Thanks for coming and watching!

SVD Performance Evaluation

#	Status	Requirement	Performance	Subsystem
M.P.1.1 M.P.1.2.1 M.P.1.2.2	✓	The system shall localize the robot arm in real-time with respect to the pelvis before and during surgery with a ***latency ≤ 500ms***.	Localization latency = **20ms**	Perception and Sensing
M.P.4.1-3	✓	The system shall compute error and interpret the movement of the pelvis during reaming with a ***latency ≤ 500 ms***, and detect changes with a ***position error of ≤ 3mm*** and ***orientation error ≤ 3 degrees***.	Latency = 20ms, Position error ≤ ***3mm,*** Orientation error ≤ ***3°***	

#	Status	Requirement	Performance	Subsystem
M.P.2.	✓	The system shall plan the trajectory of the robot arm based on the given surgical plan with a ***latency ≤ 500ms***.	Planning between **500ms - 600ms**.	Motion Planning and Controls
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 ≤ 3mm*** and ***orientation error ≤ 3 degrees.***	**Position error = 2.61mm, Orientation error = 0.24 degrees**	
M.P.5.	✓	The system shall adapt and compensate for movement by generating a new trajectory with a ***latency ≤ 500ms***.	Planning between **500ms - 600ms**.	Integration

Autonomous Reaming for Total Hip Replacement (ARTHuR)

HIPSTER

MRSD Class of 2023: Team C
Kaushik Balasundar | Parker Hill | Anthony Kyu
Sundaram Seivur | Gunjan Sethi

Strengths and Weaknesses of System

Strengths

- Modular
- Repeatable
- Low Latency
- Accurate
- Robust to human error
- Safe(ish)

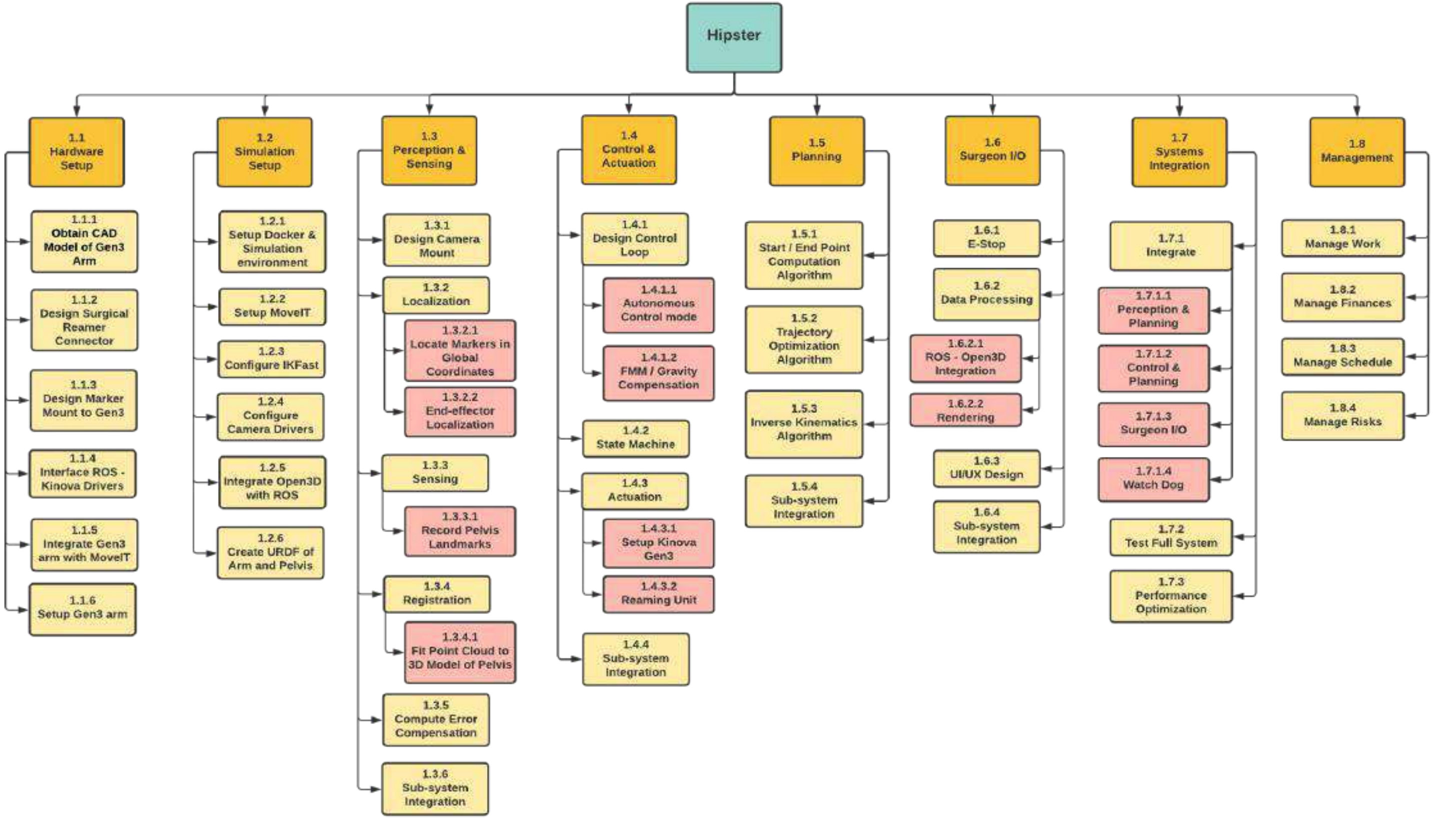
Weaknesses

- Edge Cases
- Training Required
- Sensitive
- Vibration
- Controlled Environment

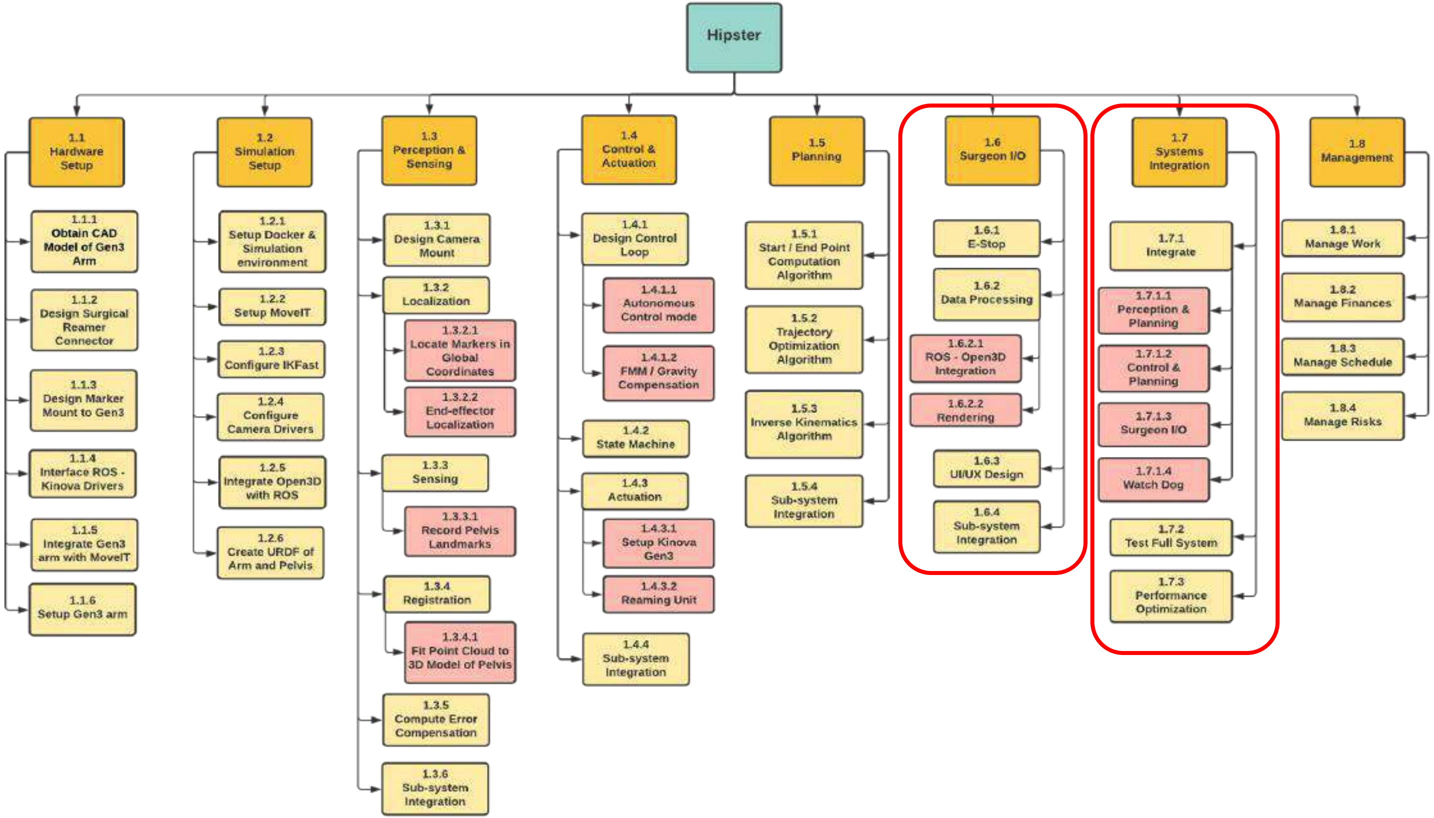
Project Management

- Work Breakdown Structure
- Schedule Status
- Test Plan
- Budget Status
- Risk Management

Work Breakdown Structure

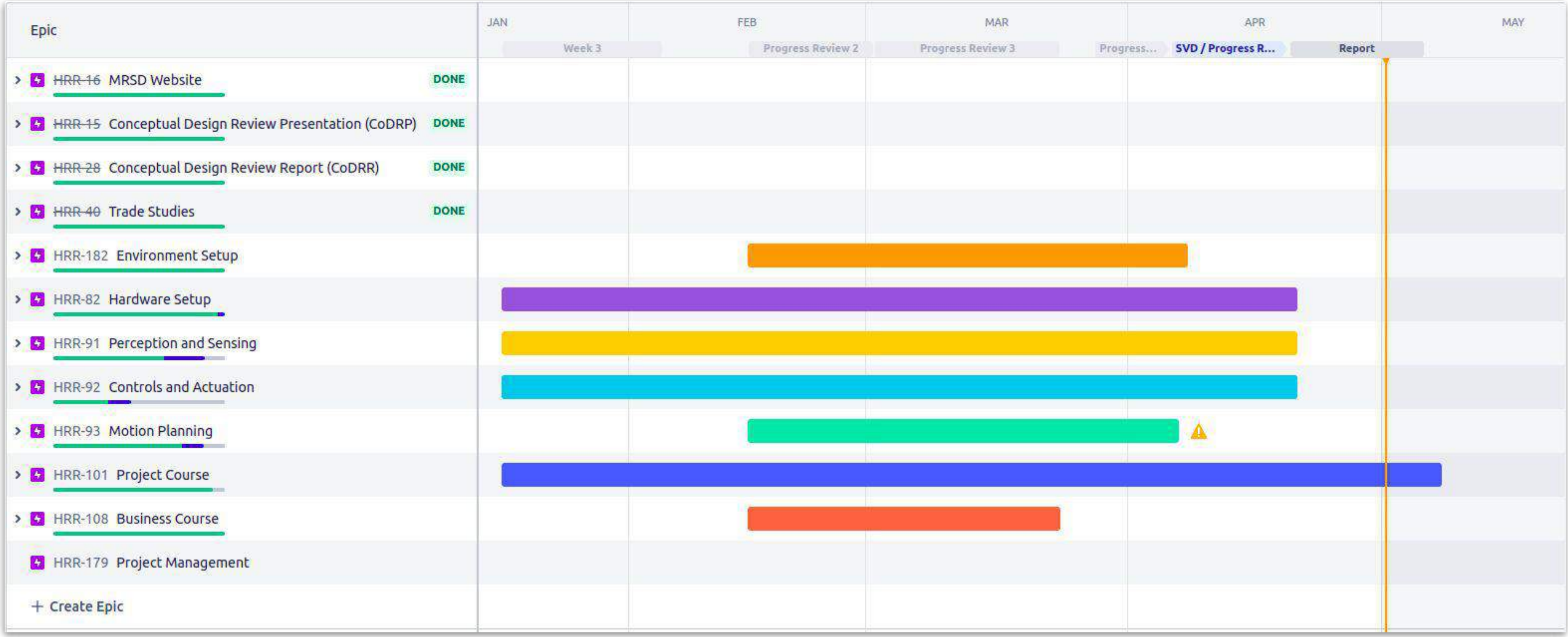


Work Breakdown Structure



Project Management

Spring Schedule



Project Management

Schedule

Status: **On Track!**

A. **Perception**
- ✓ Marker Detection & Tracking
- ✓ Error Detection
- ✓ Point Cloud Collection
- ✓ Registration
- ✓ Dynamic Compensation

B. **Motion Planning**
- ✓ IK-Fast Plugin
- ✓ Pilz Planning Pipeline
- ✓ Latency < 500 ms

C. **Controls**
- ✓ Preliminary Dynamic Compensation
- ✓ Wrench Controller

D. **Hardware**
- ✓ Preliminary PCB
- ✓ Preliminary Reamer end-effector design
- ✓ Force-Torque Sensor

Test Plan

Test Number	Capability Milestone	Location	Sequence of Events	Metrics	Performance Requirements
1	Free Motion Mode	NSH Basement	1. Place pelvis onto the table 2. Power on the robot arm 3. Hold the end-effector and gently manouver it to bring it close to the pelvis.	Personnel can move robot arm around freely	M.P.6.
2	Subsystem Integration (Percpetion + Planning + Controls)	NSH Basement	1. Run the procedure start script. 2. Record the latency of the initial pose localization.	Robot can determine its pose with respect to the pelvis within 50ms, within a pose and orientation error of 1mm and 1.5 degrees respectively	M.P.1.1 M.P.1.2.1 M.P.1.2.2 M.P.4.2.
			3. Record the time taken for the motion plan to be generated	The latency must be within 150ms.	M.P.2.
			4. Robot follows the trajectory. 5. Run Quantitative Trajectory Evaluator	The trajectory must be followed within the defined root mean square error threshold.	M.P.3.1 M.P.3.2
3	Dynamic Compensation	NSH Basement	1. As a continuation to Test 2, move the pelvis' position and orientation from its initial pose. 2. 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.	M.P.4.1 M.P.5
4	Surgeon I/O	NSH Basement	1. Start the surgeon I/O during each test. 2. The robot and the pelvis, in their current states, must be visible on the surgeon I/O	Latency within 150 ms	M.P.7.
5	E-Stop	NSH Basement	1. Run the procedure start script. 2. Press the e-stop button mid-execution	Must stop the system within 500 ms	M.P.8.

Test Plan: PR Goals

PR1:
1. Wireframes for surgeon I/O
2. Ideas & plans for dynamic compensation
3. Ideas for hardware improvement

PR2:
1. Basic implementation of dynamic compensation
2. Optimization for lower planning time
3. Increased robustness of perception and controls sub-systems

PR3:
1. Open3D & Marker Visualizations on Surgeon I/O
2. Improved hardware designs in CAD
3. Updated simulation environment

PR4:
1. Watchdog integration
2. Prototype & test improved hardware
3. Pointcloud collection visualized on Surgeon I/O

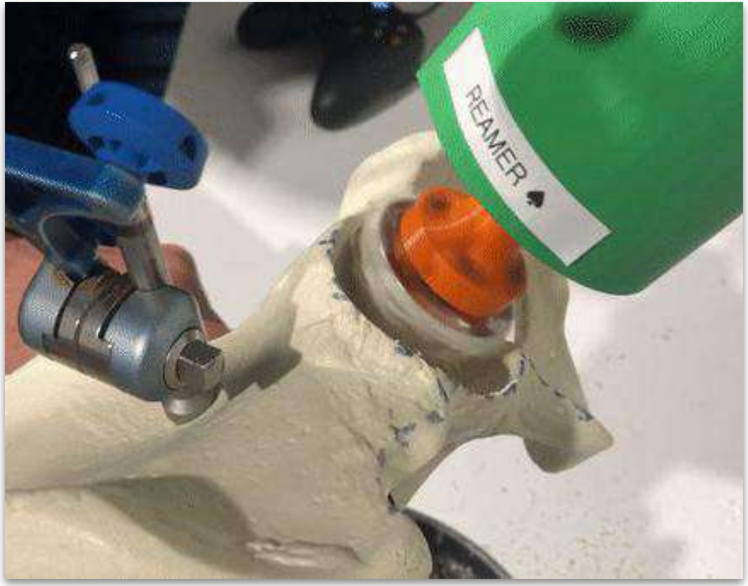
PR5:
1. Finalize hardware
2. Integrate planning & controls visualization in Surgeon I/O
3. Real-time dynamic compensation

PR6:
1. Full system integration
2. Fully functional surgeon I/O
3. Repeatability Testing

Fall Validation Demonstration

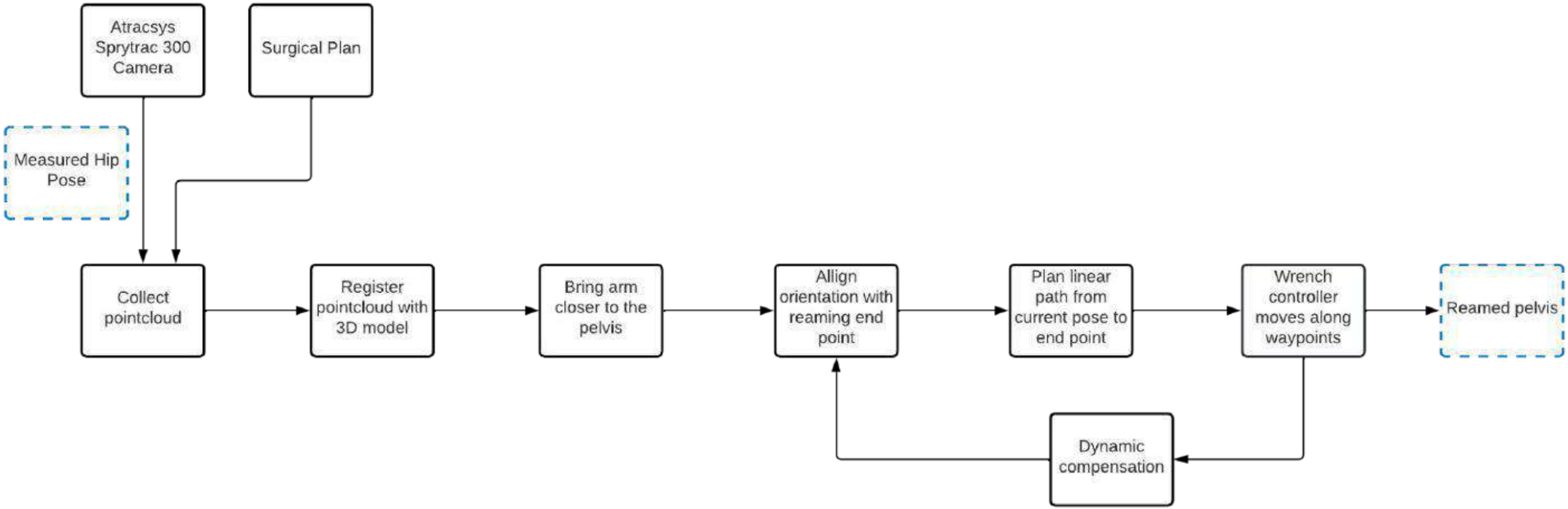
Location: NSH B512

Necessary Equipment: Atracsys camera and vesa mount, Sawbone pelvis and vise, Kinova Gen-3 robot arm, Vention table, reaming end-effector, MRSD Computer, Motor control PCB



Fall Validation Demonstration

Procedure



Fall Validation Demonstration

Procedure

1. Begin by setting up the work environment by **clamping the Sawbone pelvis** in a new position in a vise, **fixing a fiducial marker screw mount** on the pelvis, and **placing the fiducial marker** onto the end-effector of the robot arm.
2. Utilizing a probe, the **pelvis will be localized** using a point cloud to fit the pelvis to a known pelvis mesh, from which the endpoint of the reaming operation will be determined
3. Utilizing free motion mode, the **robot arm will be placed** near the center of the acetabulum.
4. The **reaming operation would then be started**, allowing the robot arm to localize itself with respect to the pelvis and begin generating a motion plan.
5. Once the **reaming motor turns on and the arm begins to move**, contacting the pelvis, the e-stop is hit to demonstrate the safety of the system.
6. The **robot arm will then be reset with free motion mode** and the reaming operation would then be allowed to progress freely.
7. As the robot arm begins to ream the acetabulum, the **pelvis would be shifted by hand** using the vise, to demonstrate the robot arm's capability of adapting to pelvic motion.
8. When the robot arm has completed the reaming operation, it will **remove itself from the pelvis**, and the resulting acetabulum can be analyzed.

Budget Status

Total Budget	5000
Total Spent	574.22
Balance	4425.78

Sr. No	Owner	Description	Month	Amount	Tax	Delivery	Total
1	Sundaram	Hemi Pelvis 48mm - Solid foam	Nov	40	4.805	9.3	54.105
2	Sundaram	Hemi Pelvis 56mm with vise attachment - Solid foam	Nov	40	4.805	9.3	54.105
3	Sundaram	Monitor Desk Mount Vesa	Nov	40	2.4	0	42.4
4	Sundaram	Panavise Mount for holding sawbones	Nov	90	6.3	0	96.3
5	Gunjan	LENTION USB-C Multi-Port Hub with 4K HDMI Output, 4 USB 3.0, Type C Charging Compatible 2021-2016 MacBook Pro, New Mac Air & Surface, Chromebook, More, Stable Driver Adapter (CB-C35, Space Gray)	Jan	35.99	2.16	0	38.15
6	Sundaram	Dry-erase markers	March	6.92			6.92
7	Parker	Servocity Planetary Gear Motor	March	59.99			59.99
8	Parker	HD Premium Planetary Gear Motor Mount, Face Tapped	March	4.99			4.99
9	Parker	6mm to 8mm Flexible Clamping Shaft Coupler	March	5.99			5.99
10	Parker	6mm to 0.250" Flexible Clamping Shaft Coupler	March	5.99			5.99
11	Parker	PH Series JST 6-pin connector (2mm Pitch)	March	0.99			0.99
12	Sundaram	Ethernet Splitter	April	$17.99			$17.99
13	Sundaram	Late July chips	April	$39.50			$39.50
14	Sundaram	Cord sleeve	April	$15.98			$15.98
15	Sundaram	Power Strip	April	$14.56			$14.56
16	Sundaram	Cable waterproof sleeve	April	$19.99			$19.99
17	Sundaram	Surgeon gown	April	$27.99			$27.99
18	Sundaram	HDMI cable	April	$7.30			$7.30
19	Sundaram	Presentation laser pointer	April	$27.99			$27.99
20	Sundaram	Conference mic	April	$32.99			$32.99

Budget	Expenditure	Balance
$5000	$574.22 (11.50%)	$4425.78

Risk Management

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 • 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 • Store robot arm in safe environment • 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 • 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 • 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 • Revisit performance requirements every sprint meeting • 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 • Host end-of-sprint meetings • Create documentation at the end of every sprint
7	Camera hardware fails	Technical	2	4	• Store camera in a safe location • Design pipeline for the use of the camera • Ask sponsor for a backup camera to use in an emergency • 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 • 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 • 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 • 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 • Discuss with sponsor potential back-up workspace

*Red indicates biggest risks

Risk Management

Risks Realized and Mitigated

Risk #	Risk	Type	Likelihood #	Consequence #
1	Robot arm does not arrive on time	Schedule	2	4
2	Robot arm breaks	Technical	2	5
3	ROS simulation does not match up to reality	Technical	4	2
4	Too many requirements	Schedule	3	3
5	Performance requirements not met	Programmatic	4	4
6	Integration issues between subsystems	Technical	5	4
7	Camera hardware fails	Technical	2	4
8	ROS and IGSTK data conversion difficulties	Technical	4	2
9	Team member has difficulties working on their part of the project	Programmatic	5	2
10	Development Environment Incompatibility	Technical	5	1
11	Unable to access workspace	Programmatic	1	5

Likelihood						
	5	#10	#9			
	4		#3		#5	#6
	3			#4		
	2				#7	#2
	1	#1, #8				#11
		1	2	3	4	5
		Consequence				

Before PDR

Likelihood						
	5	#10	#9			
	4		#3, #5			
	3			#4		
	2				#7	#2,#6
	1	#1, #8				#11
		1	2	3	4	5
		Consequence				

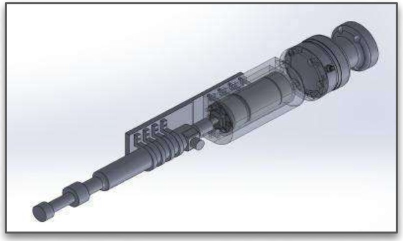
After SVD-E

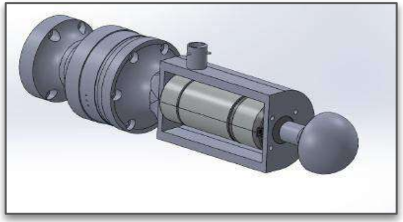
Conclusions

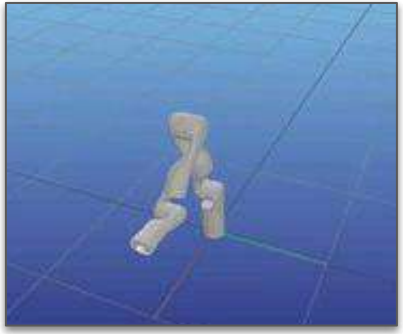
- Lessons Learned
- Key Fall Activities

Lessons Learned

- Fail faster and pivot!
  - MPC → PID Controller
  - Reamer Design
- Better use of Jira
  - Time logging
  - Consistent Updates
- Better Knowledge Transfer
- Rigorous Testing Framework
  - Unit Testing
  - Continuous Integration / Development



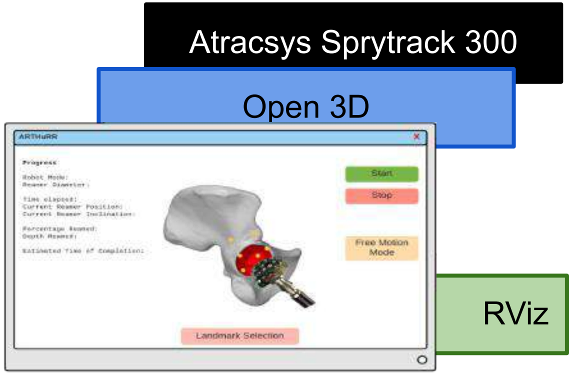


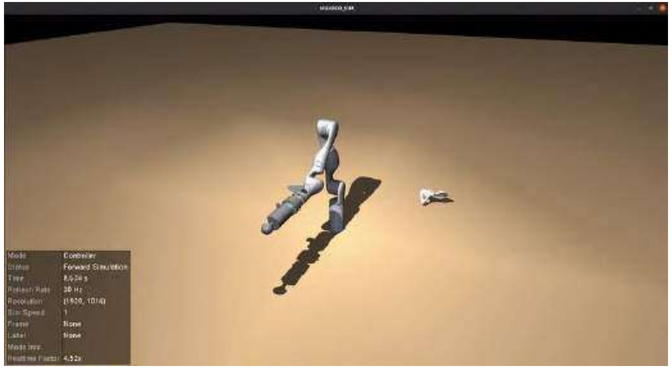


Jira

Key Fall Activities

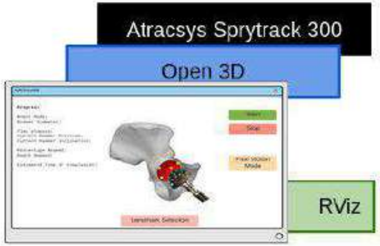
- Improving System Robustness
- Surgeon I/O Design
  - Better System Usability
  - Coherent Integration
- Watch Dog & Dynamic Compensation
- Update Simulation Environment
  - Better simulator for rapid & accurate testing
- Improving Controller
  - If time permits
  - Improve MPC convergence speed

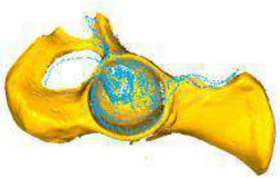


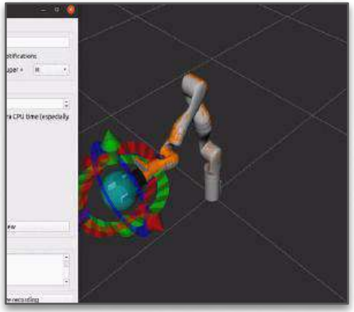


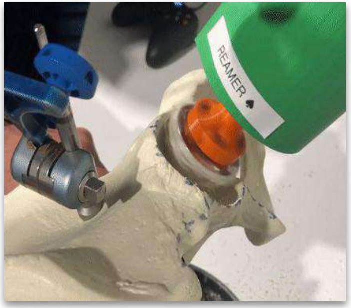
Summary

- Successfully demonstrated autonomous reaming:
  - Point Cloud Collection
  - Registration
  - Error Detection
  - Motion Planning
  - Wrench Controller
  - Dynamic Compensation
- Full system integration
- Performance Validation









Thank You!
See you next semester :)

Autonomous Reaming for Total Hip Replacement (ARTHuR)