Autonomous Reaming for Total Hip Replacement (ARTHuR)

Progress Review - 7

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

September 7th, 2022

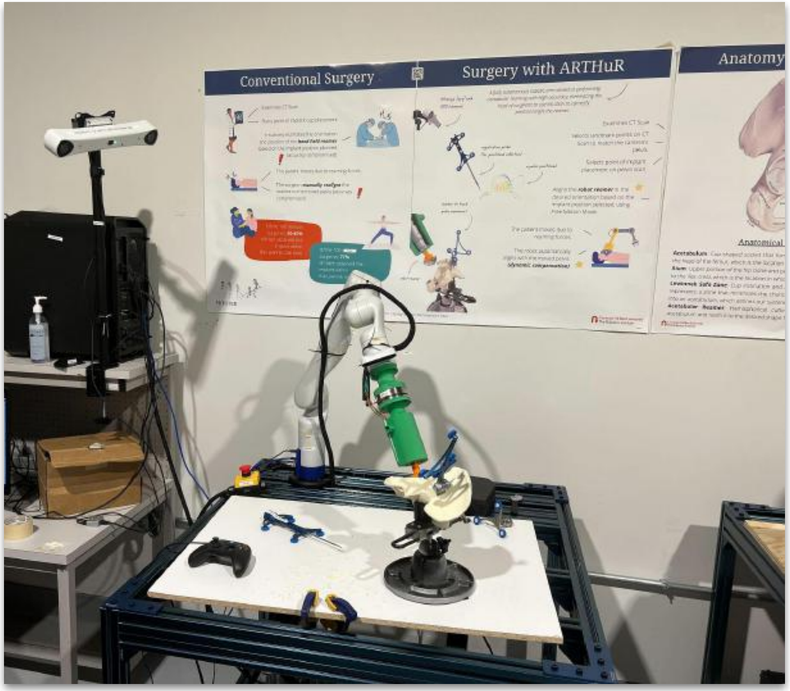
Progress Review #7 Goals

- Collect hardware from REDACTED
- Re-assemble system/workspace
- Re-evaluate requirements, risks, feedback, and system architectures
- Re-familiarization with system
- REDACTED kickoff meeting
- End-effector redesign brainstorming

Progress and Challenges

Re-assemble System

- Collected arm and camera from Smith & Nephew on August 31st
- Re-assembled system:
  - Attached camera back to VESA mount
  - Mounted Kinova onto table
  - Attached end-effector onto arm
  - Re-routed cables and wires throughout the system



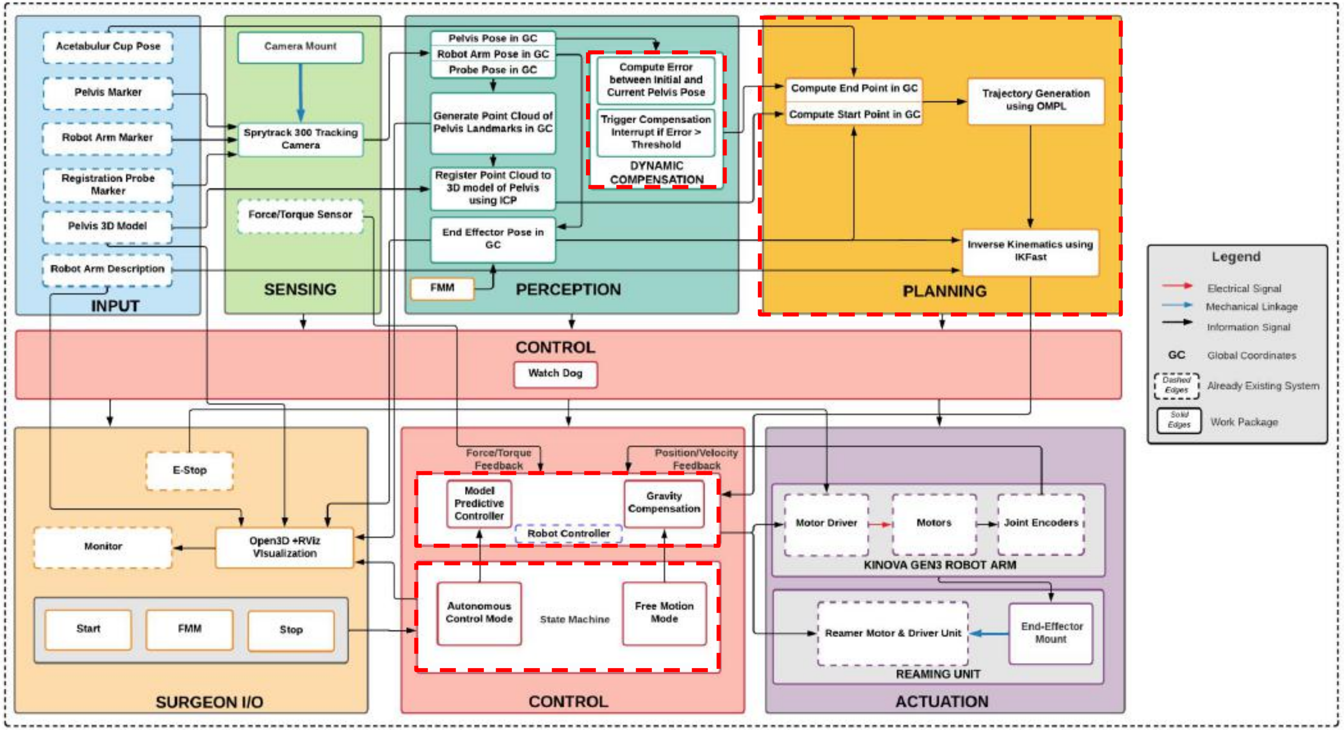
Re-Evaluated Requirements

Functional Requirement	Performance Requirement
M.F.1. The system shall use the Atracsys camera to track the pelvis, registration probe, and robot arm markers.	M.P.1.1. The system shall use the Atracsys camera to track the pelvis, registration probe, and robot arm markers with a frame rate greater than or equal to **50 Hz** or latency less than or equal to **20 milliseconds**.
	M.P.1.2. The system shall use the Atracsys camera to track the pelvis, registration probe, and robot arm markers with an accuracy of less than or equal to **0.5 mm**.
M.F.2. The system shall continuously calculate the error in pelvis movement.	M.P.2.1. The system shall continuously calculate the error in pelvis movement with a frame rate greater than or equal to **50 Hz** or latency less than or equal to **20 milliseconds**.
	M.P.2.2. The system shall use the Atracsys camera to track the pelvis, registration probe, and robot arm markers with a positional accuracy less than or equal to **2 mm**.
	M.P.2.3. The system shall use the Atracsys camera to track the pelvis, registration probe, and robot arm markers with an orientational accuracy less than or equal to **1.5 degrees**.
M.F.3. The system shall perform registration between the collected pointcloud and the given 3D pelvis scan.	M.P.3. The system shall perform registration between the collected pointcloud and the given 3D pelvis scan with a root mean square (RMS) error of **0.1 mm**.
M.F.4. The system shall dynamically compensate for the movement of the pelvis.	M.P.4.1. The system shall dynamically compensate for the movement of the pelvis by retracting or powering off the reamer with a latency of less than or equal to **25 ms**.
	M.P.4.2. The system shall dynamically compensate for the movement of the pelvis by realigning the reamer with a latency of less than or equal to **50 ms**.
M.F.5. The system shall ream the pelvis based on the provided surgical plan.	M.P.5.1. The system shall ream the pelvis based on the provided surgical plan with a positional accuracy of **2 mm**.
	M.P.5.2.The system shall ream the pelvis based on the provided surgical plan with an orientational accuracy of **1.5 degrees**.
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 in an initial position by back-driving the robotic arm
M.F.7. The system shall allow the surgeon to e-stop	M.P.7 The system will allow the surgeon to e-stop the system, stopping the system within a latency of **500 ms**

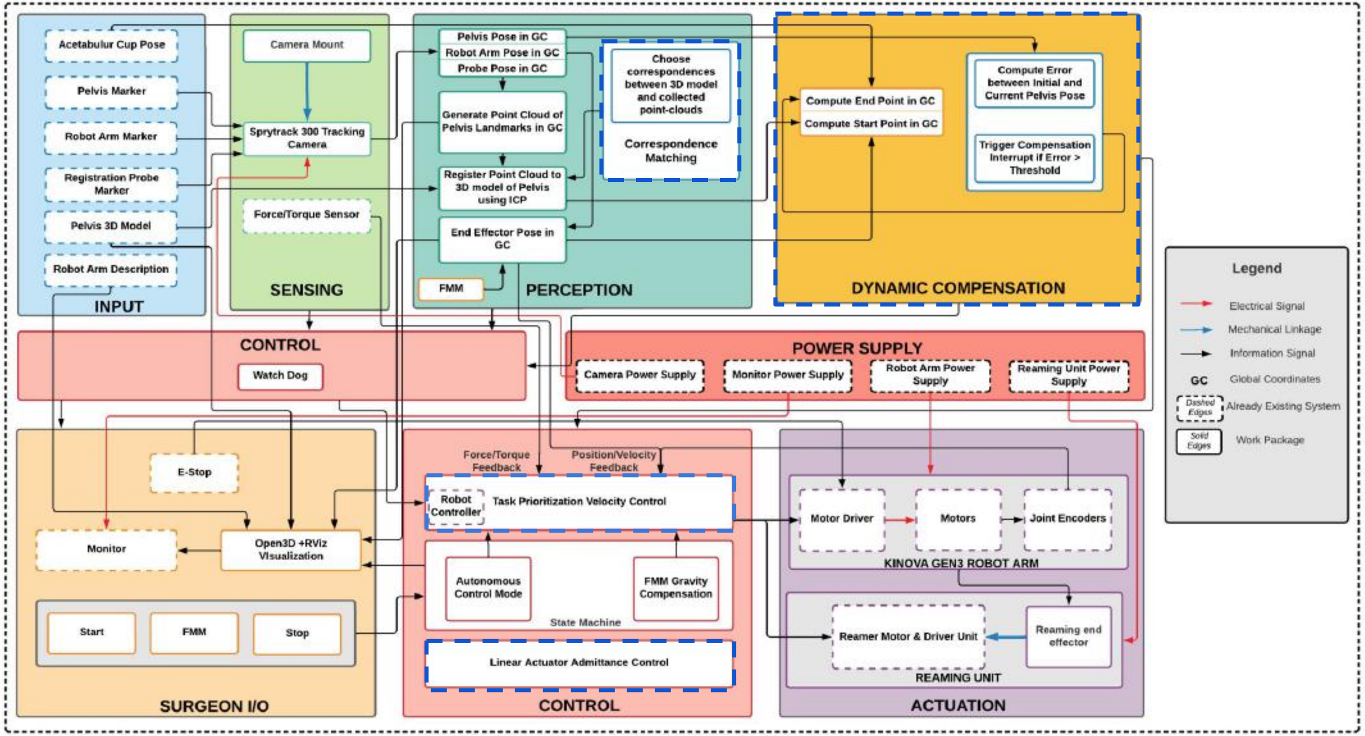
New Risks

1	**End effector development delays**	Technical	4	4	• Start early and lock design by October 15th • Brainstorm multiple solutions • 3D print to test before final manufacturing • Make system simple enough to get it manufactured in-house • Use 3D printed design as a fallback
2	**PCB failure & delays**	Technical	4	4	• Test breadboard prototype • Seek feedback from Luis • Use off-the-shelf boards to reduce PCB complexity of PCB • Order spares
3	**Dynamic compensation not achievable**	Technical	3	3	• Iterate with multiple control architectures • Utilize earlier compensation solution as a fallback • Evaluate need for isolating controller from ROS • Benchmark latencies in system • Use pub/sub communication instead of server/client communication
4	**System robustness issues**	Technical	4	3	• Create internal deadlines • Start testing early (by Oct 31st) • Fallback on the earlier working demo • Allocate last two weeks for system optimization
5	**Reamer does not ream rigid bone**	Technical	3	2	• Fallback on foam bones • Order bones early for testing
6	**UI does not integrate with system**	Technical	4	2	• Start working on the UI early • Plan architecture and consult each stakeholder • Start testing by Oct 31st

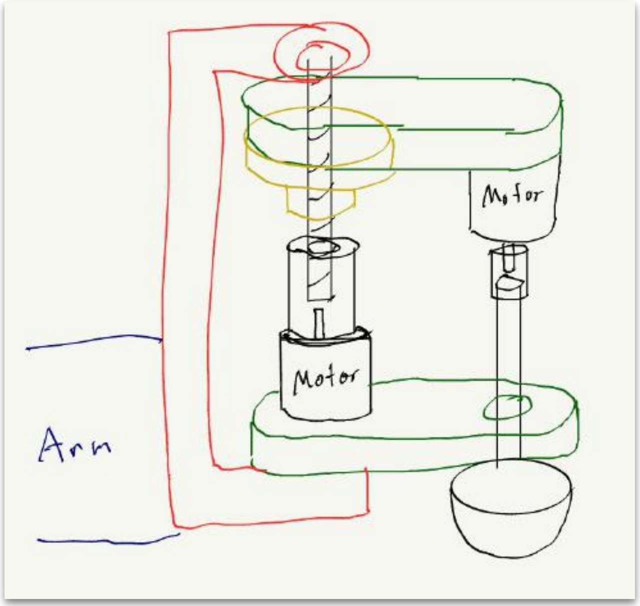
Old Cyberphysical Architecture

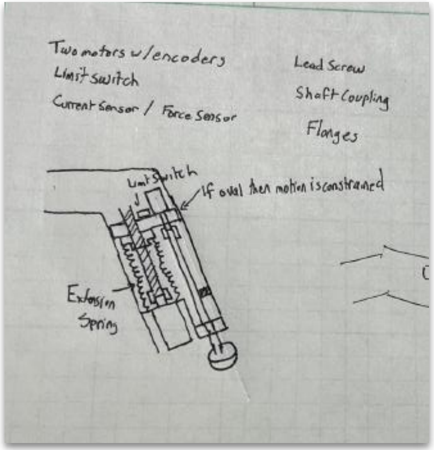


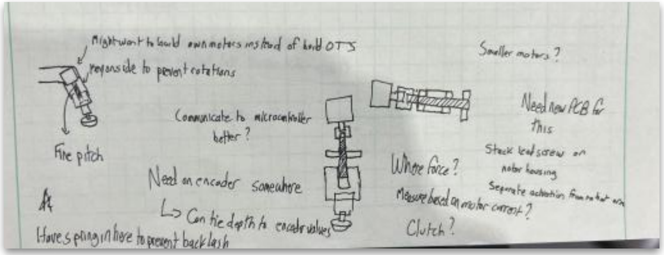
Re-evaluated Cyberphysical Architecture

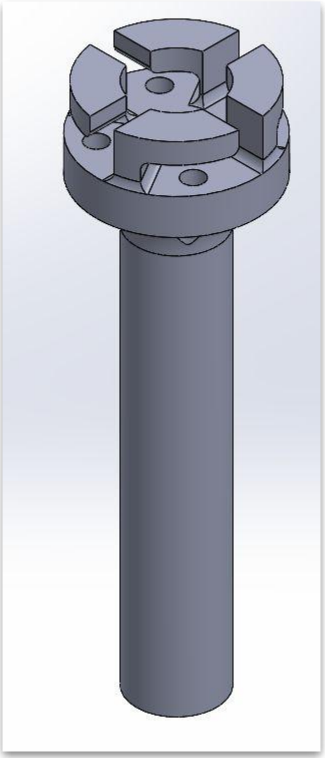


End-Effector Redesign Brainstorming



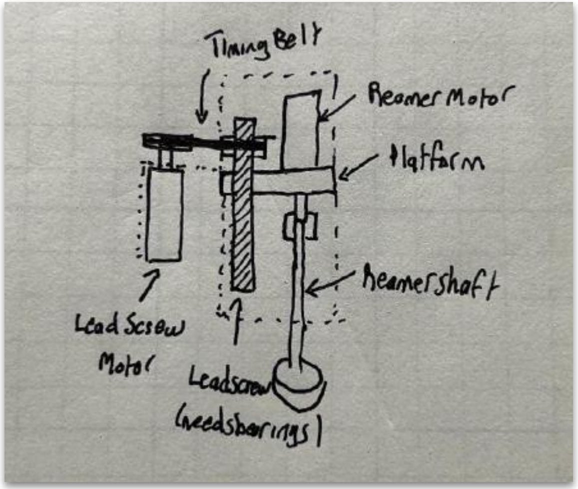


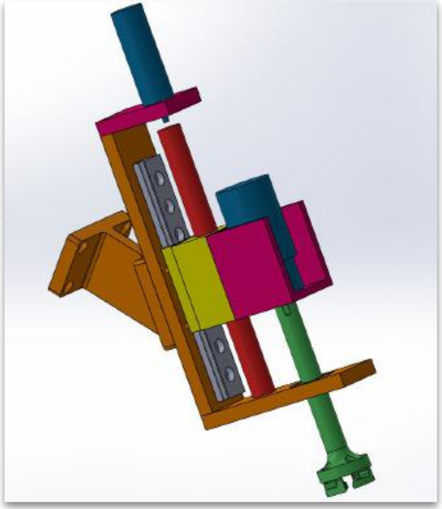




End-Effector Redesign Challenges

- Reaming shaft ideally is supported linearly and rotationally
  - Requires a linear and rotational bearing, over-constraining the system
  - Have to choose between linear and rotational support
    - Need linear support to move stage down, so no rotational bearings
- Length needs to be minimized
  - Motors need to be attached to minimize overall length of end-effector
- Need way to measure downward applied force





Watchdog Module

Goals

- Log all important inputs, outputs ar
  system metrics
- Prevent system from breaking dow
  to sub-system level errors

Inputs<> WatchDog

		Error metric, latency
Reaming Point Selection	Filesystem	Same as input Acetabular Cup Pose

Dynamic Compensation <> WatchDog

Value	Type of Logging	Description
Pelvis Pose Error Calculation	Terminal (just a message maybe?)	Error < threshold
Trigger DC	Filesystem (log pose + timestamp!)	Error > threshold
Arm realignment	Terminal, Filesystem (with timestamp, latency)	Has the arm realigned successfully?

Control <> WatchDog

Value	Type of Logging	Description
Singularity Check	Terminal - Error	Log if joints are close to or at singularity
Reaming Progress	Terminal	depth of reaming completed
Reamer ON/OFF	Terminal	
ESTOP	Terminal	Button to kill all processes
Actuator Health	Terminal (maybe a plot?)	Motor speed, force values etc (?)
PCB Health	Terminal	Current

New Project Management

- Jira doesn't work for us
- Using a white board and sticky notes for easier task management amongst the team
- Will continue using Jira for scheduling and Confluence for documentation
- Standups every Monday

Improved Dynamic Compensation

How?

- Tuned PID gains of wrench controller
- Increased controller rate from 20 to 40 hz
- Removed planner, which caused latency

Limitations

- Controller limited to 40 hz (API bottleneck)
- Only control of end-effector, not individual joints
- Can’t execute multiple tasks simultaneously

Task Prioritization Kinematic Controller

Tasks:

1. Reaming Alignment Task (5 DoF)
2. Camera Alignment Task (1-2 DoF)
3. Singularity Avoidance Task (1 DoF)
4. Joint Limit Avoidance Task (1 DoF)

Task Prioritization Algorithm

$$\dot{q} = \sum_{i=1}^{n} \dot{q}_i, \text{ where}$$

$$\dot{q}_i = \left(J_i\left(\prod_{j=1}^{i-1} N_j\right)\right)^{\#}\left(\dot{x}_i - J_i \cdot \left(\sum_{j=1}^{i-1} \dot{q}_j\right)\right),$$

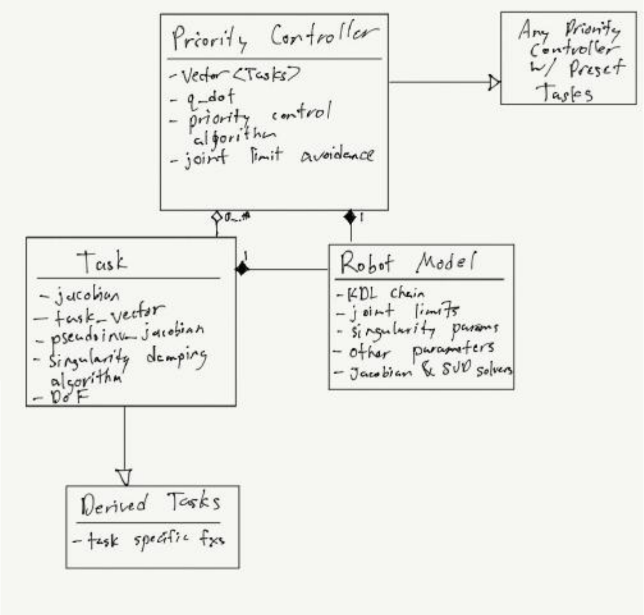
$$N_j = \left(I - \left(J_{j-1} \cdot N_{j-1}\right)^{\#} \cdot \left(J_{j-1} \cdot N_{j-1}\right)\right),$$

$$N_1 = I, \text{ and}$$

n is the number of tasks, where highest priority is i = 1.

J_i is the jacobian for a task.

N_j is the null space of a task.



Future Work

Future Work

- CAD/3D print of new end-effector design
- Source off-the-shelf components needed for new end-effector design
- Watchdog/UI wireframe and initial implementation
- Test tracking performance of basic Inverse Kinematic Controller (1000 Hz)
    - Determine if we want to pursue this controller or keep wrench controller
- Implement online base estimation to remove need of offline hand-eye calibration

Thank you!