NSF CMMI #1734461, #1734360

NRI-2.0: INT: Manufacturing America:

In-Situ Collaborative Robotics in Confined Spaces

Nabil Simaan[1]

Howie Choset[2]

Andrew Orekhov[1]

Colette Abah[1]

Garrison Johnston[1]

David Neiman[2]

Saumya Saxena[2]

[1] Vanderbilt University

[2] Carnegie Mellon University

Introduction

Motivation:

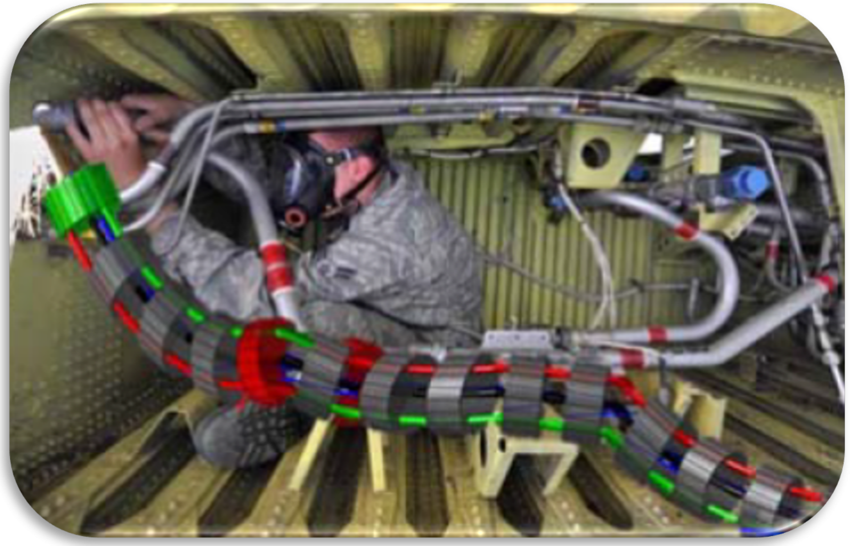
- WMDs: Work related musculoskeletal disorders: risks increase in confined space
- Over 600,000 WMD/year account for 34% of lost workdays

Vision:

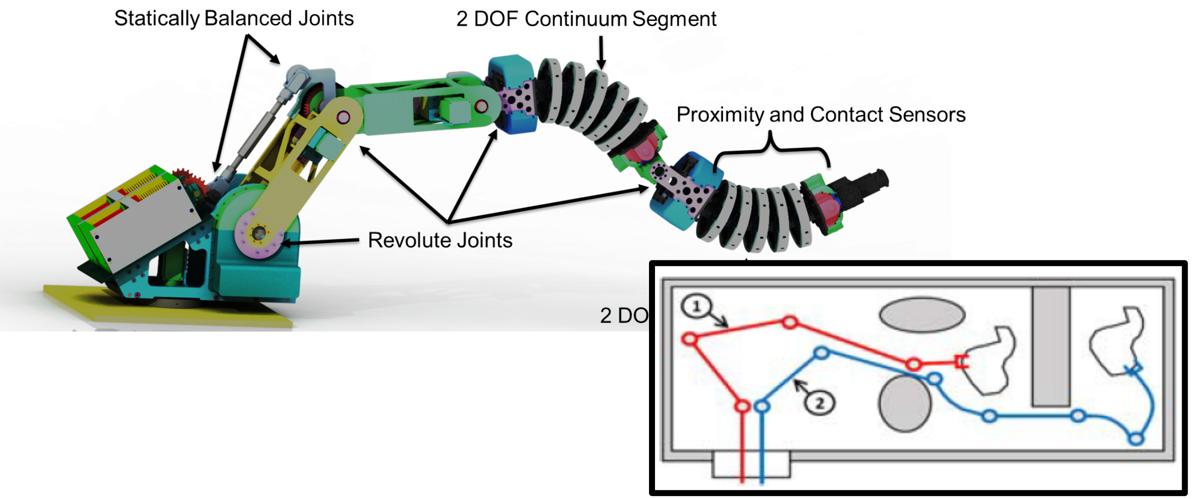
- Enable safe and highly dexterous cooperative robotic manipulation in deep confined spaces.

Approach:

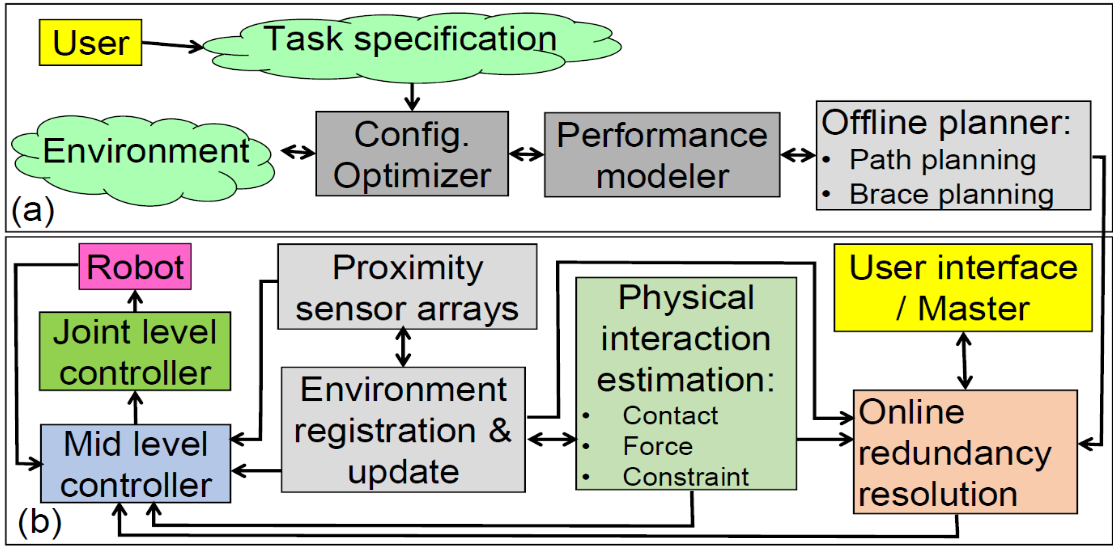
- Reconfigurable serial-continuum robots
- Whole body sensing and interaction
- Planning and control for bracing
- Sensing & environment model update



Envisioned Embodiment



Planning, Sensing & Control



Evaluation

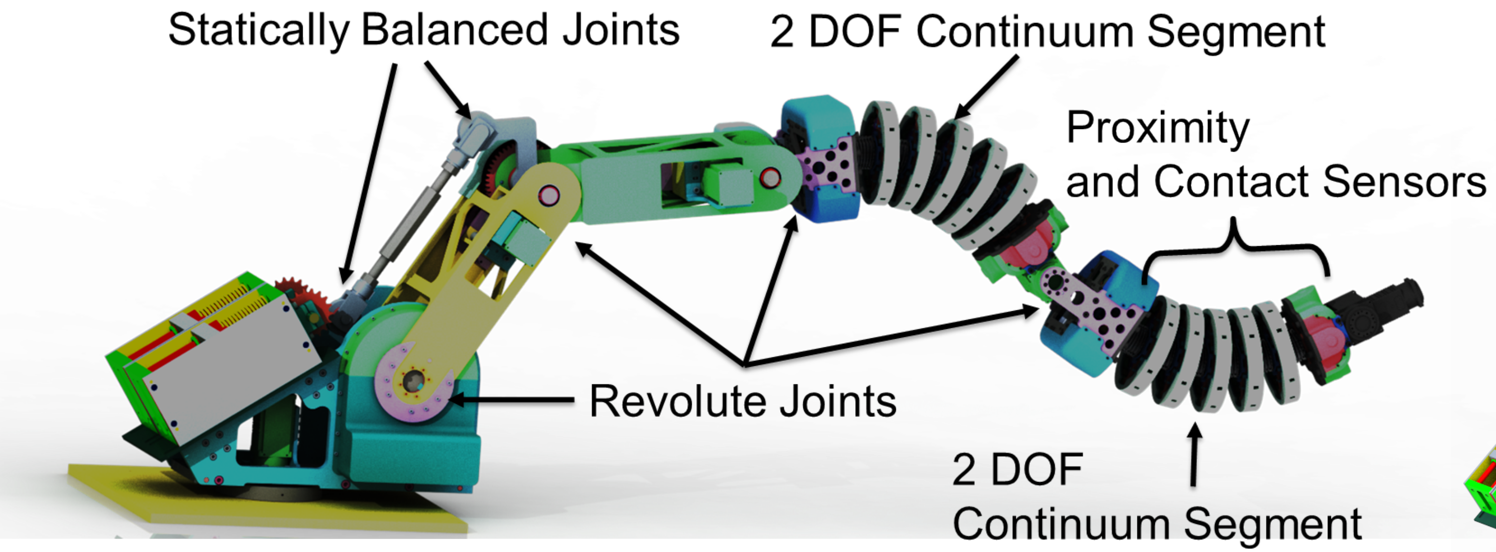
- Evaluate tasks in cooperative manipulation & tele-manipulation

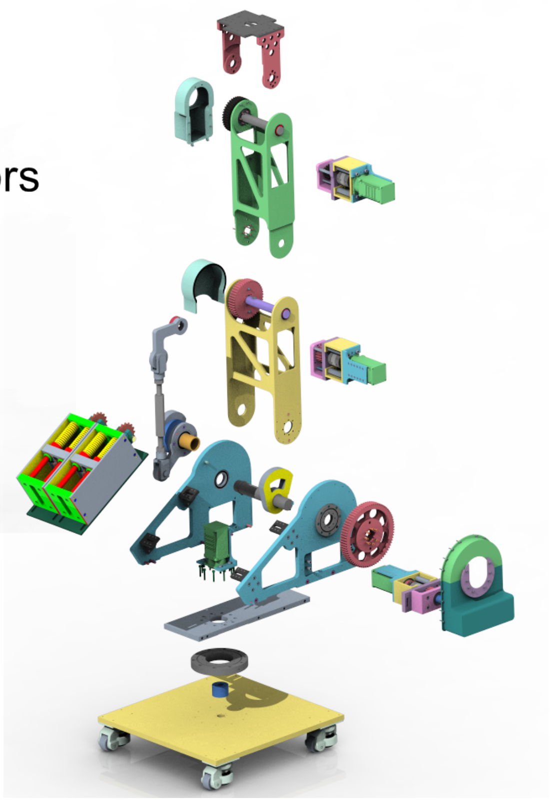
	Caulking		Pipe Assembly	Sanding
User Placement	*Ex-situ*	*In-situ*	*In-situ*	*In-situ*
Collaborative Nature	Geometric virtual fixture (VF)	Collaborative VF	Collaborative admittance	Collaborative admittance control
Evaluation metrics	Task completion time, Tool path stability, User force			

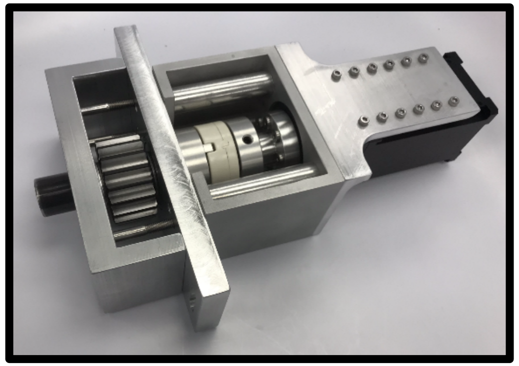
Program Themes:

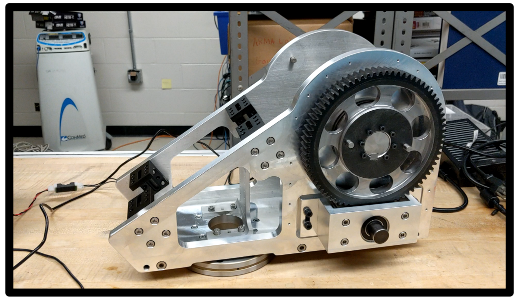
- **Societal impact** → WMD reduction
- **Scalability** → reconfigurable
- **Collaboration** → whole arm multi-point physical interaction
- **Physical embodiment** → Continuum articulated design
- **Lowering barriers to entry** → Intelligent cooperative control

Full Robot Assembly







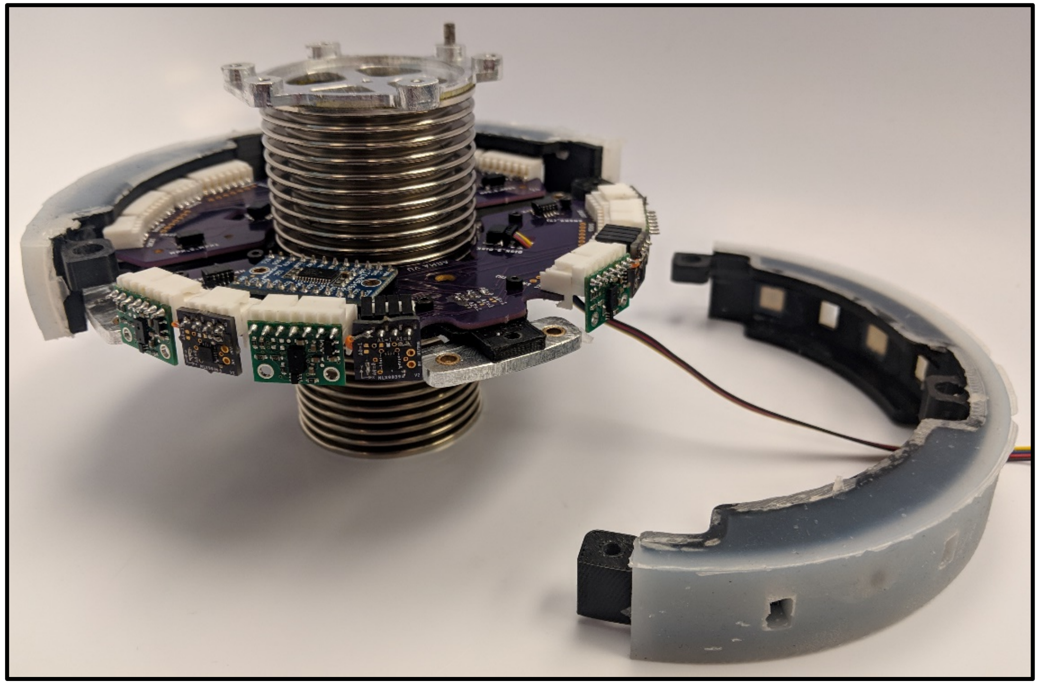


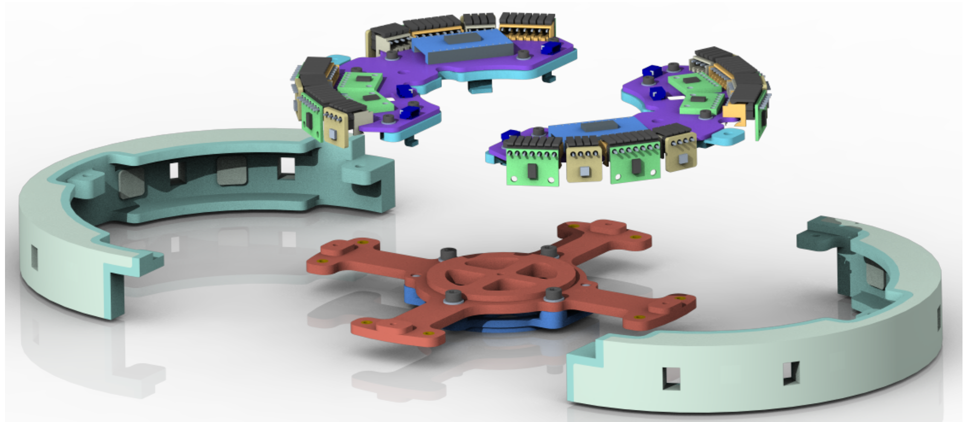
- Design for **passive** and **active** safety
- Capable of bracing
- Capable of mapping
- Reconfigurable

- Minimal actuation
- Compliant
- Sensing along its length (contact, proximity, force)

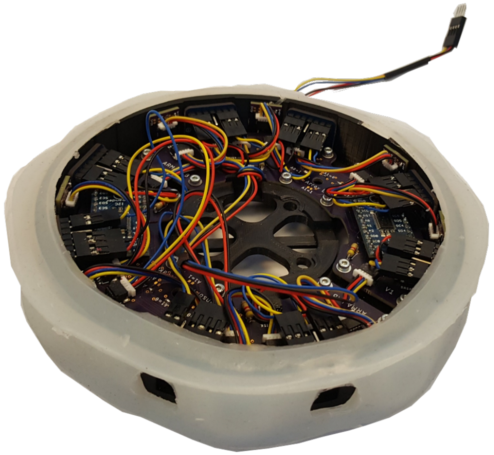
Sensory Acquisition Module

- Proximity sensing
- Contact detection
- Force sensing



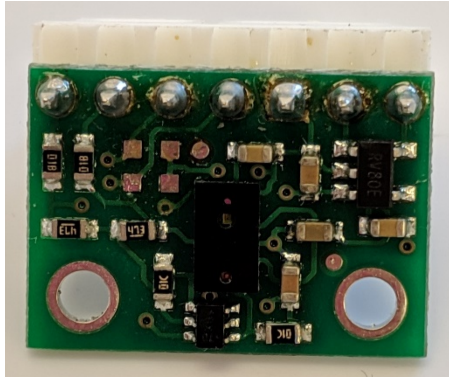


v1

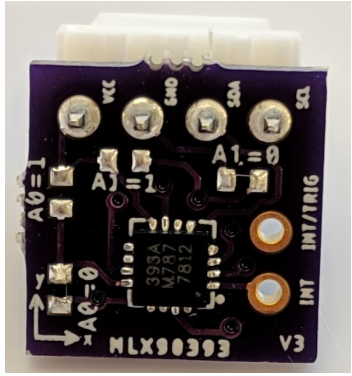


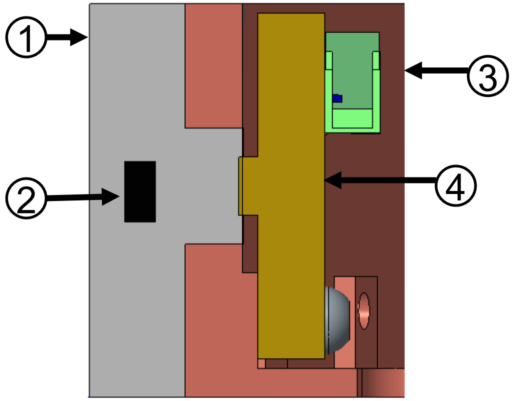
V2

Time-of-Flight Sensor



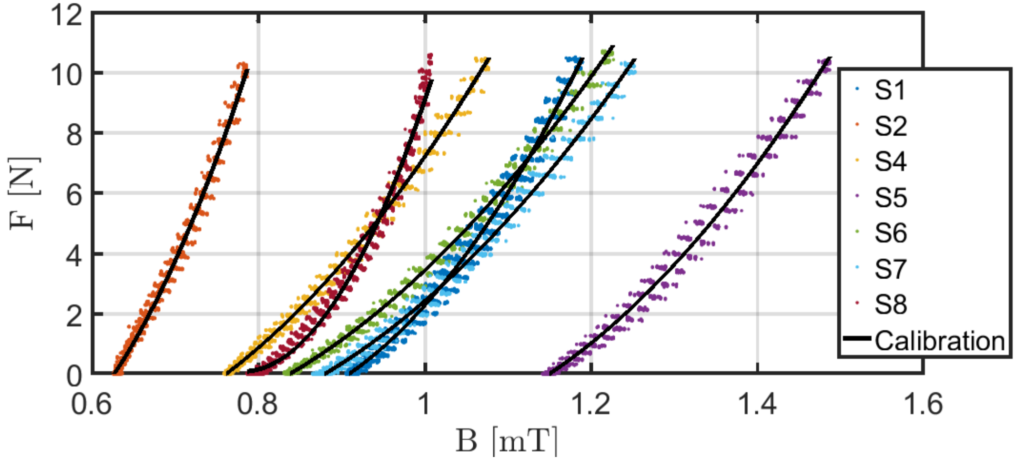
Hall-effect Contact/ Force Sensor



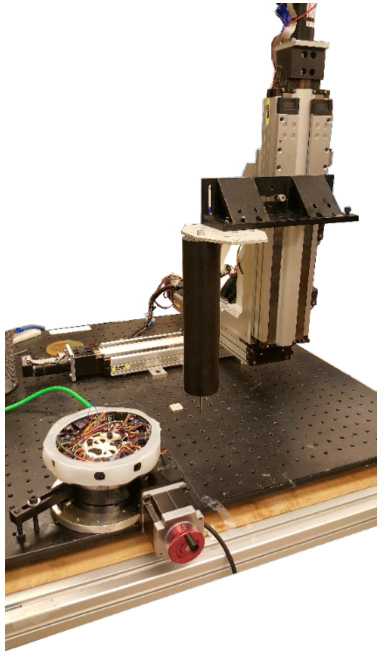


Sensory Disk Characterization

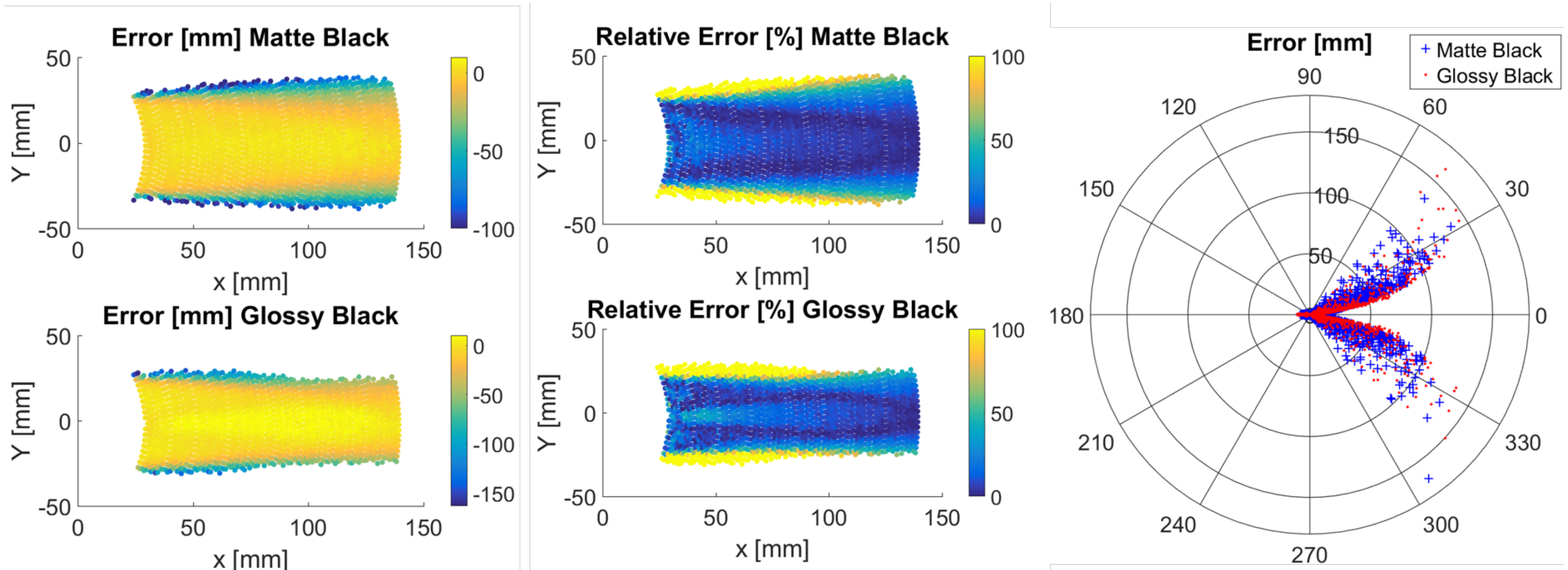
- Touch sensors were calibrated to determine relationship between applied force and magnetic sensor reading
- Time of flight sensors were characterized to determine
  - Size of the detection cone
  - Error in detection cone
  - Variation from surface reflectivity



Magnetic Touch Sensor Calibration Results



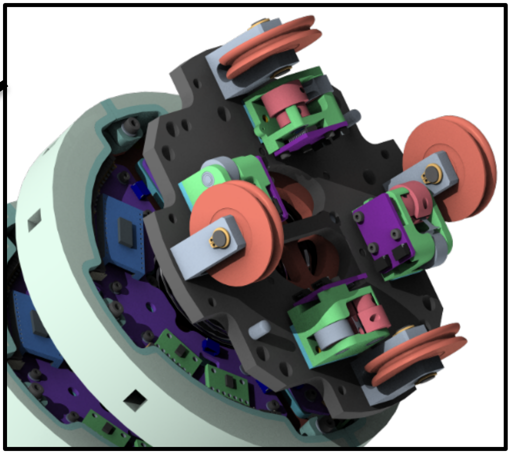
Experimental Setup

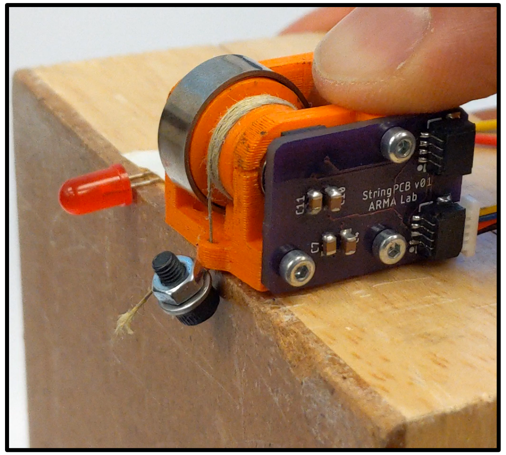


Time of Flight Sensor Characterization Results

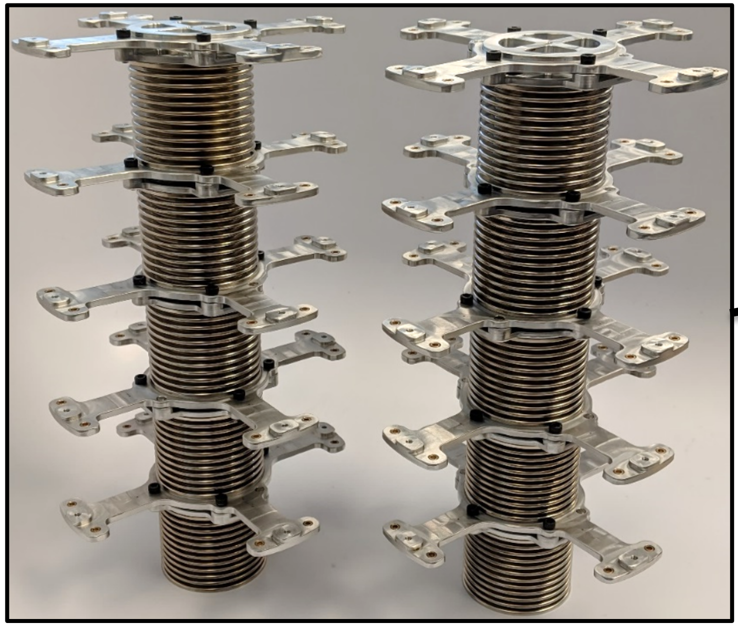
Continuum Segment Module

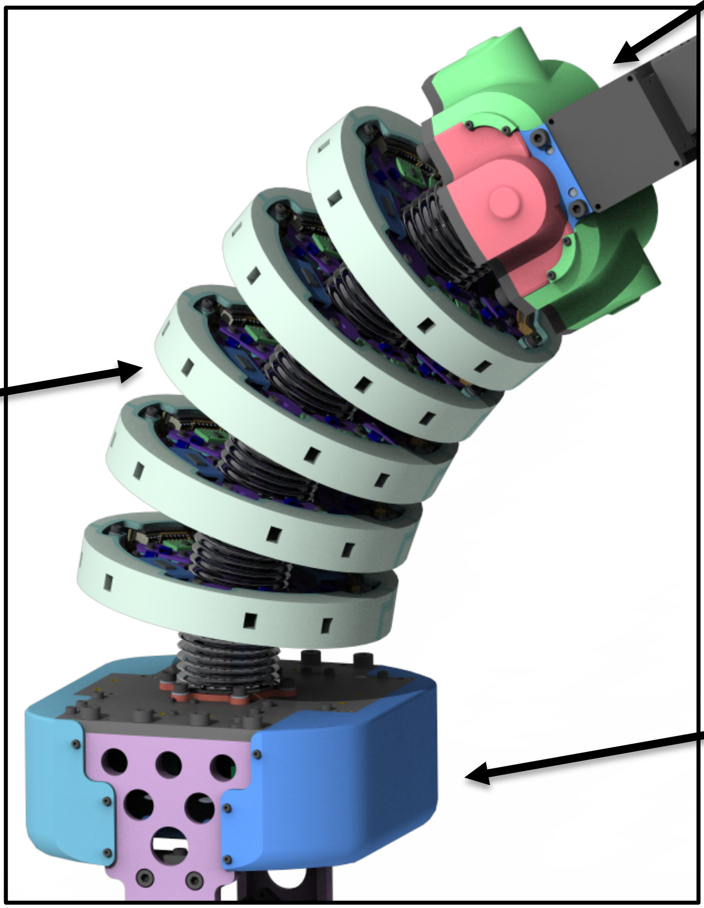
<u>Integrated string potentiometers for shape sensing</u>



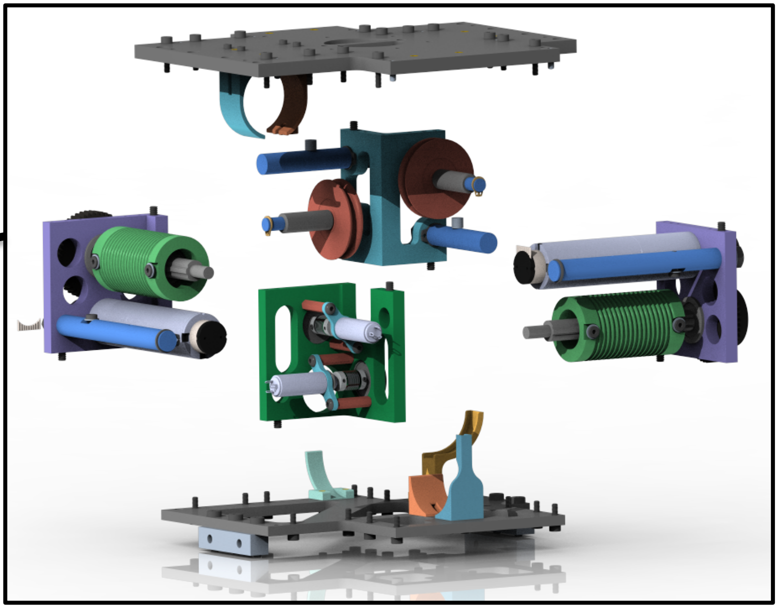


<u>Flexible bellows</u>



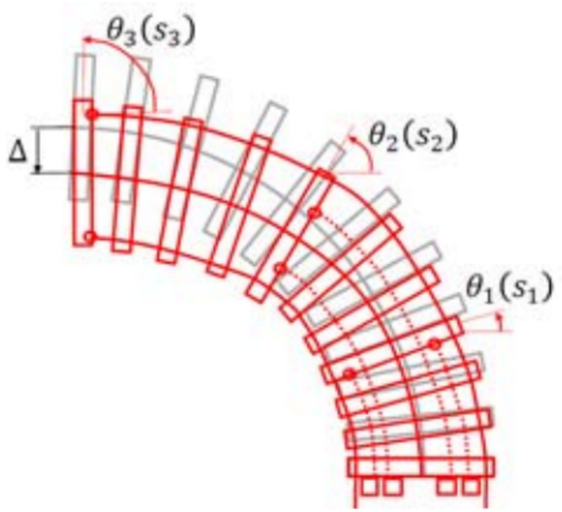


<u>Actuation unit</u>



- Goal: estimation of joint loads from deflected (sensed) shape

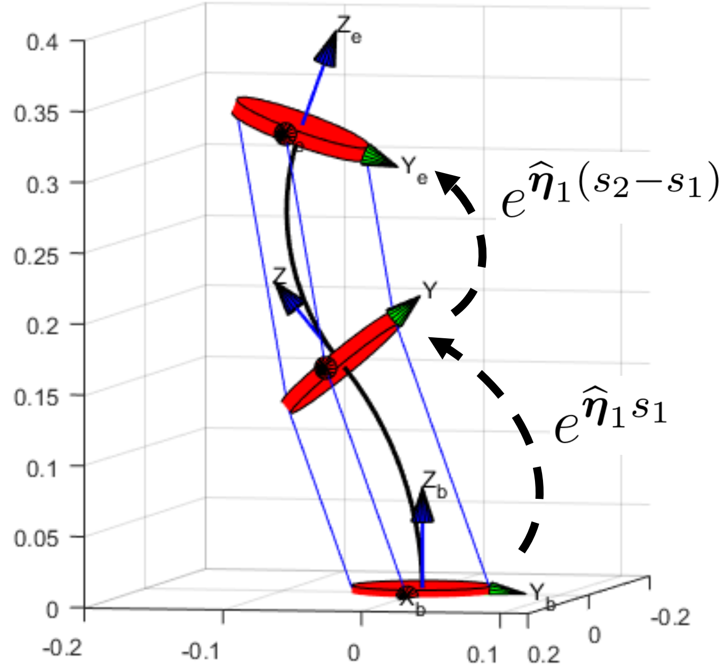
Real-Time Shape Sensing with String Potentiometers

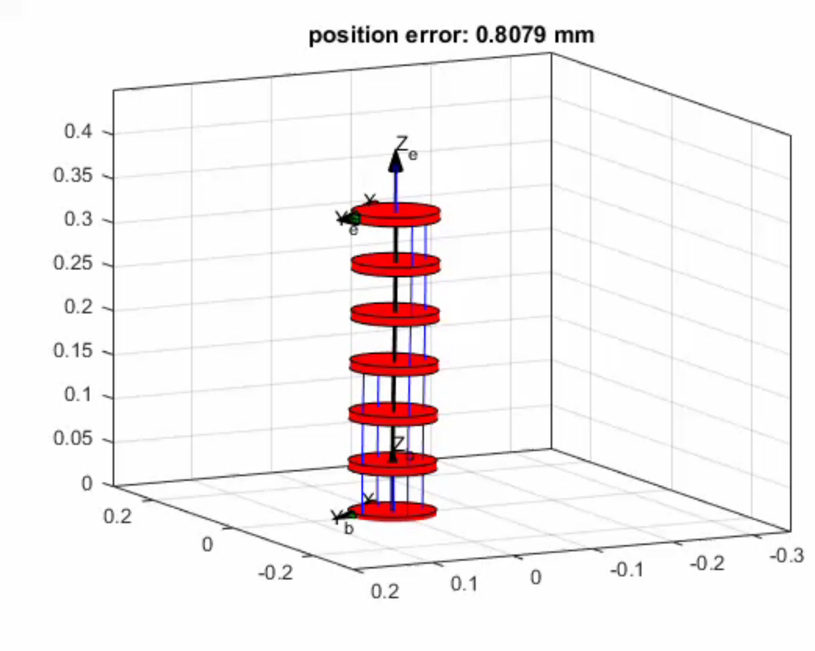


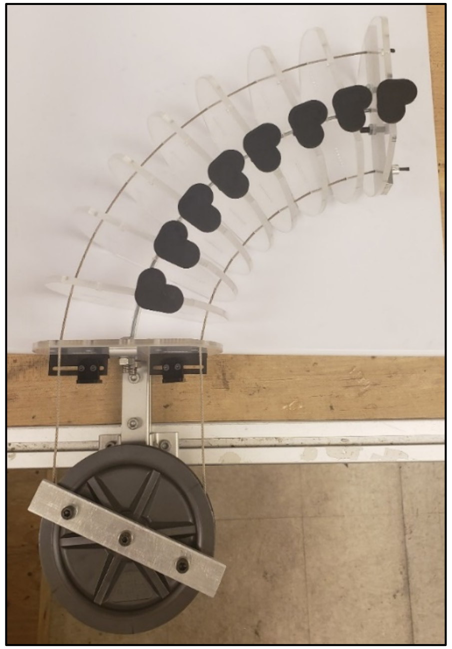
String pot lengths: $$L_i = \sum_{j=1}^{n} \|\mathbf{p}_j - \mathbf{p}_{j-1}\|$$

Solving for the shape: $$\min_{\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n} \frac{1}{2}\sum_{i=0}^{j}(L_i^* - L_i)^2$$

measured predicted

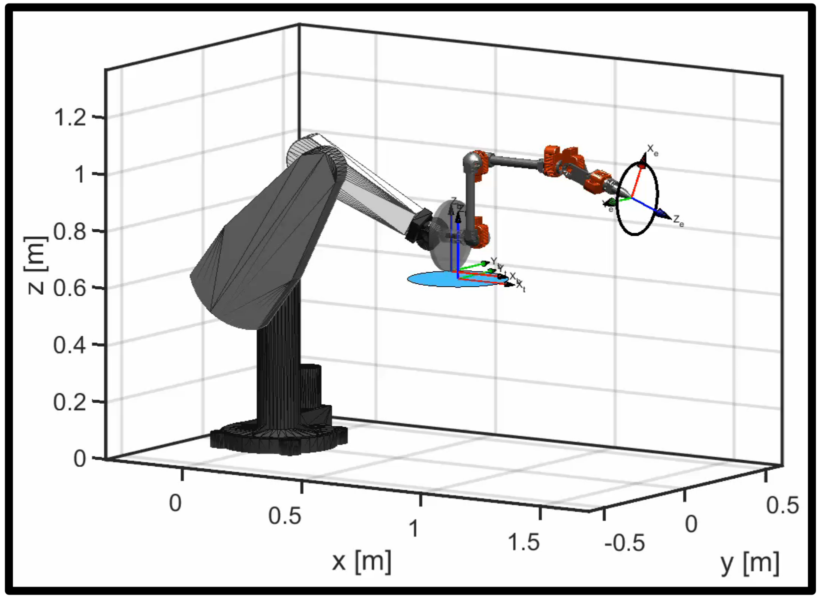




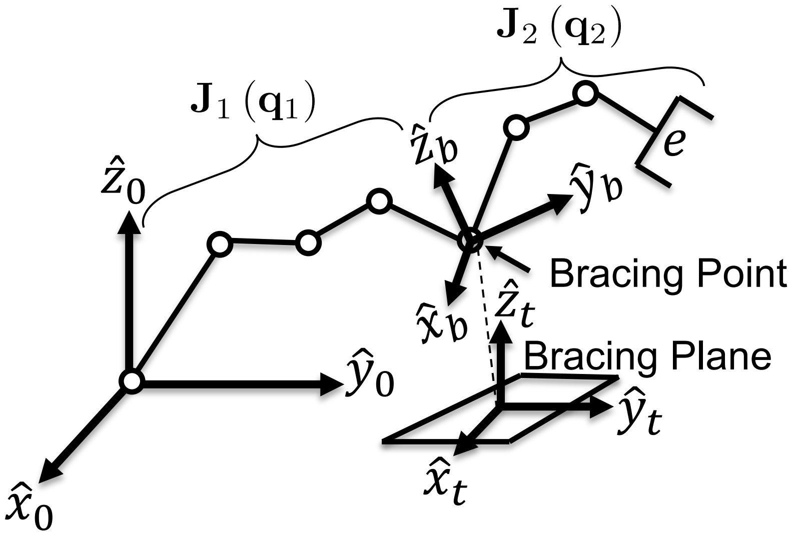


Bracing and Redundancy Resolution For Underpowered Robots

- **Kinematic and compliance modeling of braced manipulators**
- **Redundancy resolution strategy:** Use gradient projection to
  - Maintain bracing constraints
  - Minimize compliance in a task dependent direction
  - Keep robot from falling off bracing plane
  - Maximize kinematic isotropy



Video Demonstrating Redundancy Resolution Strategy Simulation

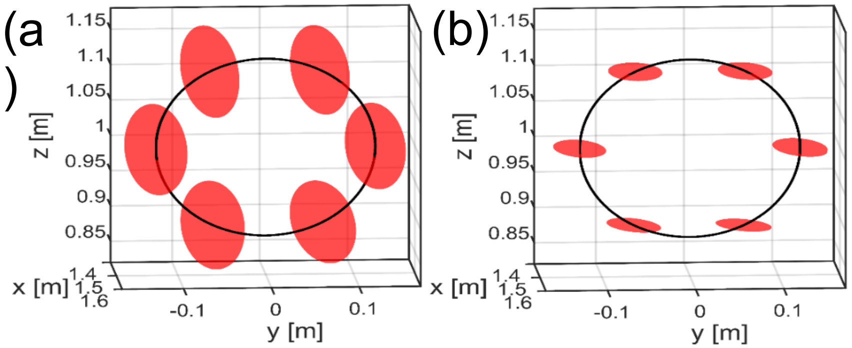


$$\begin{bmatrix} \dot{\tilde{\mathbf{b}}} \\ \dot{\mathbf{q}}_2 \end{bmatrix} = (\mathbf{A}^+)\,\Delta^0\mathbf{x}_e + (\mathbf{I} - \mathbf{A}^+\mathbf{A})\,\alpha\nabla g$$

$$g = \alpha_1 k + \alpha_2 C_i + \alpha_3 \theta_z + \alpha_4 d$$

$$\Delta^0\mathbf{x}_e = \underbrace{\begin{bmatrix} \mathbf{S}_1\mathbf{H} & \mathbf{S}_2\mathbf{J}_2 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \dot{\tilde{\mathbf{b}}} \\ \dot{\mathbf{q}}_2 \end{bmatrix}$$

$$\dot{\mathbf{q}}_1 = (\mathbf{J}_1^{-1})\,\mathbf{H}\dot{\tilde{\mathbf{b}}}$$

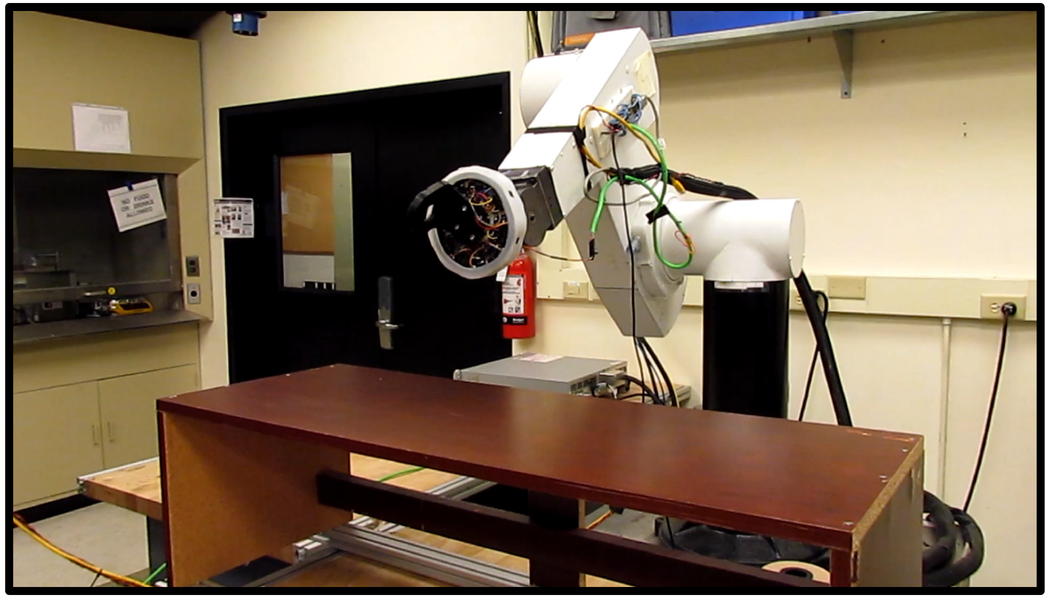


Compliance Ellipsoids Comparing: (a) Free space Motion to (b) Bracing

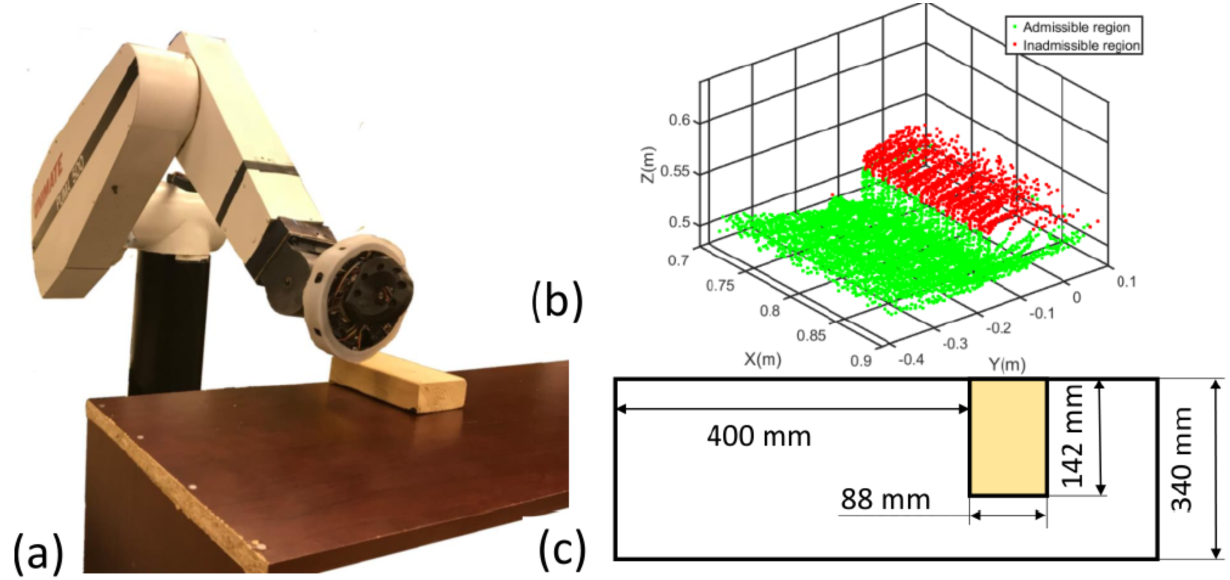
[1] G.L.H. Johnston, A.L. Orekhov, and N. Simaan. "Kinematic Modeling and Compliance Modulation of Redundant Manipulators Under Bracing Constraints." 2020 *IEEE International Conference on Robotics and Automation.* [Accepted Jan. 2020].

Using Sensory Disk for Bracing

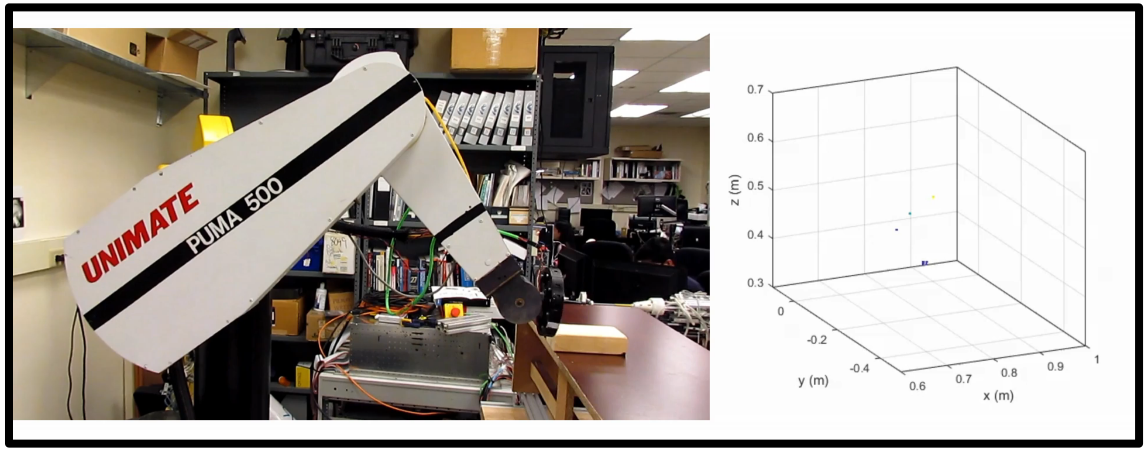
- The sensor disk can be used to identify regions in the environment that are acceptable for bracing
- Completed experiments on mapping environment and extracting acceptable and unacceptable bracing regions



Using sensory disk to identify bracing plane



(a) Robot with attached sensor disk (b) Mapping experiment result (c) Ground truth

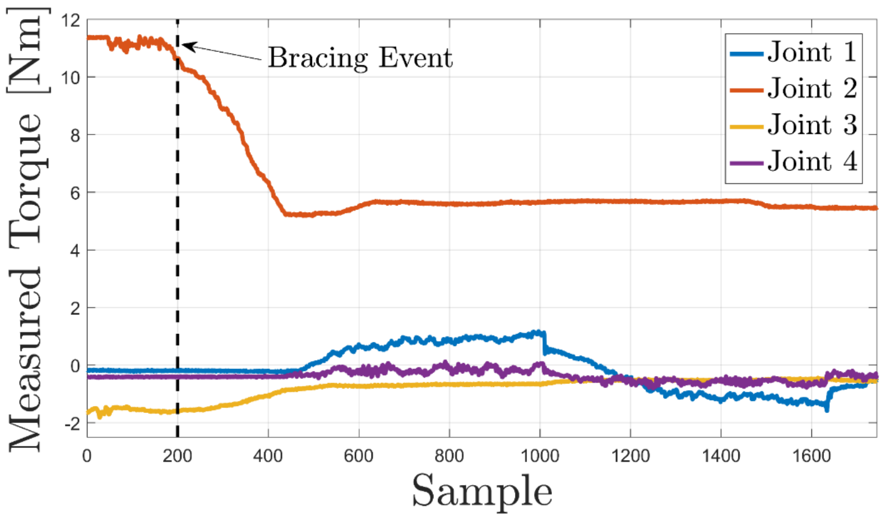


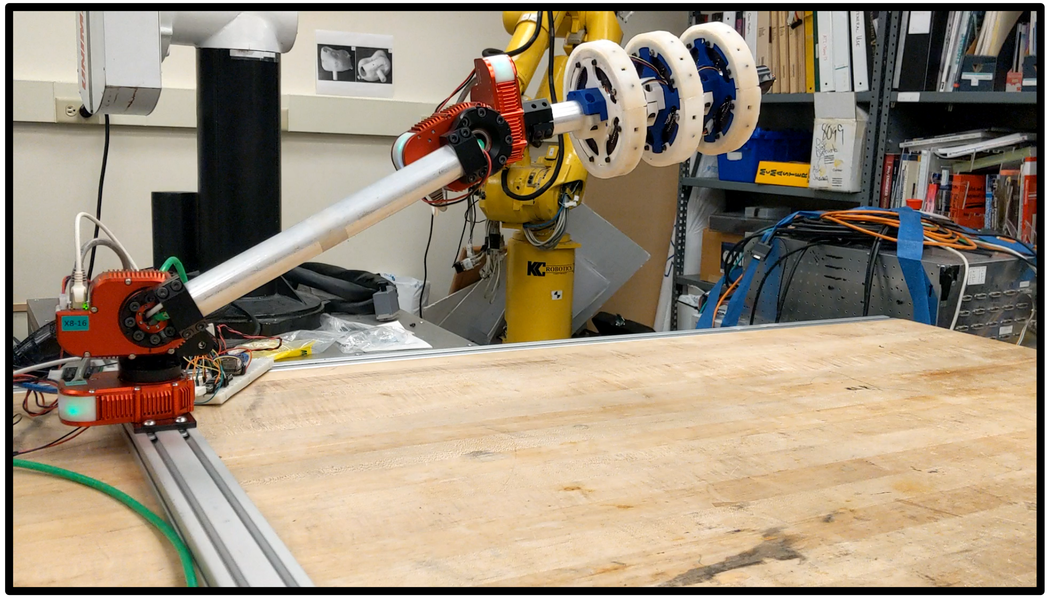
Using sensory disk to map the environment

[2] C. Abah, A.L. Orekhov, G.L.H. Johnston, P. Yin, H. Choset, and N. Simaan, "A Multi-modal Sensor Array for Safe Human-Robot Interaction and Mapping." *IEEE International Conference on Robotics and Automation (ICRA)*. May 2019.

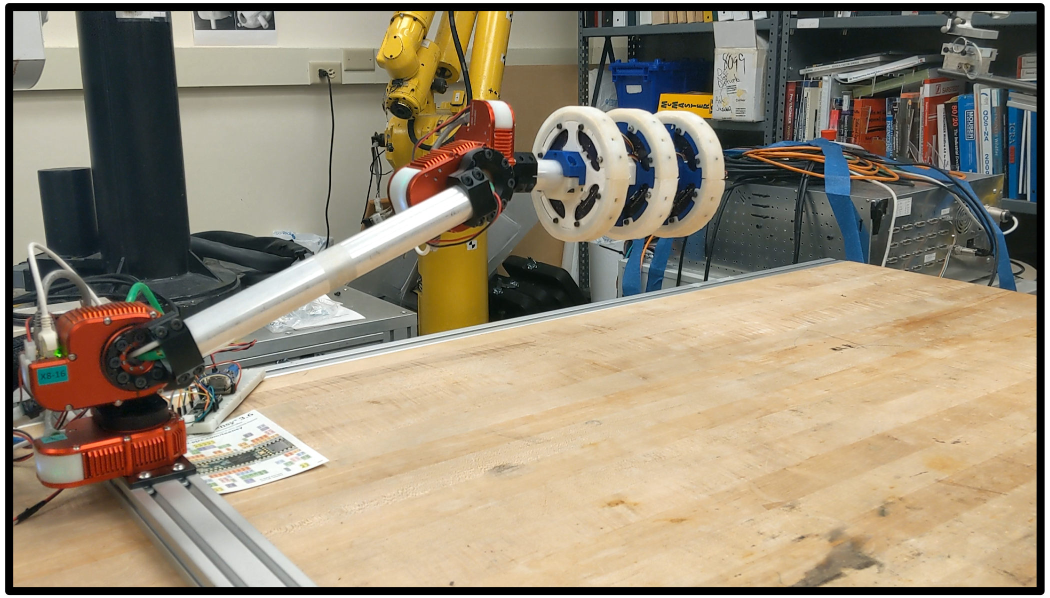
pHRI with Sensory Disk

- The sensory disk can be used to control the robot using direct contact or via the time of flight sensors
- Adds additional sensing for admittance control and safety awareness





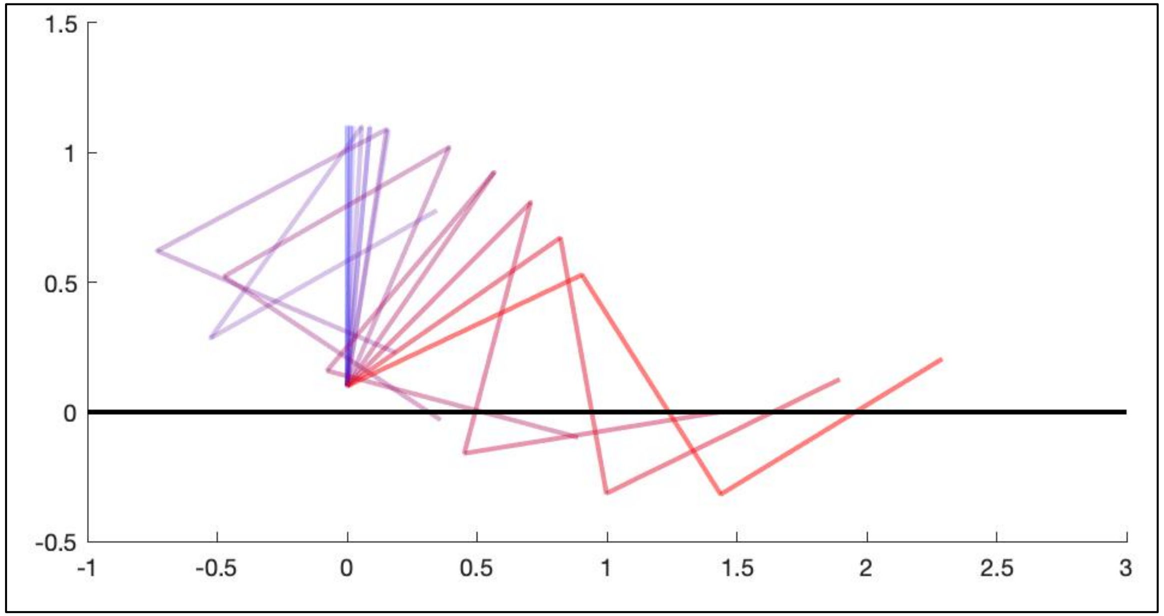
Using touch sensors

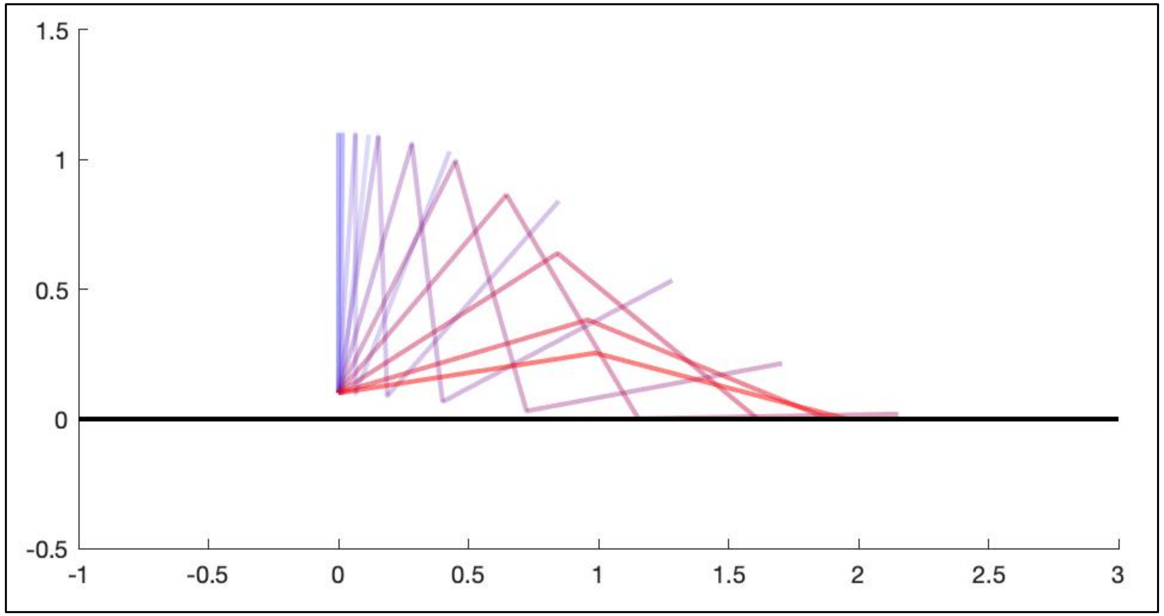


Using ToF sensors while bracing

[3] A. L. Orekhov, G.L.H. Johnston, C. Abah, H. Choset, and N. Simaan, "Towards Collaborative Robots with Sensory Awareness: Preliminary Results Using Multi-Modal Sensing," in *ICRA 2019 workshop "Physical human-robot interaction: a design focus,"* May 2019

Solve a trajectory optimization problem





<u>Given:</u>

- Trajectory of configurations and velocities
- Constraints
  - e.g., arm must stay in upper half-plane
  - e.g., satisfy a torque limit

<u>Goal:</u> Minimize cost of the trajectory while satisfying the constraints

<u>Note:</u> Robot parameters and (possibly discontinuous!) dynamics
<u>Need</u>: Cost function (typically penalizes final distance to a goal, control action, etc.)

Formal Problem Statement

Given:

- An initial robot trajectory x and its associated control actions u
  - $x = (x_1, x_2, \ldots, x_T)$, $u = (u_1, u_2, \ldots, u_T)$
  - Should look mostly reasonable, but may violate constraints
- Constraints h_j, for $j = 1, 2, \ldots, n_{constraints}$
  - We require that $h_j(x_i, u_i) < 0$ for all $i = 1, 2, \ldots, T$

With:

- The (possibly discontinuous) robot dynamics function f
  - For any starting state x_i and any control input u_i, the resulting state is $x_{i+1} = f(x_i, u_i)$
- A cost function L (for now, we provide the cost function)
  - $L(x, u) = L_T(x_T) + \sum_{i=1}^{T-1} L_i(x_i, u_i)$

Goal:

- Find x and u to (locally) minimize L, while satisfying the dynamics and constraints

Why This Problem Is Hard

- Contact and interaction with obstacles
  - Standard obstacle avoidance methods don’t apply
- Discontinuous dynamics
  - Many optimization methods (think gradient descent!) assume a continuous (or worse, differentiable) dynamics function
- Exponentially many possible bracing point combinations

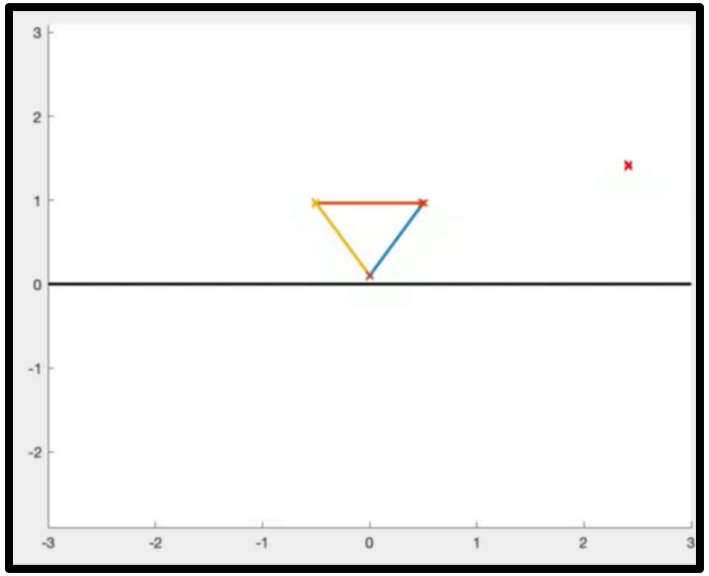
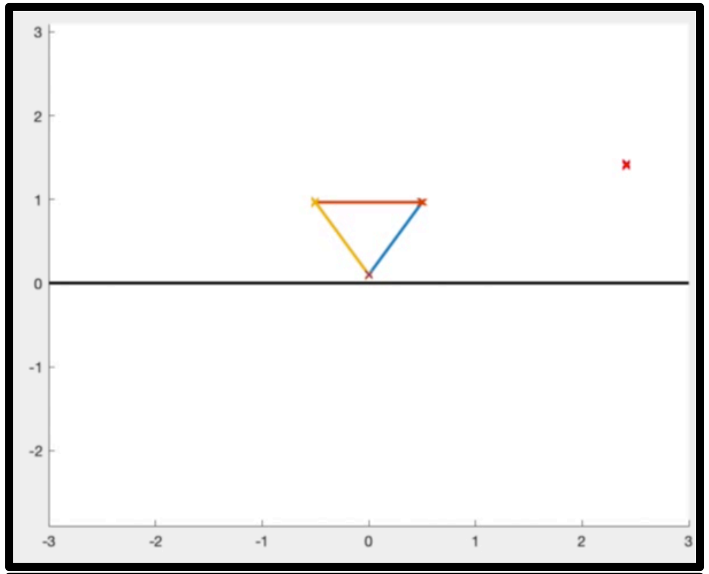
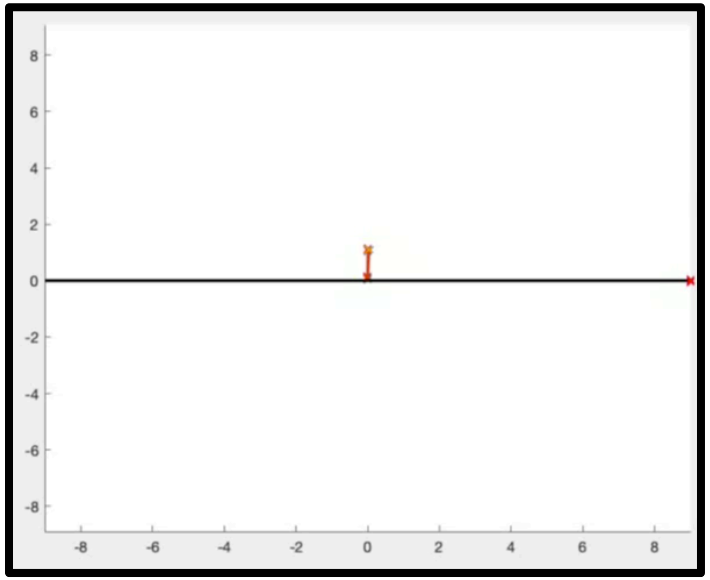
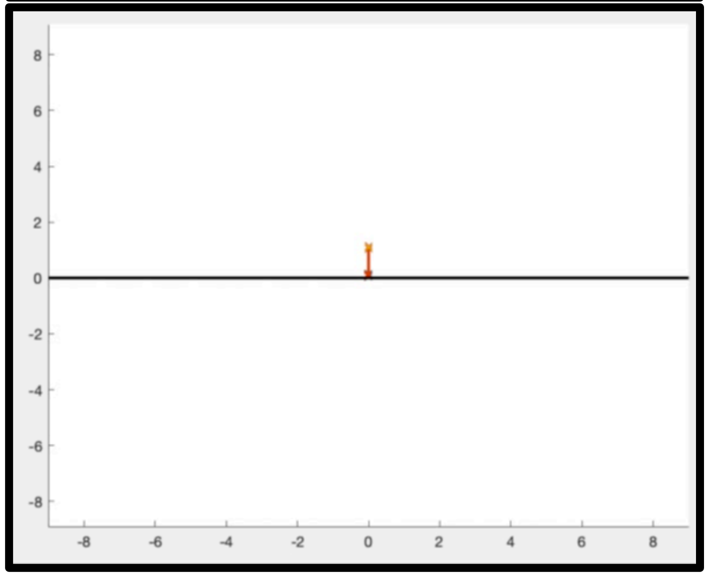
Related Work

- Penalties for constraint violations
  - Numerical instability as the penalties get large
- Augmented Lagrangian method
  - Uses constraint penalties and an iterative estimate of the Lagrange multipliers
  - Penalties don't get as large, so this improves stability
  - Gradient information may make it ignorant to discontinuous dynamics
- ILQG framework allows for bounds on the control
  - But no way to deal with obstacles or their associated dynamics
- Walking and rock-climbing robots can plan foothold locations
  - Independent of previous foothold history, which we can't assume

Constraint-Aware ILQR

- For a fixed set of constraints, we can compute Lagrange multipliers and the optimal control policy to exactly satisfy ("activate") those constraints:
  - $(g_i + G_i \delta x_i) + H_i \delta u_i + \boldsymbol{C_u \lambda} = 0$ if the change in control δu_i is a stationary point
    - First two terms compute change in cost-to-go from applying δu_i, as in standard ILQR.
    - Bold term is new. C_u is the matrix of partial derivatives of the active constraints with respect to δu_i, and λ is the vector of Lagrange multipliers.
  - $0 = h_j(x_i + \delta x_i, u_i + \delta u_i) = h_j(x_i, u_i) + \frac{\partial h_j}{\partial x_i} \delta x_i + \frac{\partial h_j}{\partial u_i} \delta u_i$ for any (linearized) active constraint h_j
  - Create and solve a system of linear equations; solve for λ by eliminating each $\frac{\partial h_j}{\partial u_i} \delta u_i$ term
- In the backward step, we use an estimate of active constraints to compute optimal control and cost-to-go
- In the forward rollout, we compute the optimal control for each active constraint set and select the cheapest allowed control

Some Results

Initial	Optimized
	
	

Top row:
Arm starts folded into a triangle, and braces with the first joint to reach the target. The target is barely reachable if bracing with the first link

- 35 iterations
- 75 seconds
- Stabilizes in ~5 iterations
- Can replan mid-execution

Bottom row:
9 link arm extending horizontally

- Not fully optimized
- Each iteration takes at least a minute depending on how many active constraint sequence guesses are used

Uncertainty-based planning