CONSTRAINED OPTIMAL CONTROL OF A SPHERICAL PARALLEL MANIPULATOR

AVIRAM YANOVER

DANIEL CHOUKROUN

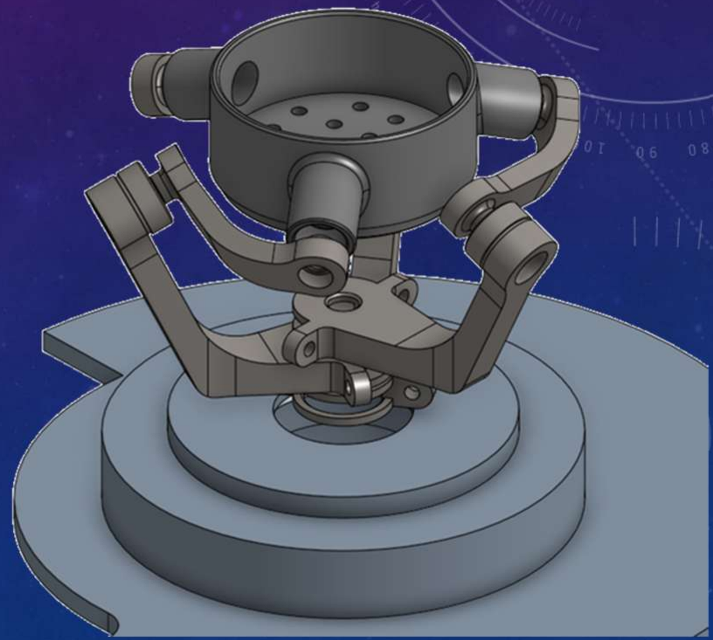
BEN GURION UNIVERSITY OF THE NEGEV

MECHANICAL ENGINEERING DEPARTMENT

2025 IAAC Control Conference (IAAC3), Apr 28th, 2025, Hertzelia

OUTLINE

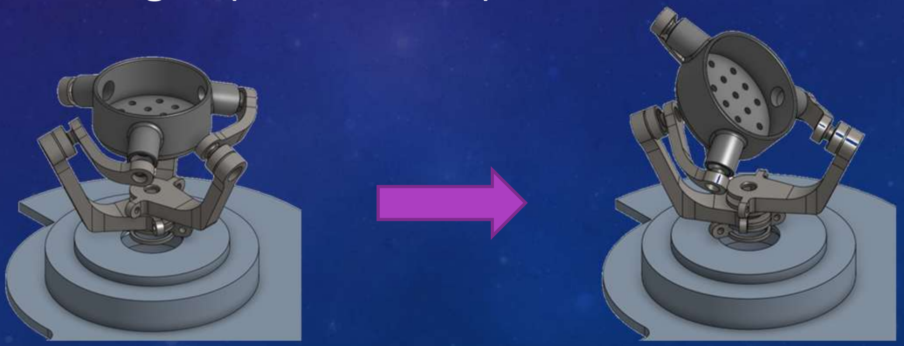
- Introduction
- Plant model
- Controller designs & Rest-to-Rest Simulation Results
- Target Tracking
- Discussion



INTRODUCTION

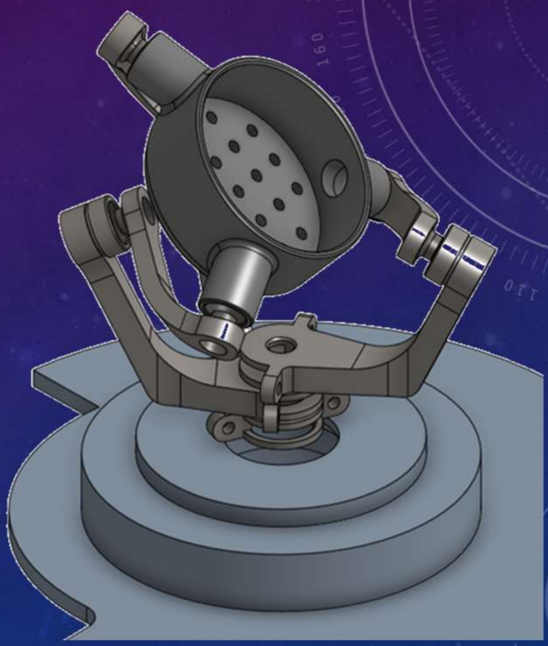
INTRODUCTION

- A mechanism for rotating a platform around a fixed center
- The links and platform form a kinematic loop
- Belongs to the family of parallel robots
- We wish to find the best sequence of control inputs that rotates the platform from any initial position to point at a desired line of sight (rest-to-rest).



INTRODUCTION (CONT.)

- Trajectories are expected to be short
- Limited on-board computing power
- Perturbations in sensing and actuation
- Loss of controllability due to singularities
- Simple feedback-based control is fast and geometrically intuitive but not always safe.
- Model-Predictive Control is computationally intensive



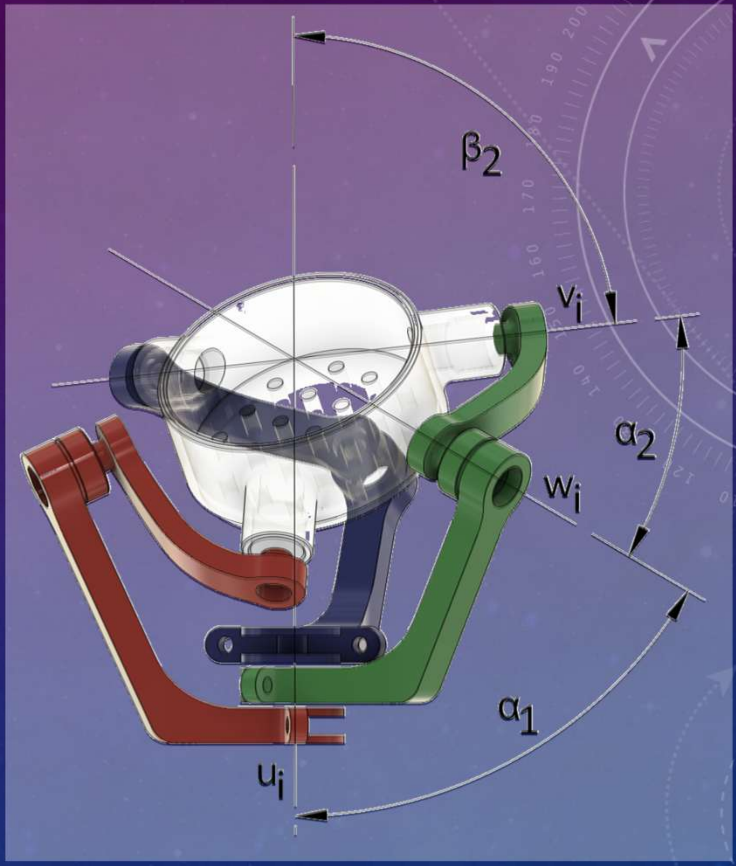
THE PLANT MODEL

THE PLANT

- Initial inverse kinematics
- Forward kinematics
- Singularity and collision monitoring

$$\dot{Q} = [\boldsymbol{\omega} \times]Q$$
$$\dot{\boldsymbol{\theta}} = \boldsymbol{\gamma}$$
$$\boldsymbol{\omega} = J_{DK}(Q, \boldsymbol{\theta})\, \boldsymbol{\gamma}$$
$$J_{DK}(Q, \boldsymbol{\theta}) = -A^{-1}B$$

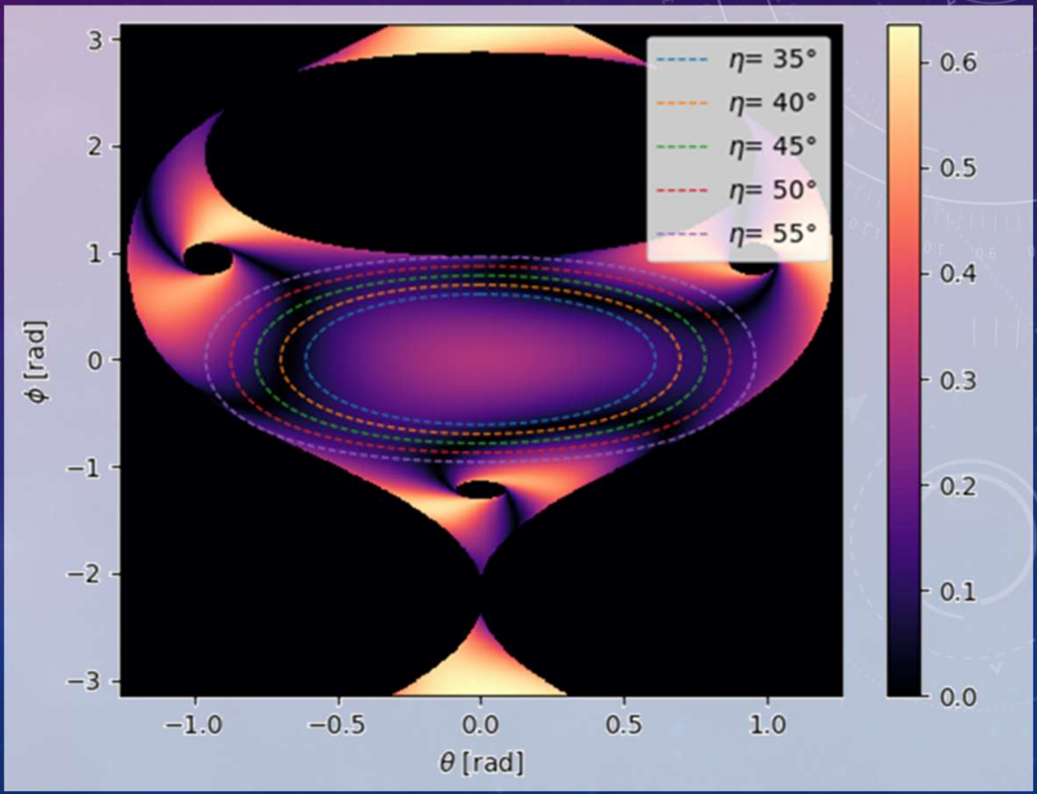
Q – Platform's Rotation Operator
$\boldsymbol{\theta}$ - Joint Angles
$\boldsymbol{\omega}$ – Platform's Angular Velocity
γ – Command Joint Rates



$$A = \begin{bmatrix} (\boldsymbol{w}_1 \times \boldsymbol{v}_1)^T \\ (\boldsymbol{w}_2 \times \boldsymbol{v}_2)^T \\ (\boldsymbol{w}_3 \times \boldsymbol{v}_3)^T \end{bmatrix}$$
$$B = diag(\boldsymbol{w}_1 \times \boldsymbol{u}_1 \cdot \boldsymbol{v}_1, \boldsymbol{w}_2 \times \boldsymbol{u}_2 \cdot \boldsymbol{v}_2, \boldsymbol{w}_3 \times \boldsymbol{u}_3 \cdot \boldsymbol{v}_3)$$

2ND KIND SINGULARITY VS CO-ELEVATION CONTOURS

- Det(A) as a function of ϕ and θ
- Singularity zones in black color
- Contours correspond to different elevation angles



CONTROLLER DESIGN

QUATERNION FEEDBACK CONTROL

- Inspired by satellite attitude control
- Quaternion kinematics

$$\dot{\boldsymbol{q}} = -\frac{1}{2}\begin{bmatrix} [\boldsymbol{e}\times] + qI_3 \\ -\boldsymbol{e}^T \end{bmatrix}\boldsymbol{\omega}$$

$$\boldsymbol{\omega} = J_{\boldsymbol{DK}}(Q, \boldsymbol{\theta})\boldsymbol{\gamma}$$

- Command joint rates

$$\boldsymbol{\gamma} = -\mathrm{K}(B^{-1}A)\boldsymbol{e}$$

- Non-linear regulator, globally convergent, Lyapunov analysis
- Geometrically intuitive, LQ optimal
- Unconstrained (UF)

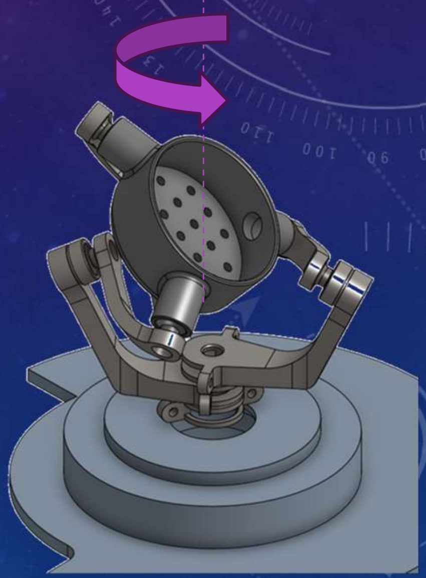
PROPOSED APPROACH

- Separation of Elevation and Azimuth control
  - Elevation control via advanced computational methods
  - Azimuth control via proportional feedback with saturation

$$\gamma(t) = \gamma_{EL}(t) + \gamma_{AZ}(t)$$

AZIMUTH CONTROLLER

- Azimuth control is designed as a proportional control with azimuth error feedback.
- Identical command rates are applied to the three joints.
- The joint rate command is limited by the joint maximum speed trimmed with the rates allocated to the elevation control.



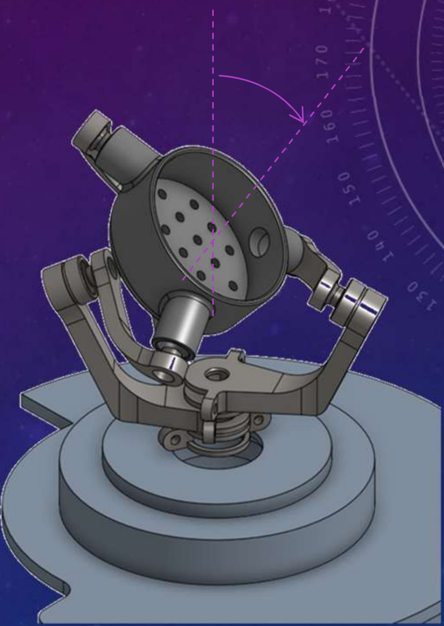
ELEVATION CONTROLLER

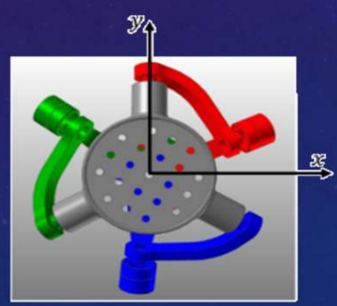
- Reinforcement Learning Approach
- Grid Search Approach

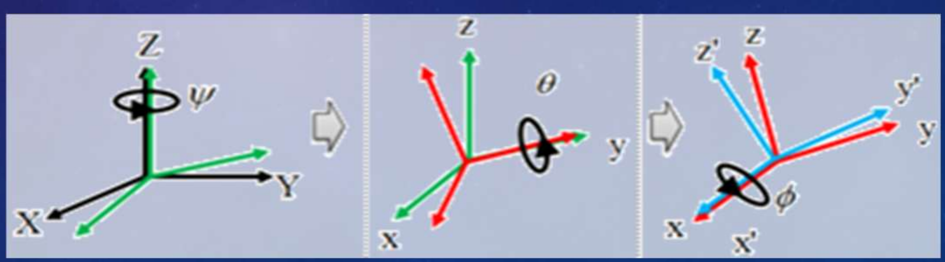
REINFORCEMENT LEARNING METHOD

ELEVATION DRL CONTROL

- Euler angles, 3-2-1 sequence, Base to Platform
- Two methodologies:
  - A2C (Discrete action space)
  - TD3 (Continuous action space)
- Stable-Baselines3* package in Python



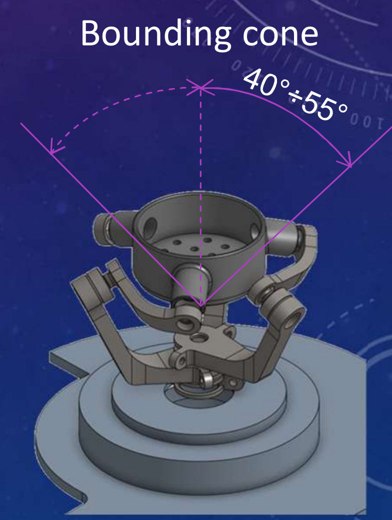




* https://stable-baselines3.readthedocs.io/en/master/#

ELEVATION DRL CONTROL

- The action space for A2C encompasses a finite number of discrete actions.
- For TD3, the action space is continuous, allowing for any joint rate within the range of −400 to 400 degrees per second.
- Episodes initiate from a randomly determined state within a predefined bounding cone.
- Episodes conclude in one of three conditions:
  - The platform aligns with the desired line of sight within an error tolerance,
  - The platform exceeds a singularity threshold of 0.05,
  - The episode surpasses 400 steps.



ELEVATION DRL CONTROL - THE REWARD FUNCTION

- The reward function is formulated as

$$R(s_t, a_t, s_{t+1}) = T(s_{t+1}) + F(s_t, a_t, s_{t+1})$$

- $T(s_{t+1})$ is the terminal outcome:

$$T(s_{t+1}) = \begin{cases} 50, if\ target\ reached \\ -70, if\ singularity\ thereshold\ exceeded \end{cases}$$

- $F(s_t, a_t, s_{t+1})$ is an immediate outcome:

$$F(s_t, a_t, s_{t+1}) = 10(\Upsilon_{A_{t+1}} - \Upsilon_{A_t}) + 30(\eta_t - \eta_{t+1}) - 0.2$$

Υ_{A_t} denote the singularity index $\det(A_t)$ at step t
η_t denote the elevation error at step t

TRAINING

Alg.	**A2C**	**TD3**
Action Space	Discrete	Continuous
N	400	
$\dot{\theta}_s$	$100, 200, 400$	[-400, 400]
τ	0.05	
ϵ	$0.2°$	
ϑ_{max}	$40°, 55°$	
α	$7e-4$	
γ	0.99	
α_1	$65°$	
α_2	$60°$	
β_1	$0°$	
β_2	$110°$	

Max. no of steps

Angular velocity [°/sec]

Singularity threshold

Tolerance for reaching the target

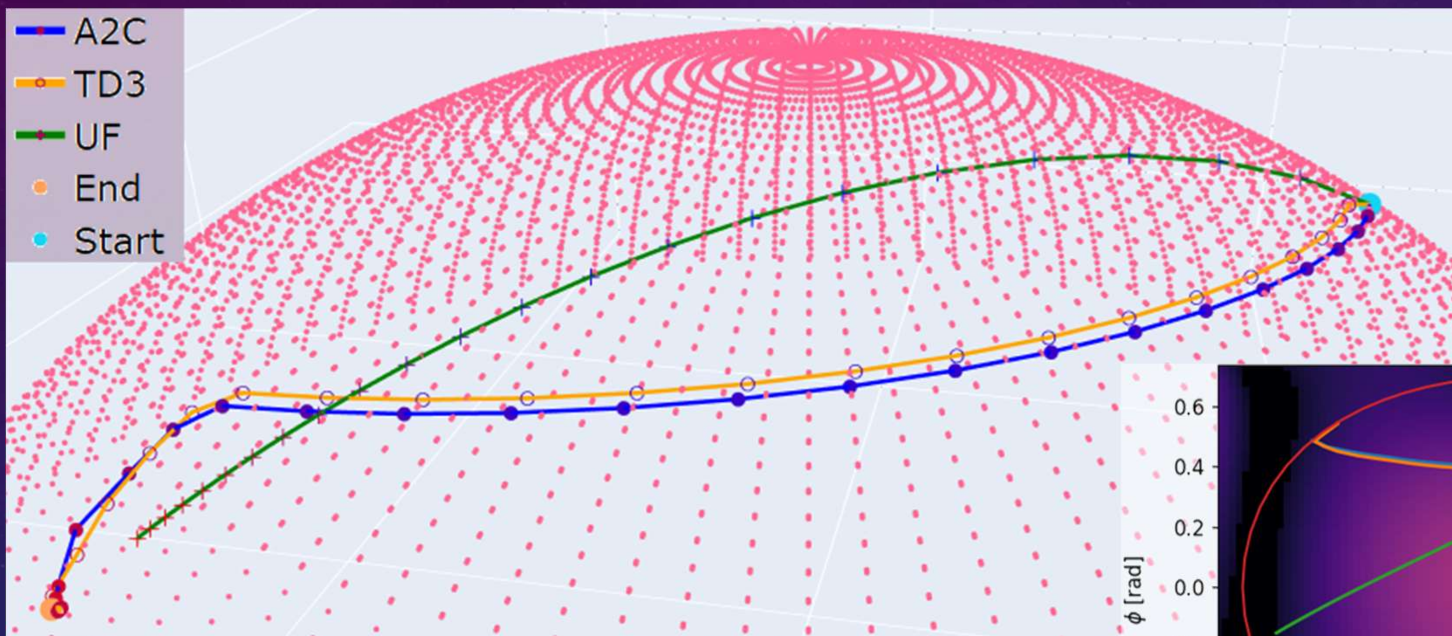
Bounding cone angle

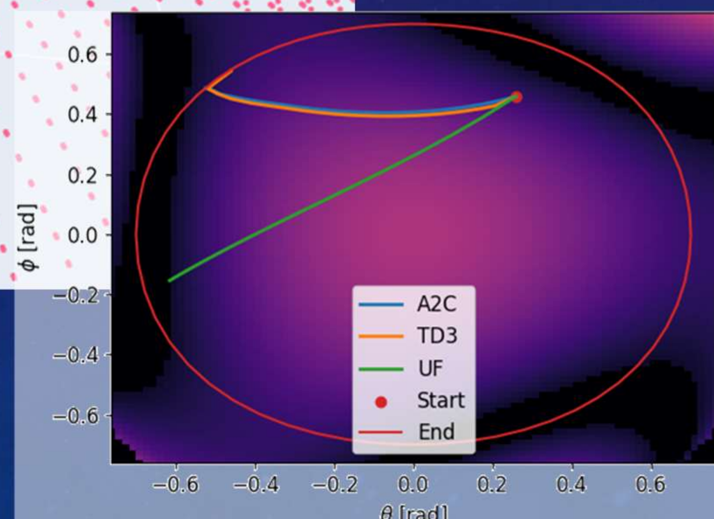
Learning rate

Discount factor

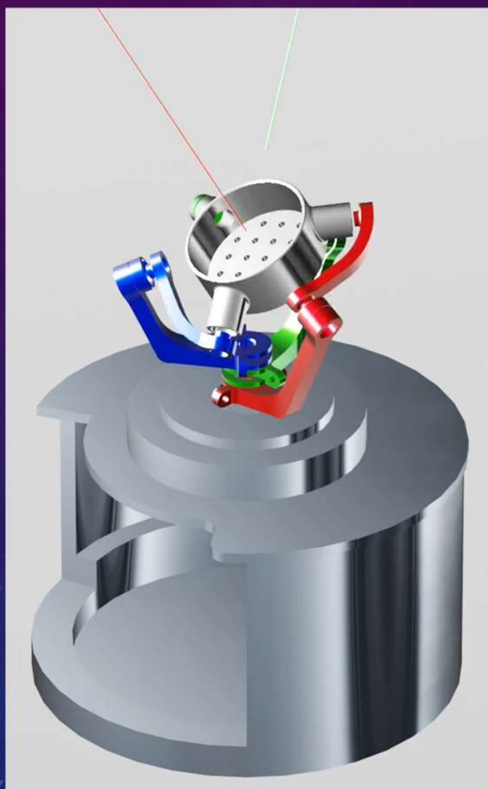
SPM geometry

RL TRAJECTORIES

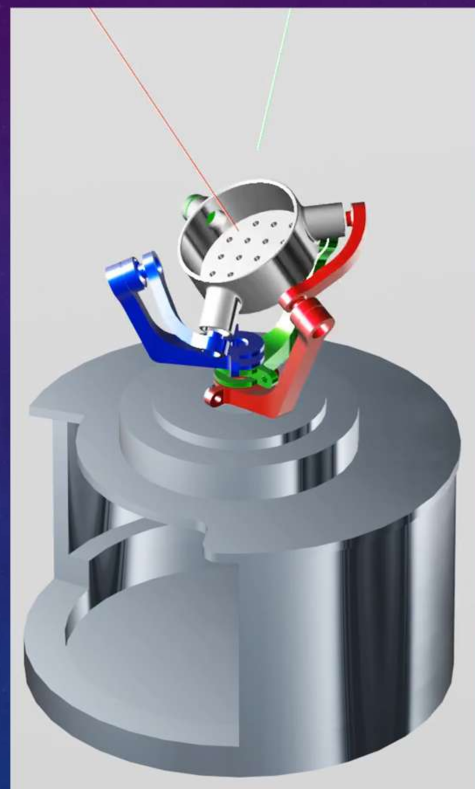




ANIMATION



A2C

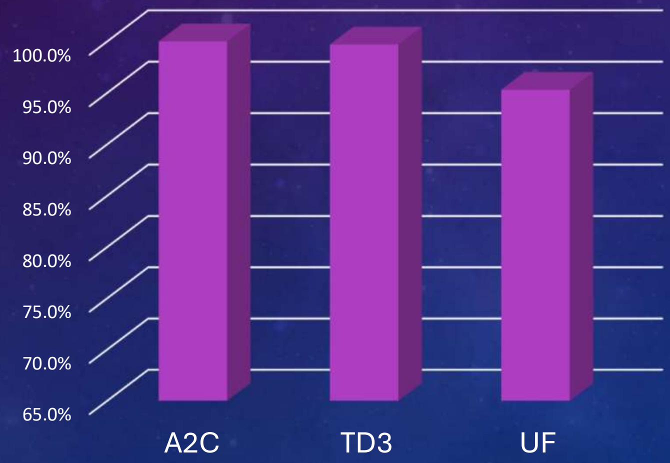


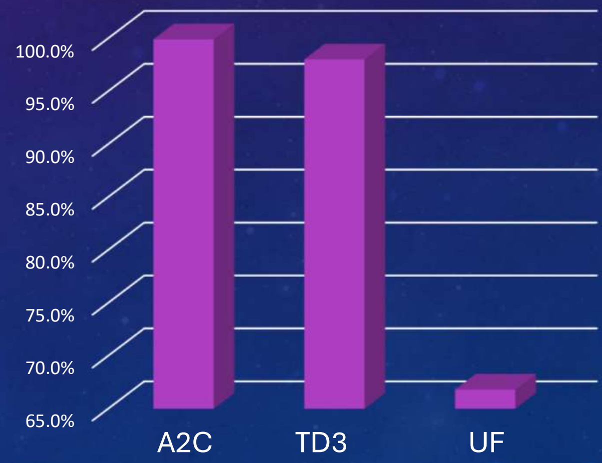
TD3

UF

DRL SIMULATION RESULTS SENSITIVITY TO LOS BOUNDARIES

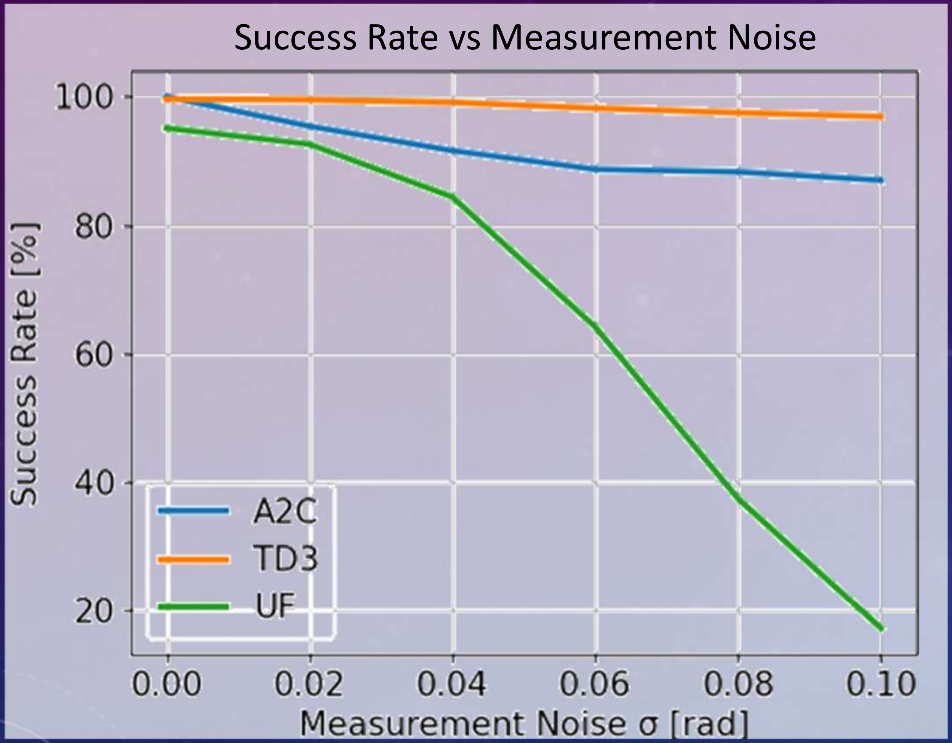
	A2C	TD3	UF	A2C	TD3	UF
Source El. Range	(0° - 40°)			(0° - 40°)		
Target El. Range	(0° - 40°)			(35° - 40°)		
Success rate	100%	99.7%	95.3%	99.9%	98%	66.8%
Average Arc Length*	50°	47°	33°	76°	66°	42°

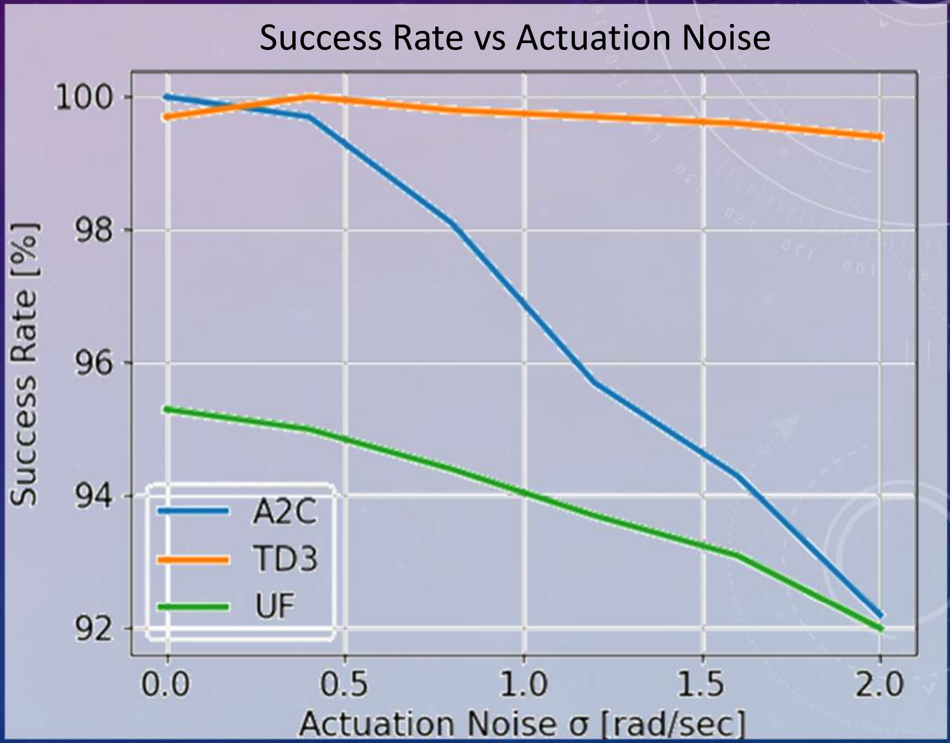




* Successful episodes only

SENSITIVITY TO NOISES (0°-40°)





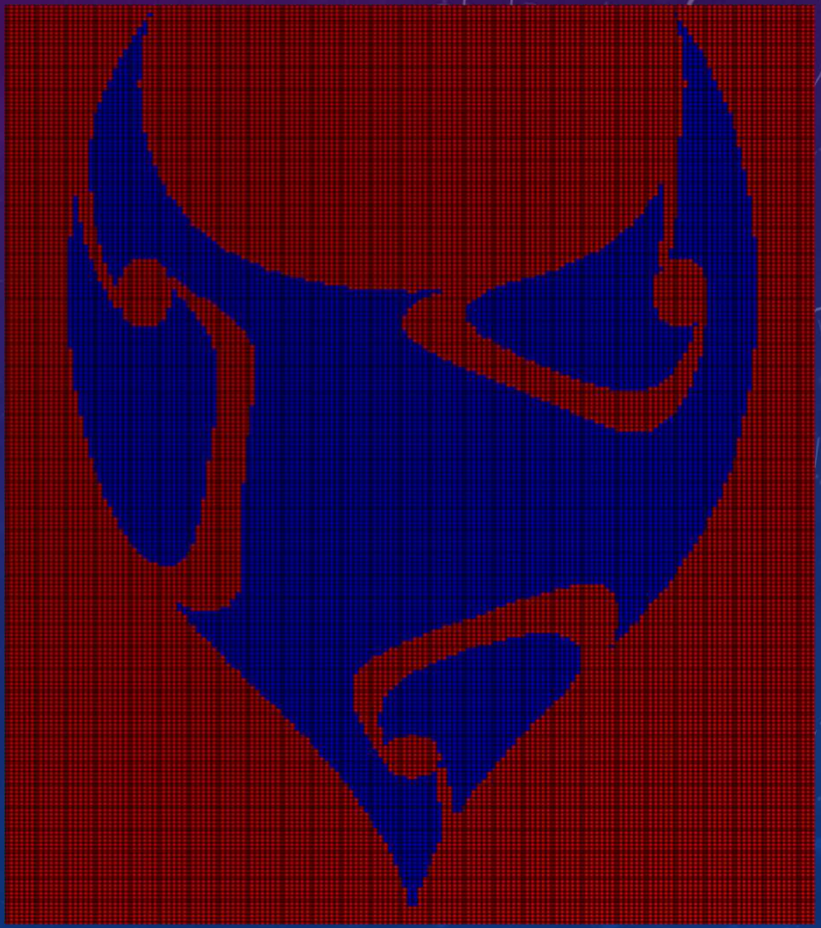
GRID SEARCH APPROACH

SINGULARITY MAP DISCRETIZATION

- The singularity map is constructed as a discretized grid of cells in the ϕ-θ plane.
- Each grid cell corresponds to a small region.
- The singularity status of each cell is determined by sampling points :

$$S_{ij} = \begin{cases} 0 & if\ any\ sampled\ point\ in\ the\ cell\ is\ singular, \\ 1 & otherwise\ (singularity - free) \end{cases}$$

- Step size is determined by the cell size.



CONTROL ALGORITHM

- **Step 1:** Obtain the current manipulator position in Euler angles (ϕ_S , θ_S).
- **Step 2:** Get the target position relative to the manipulator.
- **Step 3:** Choose a target solution (ϕ_T, θ_T) among all feasible ϕ-θ positions pointing to the target elevation, using one of two methods:
  - Furthest-from-singularity method

$$(\phi_T, \theta_T) = \arg \max_{(\emptyset,\, \theta)} (distance\ to\ singular\ points)$$

  - Closest-to-source method

$$(\phi_T, \theta_T) = \arg \min_{(\emptyset,\, \theta)} \|(\emptyset, \theta) - (\phi_S, \theta_S)\|_2$$

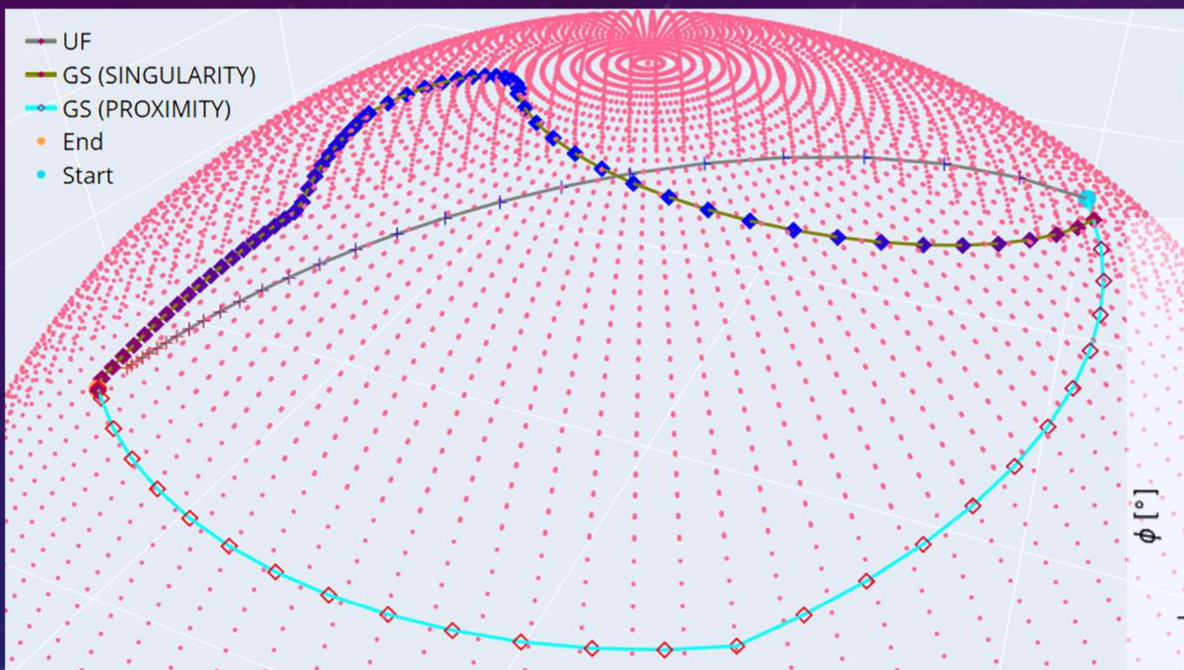
CONTROL ALGORITHM (CONT.)

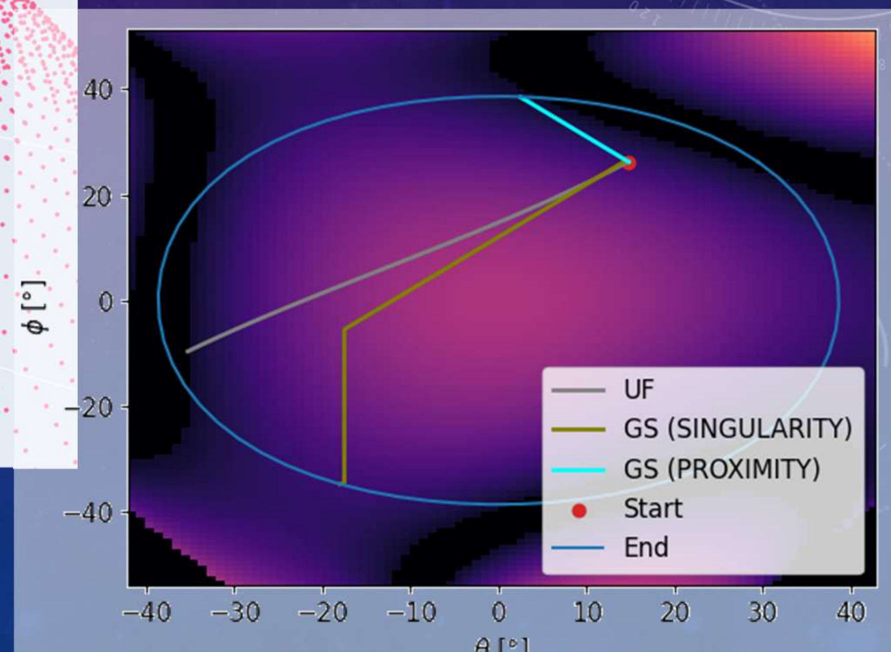
- **Step 4: Calculate a feasible path using some grid search algorithm (Dijkstra, A*, Greedy, Beam Search, etc.)**
- **Step 5: Follow the path to the target's cell using inverse kinematics, from one cell to the next, then move directly to the target,**

  Or:

- **Repeat (for tracking a moving target)**
  - **At each control cycle k:**
    - **Update the Platform's current position** $({\phi_S}^k, {\theta_S}^k)$
    - **Update the target's current position** $({\phi_T}^k, {\theta_T}^k)$
    - **Recalculate the path with an appropriate singularity map**
      - Coarse resolution for long trajectories.
      - Fine resolution for refinements near the target.
    - **Move to the next cell or directly to the target**

GRID SEARCH TRAJECTORIES



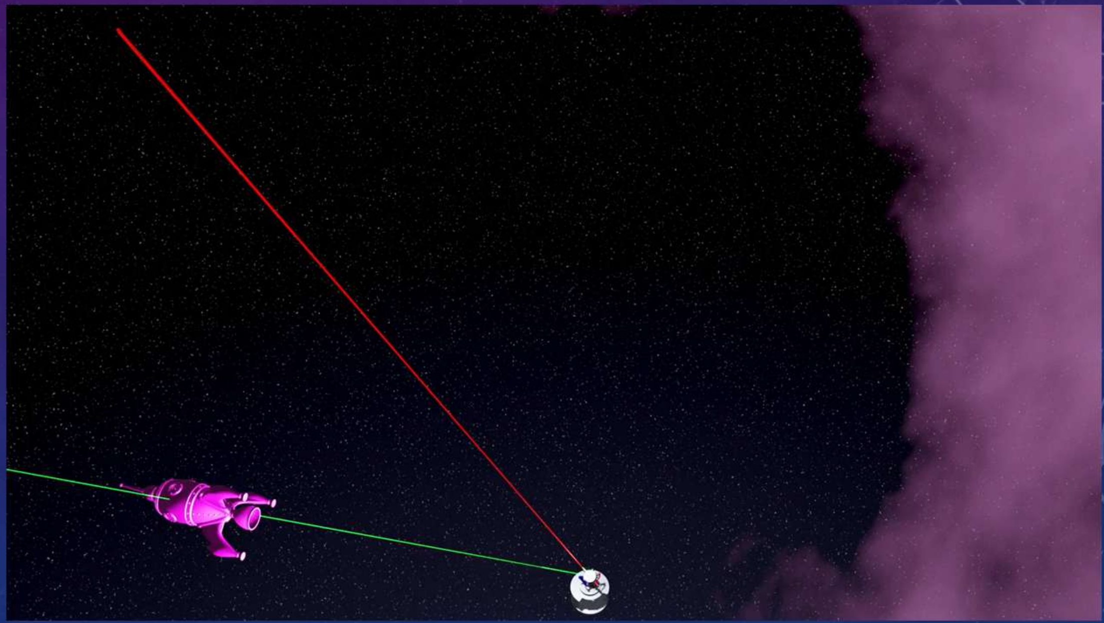


TRACKING

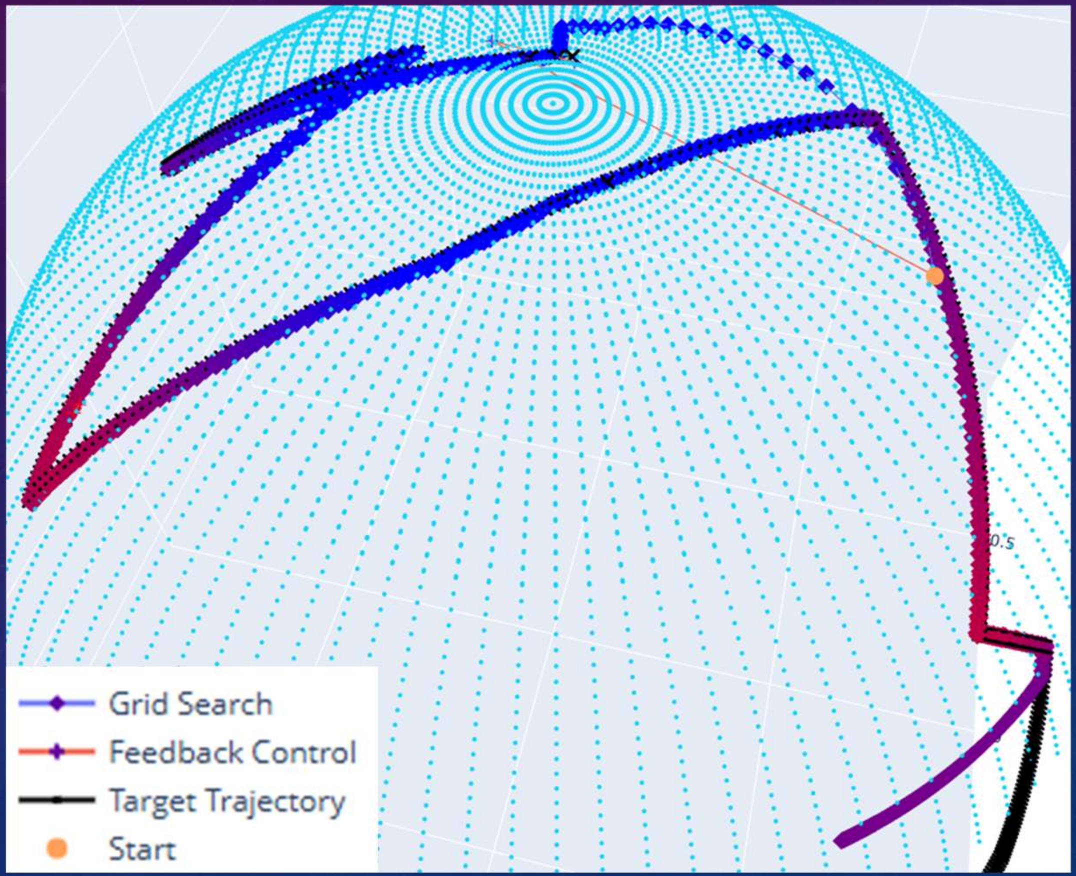
Seed=42

RANDOM LINEAR TARGET

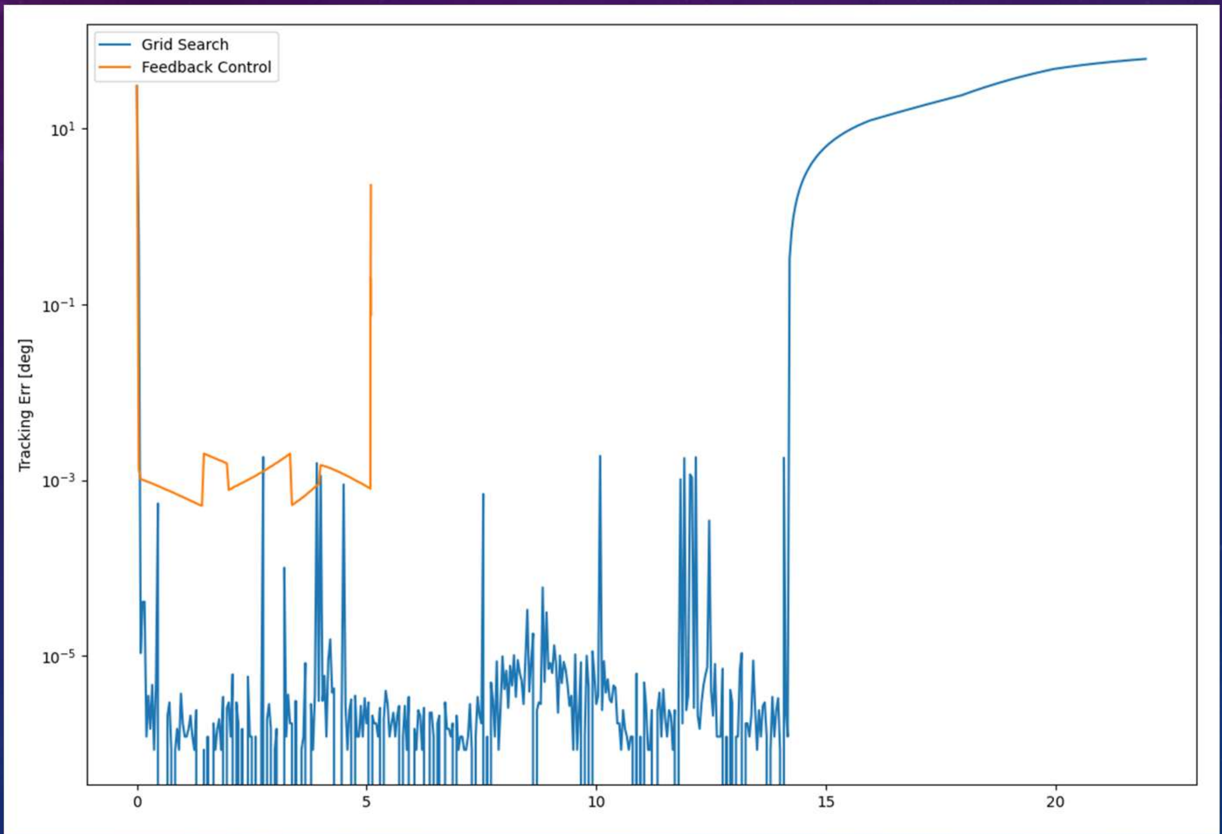
- Tracking Time = 22sec
- Duration for which the point moves in a single random direction before potentially changing its direction = 2sec
- Speed = 50m/sec
- Sampling Freq. = 24hz
- $p_0 = (10, 0, 100)\ m$



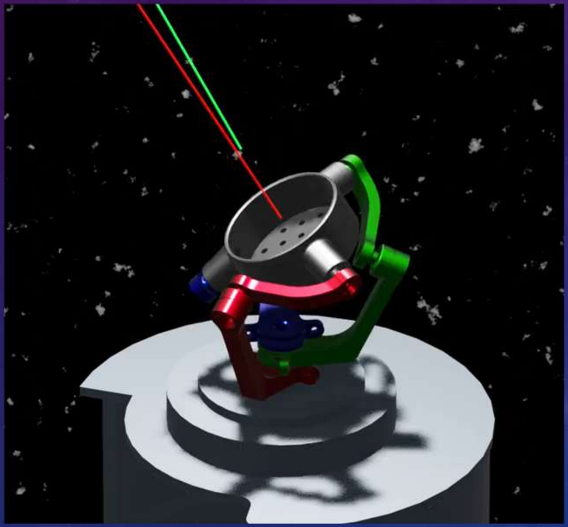
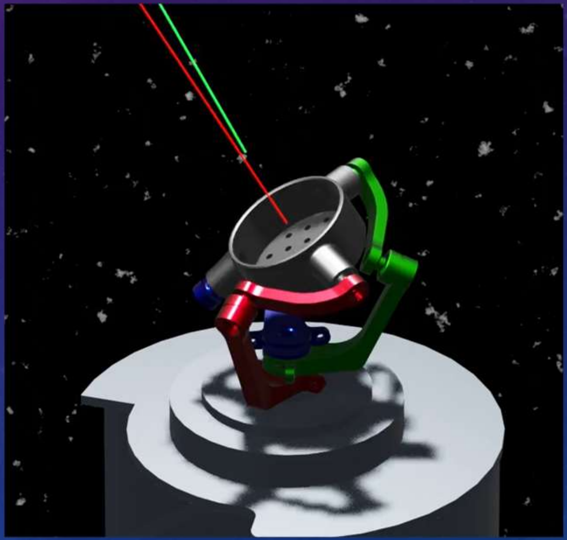
RANDOM LINEAR TARGET TRACKING LOS



RANDOM LINEAR TARGET TRACKING ERROR



RANDOM LINEAR TARGET TRACKING ANIMATION

Grid Search	Unconstrained Feedback Control
	

RANDOM LINEAR TARGET TRACKING ANIMATION

Grid Search	Unconstrained Feedback Control

DISCUSSION

DISCUSSION – PROS & CONS

RL (Discrete Actions)	Grid Search	Unconstrained Feedback Control

RL (Discrete Actions)

Pros:

- Robustness
- Efficiency
- High Success Rates

Cons:

- Pre-Training Overhead
- Non-Deterministic Behavior
- Reduced Maneuverability

Grid Search

Pros:

- Deterministic
- No Pre-Training
- Safety & Reliability

Cons:

- Computational Toll
- Path Optimality
- Smoothness
- Reduced Maneuverability

Unconstrained Feedback Control

Pros:

- Simplicity
- Shortest Trajectories
- Low Computational Cost
- High Maneuverability

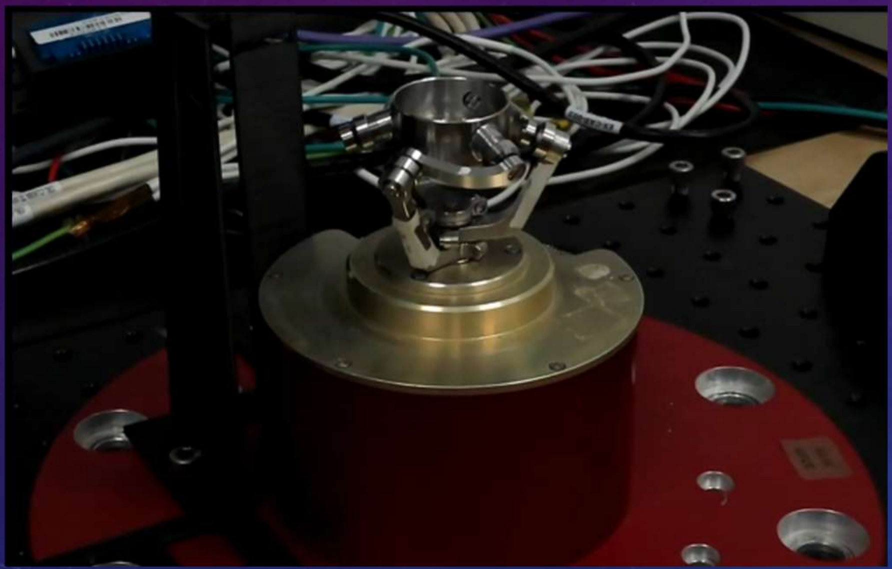
Cons:

- Lack of Singularity Avoidance
- Limited Robustness
- Tracking Limitations

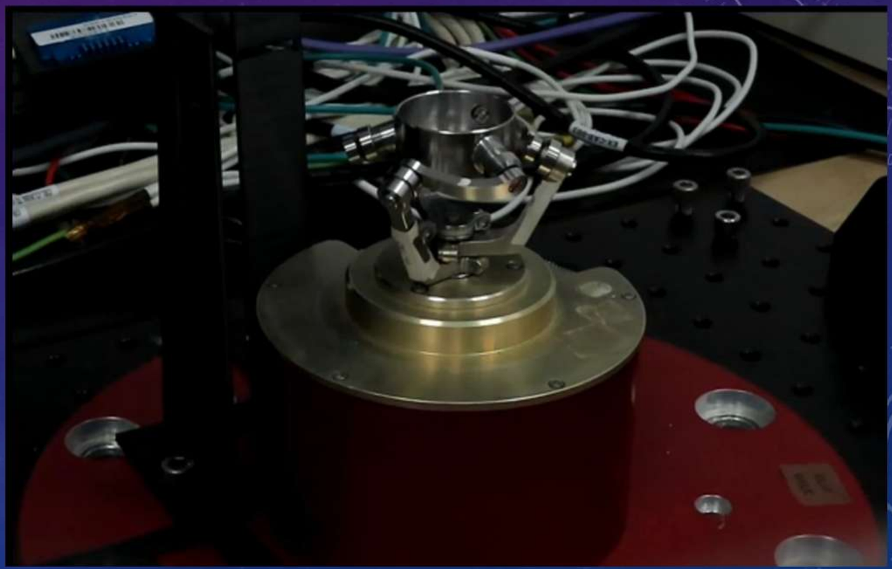
DISCUSSION - INSIGHTS AND FUTURE DIRECTIONS

- The reduced maneuverability in RL and Grid Search approaches is a trade-off for their safety and reliability.
- The unconstrained method excels in smooth and efficient motion but struggles with singularity avoidance at high elevation angles.
- The RL model, being a lightweight neural network, is well-suited for deployment on low-power hardware such as ASICs.
- The grid search approach requires CPU resources for real-time pathfinding and command conversion.
- The multi-resolution approach in Grid Search reduces computational costs, making its real-time performance comparable to RL inference.
- Hybrid approaches could leverage the strengths of the unconstrained method with RL or Grid Search to achieve optimal control.

RL DEMO EXPERIMENT



A2C



TD3