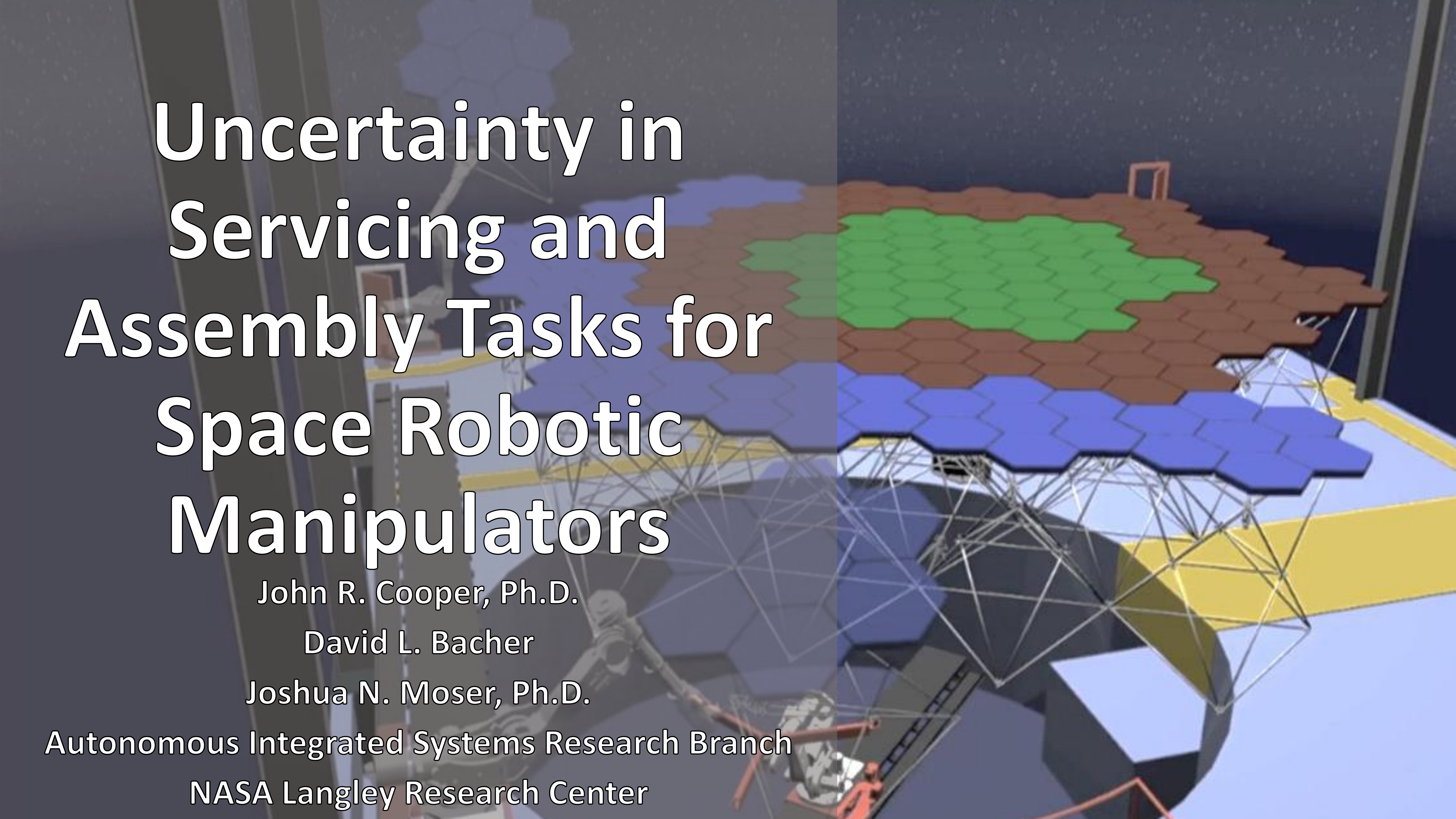


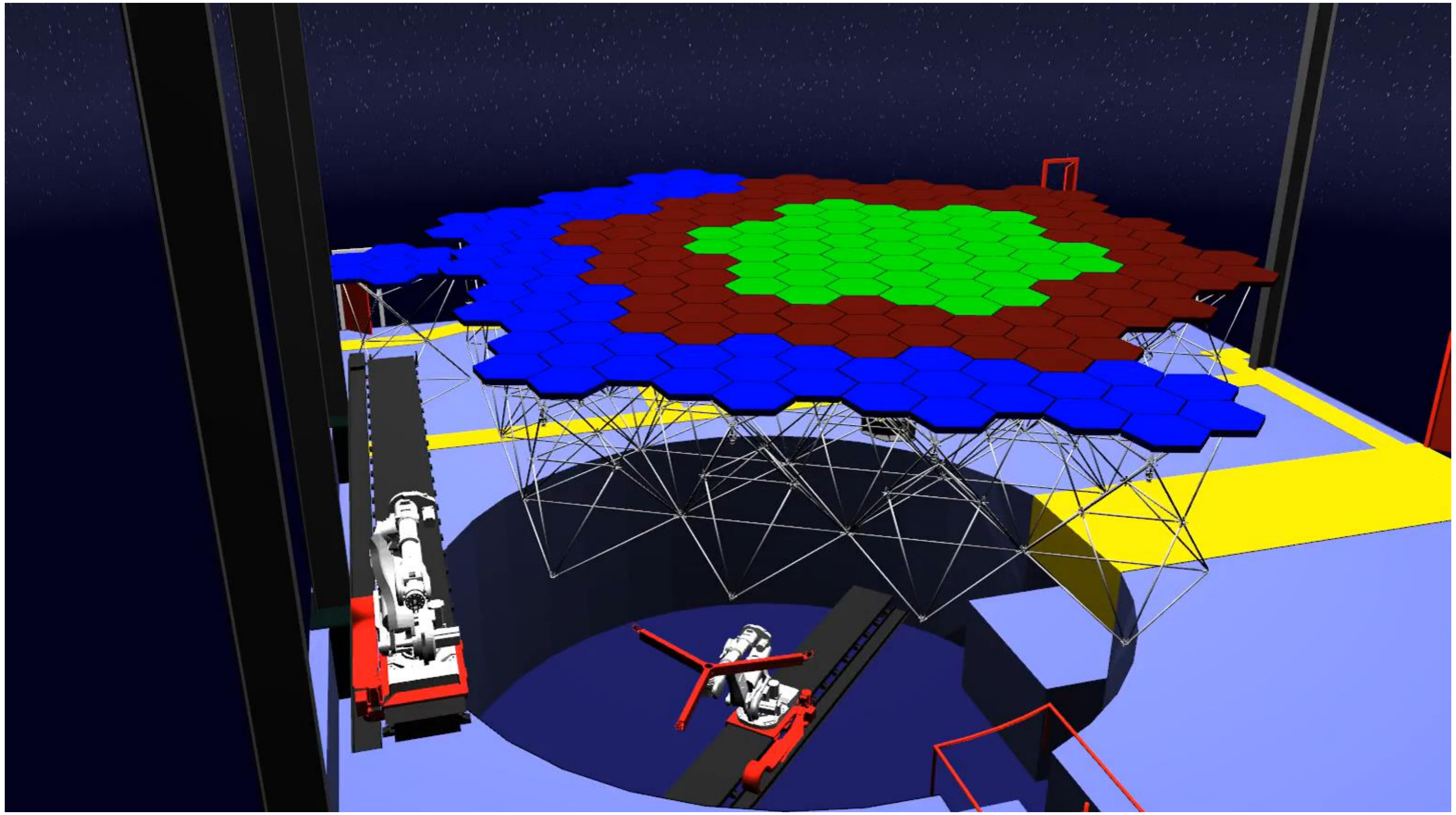
Uncertainty in Servicing and Assembly Tasks for Space Robotic Manipulators

John R. Cooper, Ph.D.
David L. Bacher
Joshua N. Moser, Ph.D.
Autonomous Integrated Systems Research Branch
NASA Langley Research Center

In-Space Assembled Telescope Ground Demo

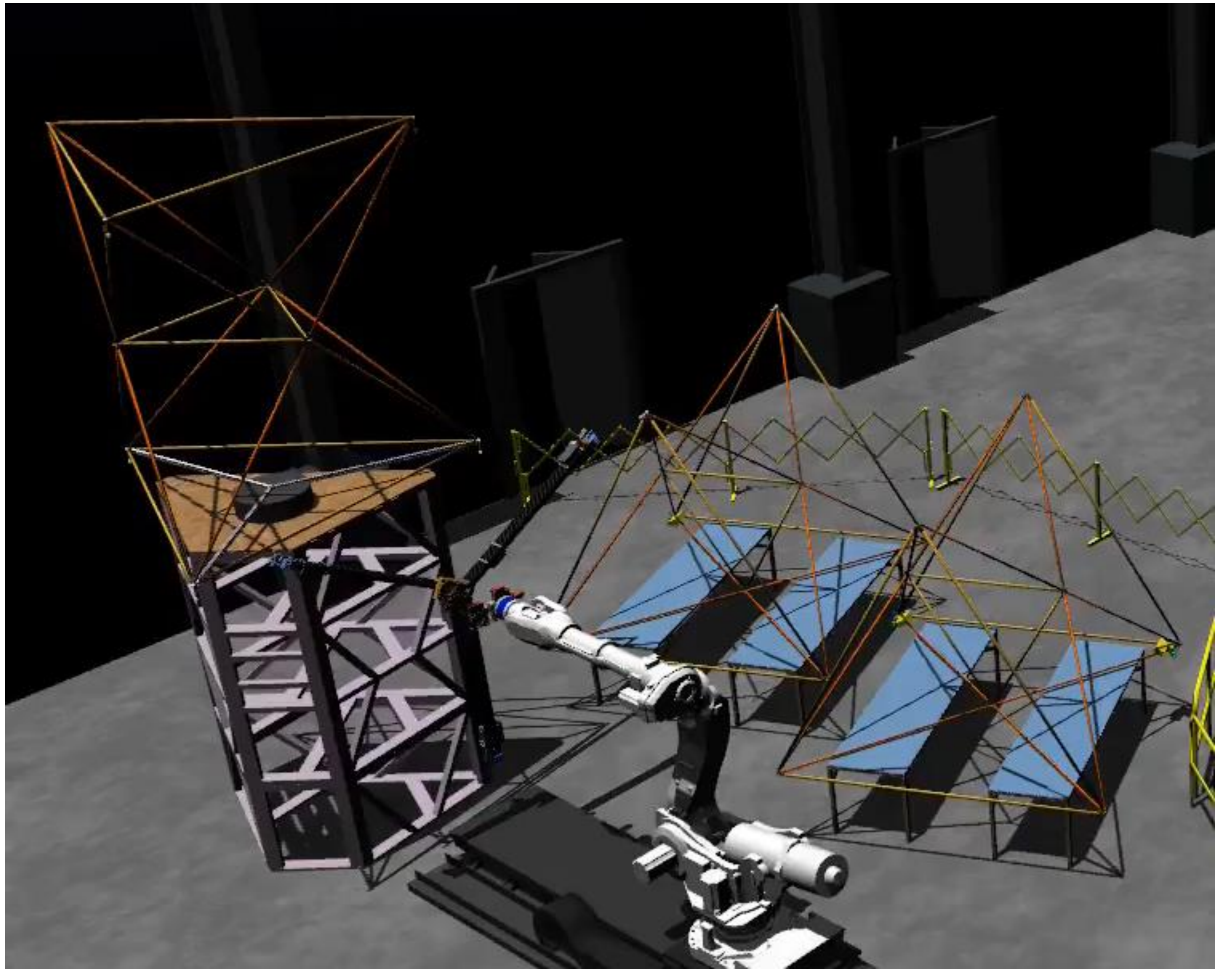
Langley Research Center

- Design reference mission based on in-Space Assembled Telescope study
- 20 m diameter mirror backbone structure
- Full scale assembly ground demonstration



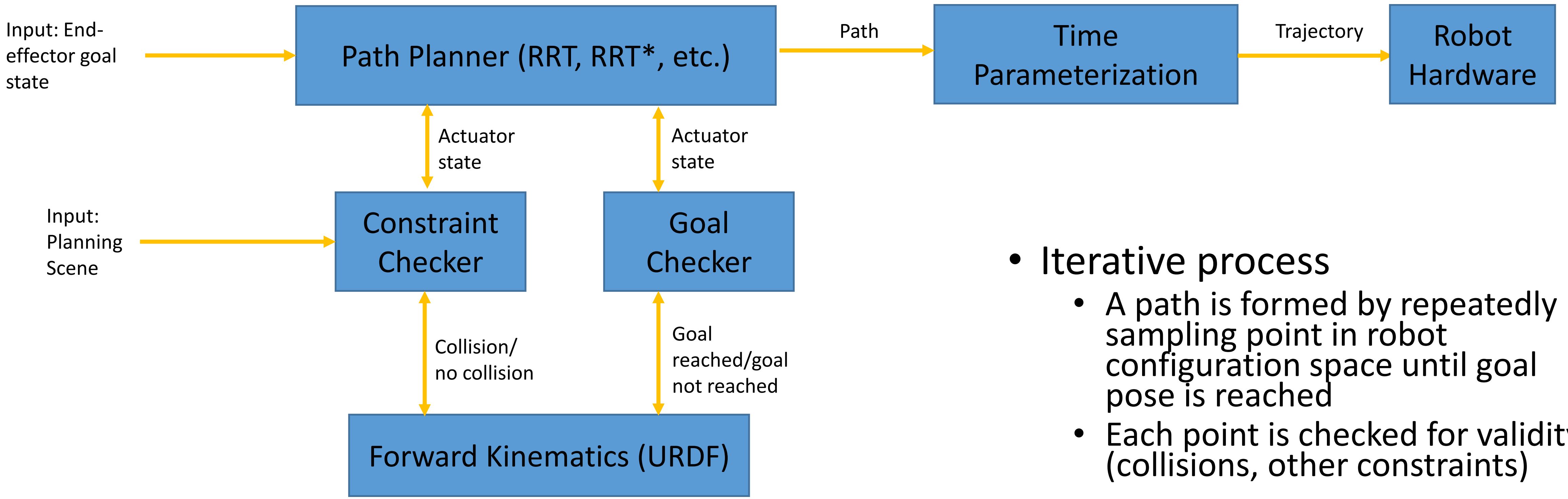
Phase 1a Demonstration

- Three-truss assembly
  - Custom-designed Grapple Tool grabs Tri-Truss modules
  - Motion planned and executed
  - Grapple tool joins Tri-Truss nodes
- Robot must move between multiple poses to pick up payloads (tools, trusses, etc.)
- Three relevant capture envelopes:
  1. Pick up grapple tool
  2. Pick up tri-truss
  3. Place tri-truss
- Trajectories must be collision-free
- Trajectories must be generated for the present workspace
  - Uncertain
  - Dynamic



Trajectory Generator Block Diagram

Langley Research Center

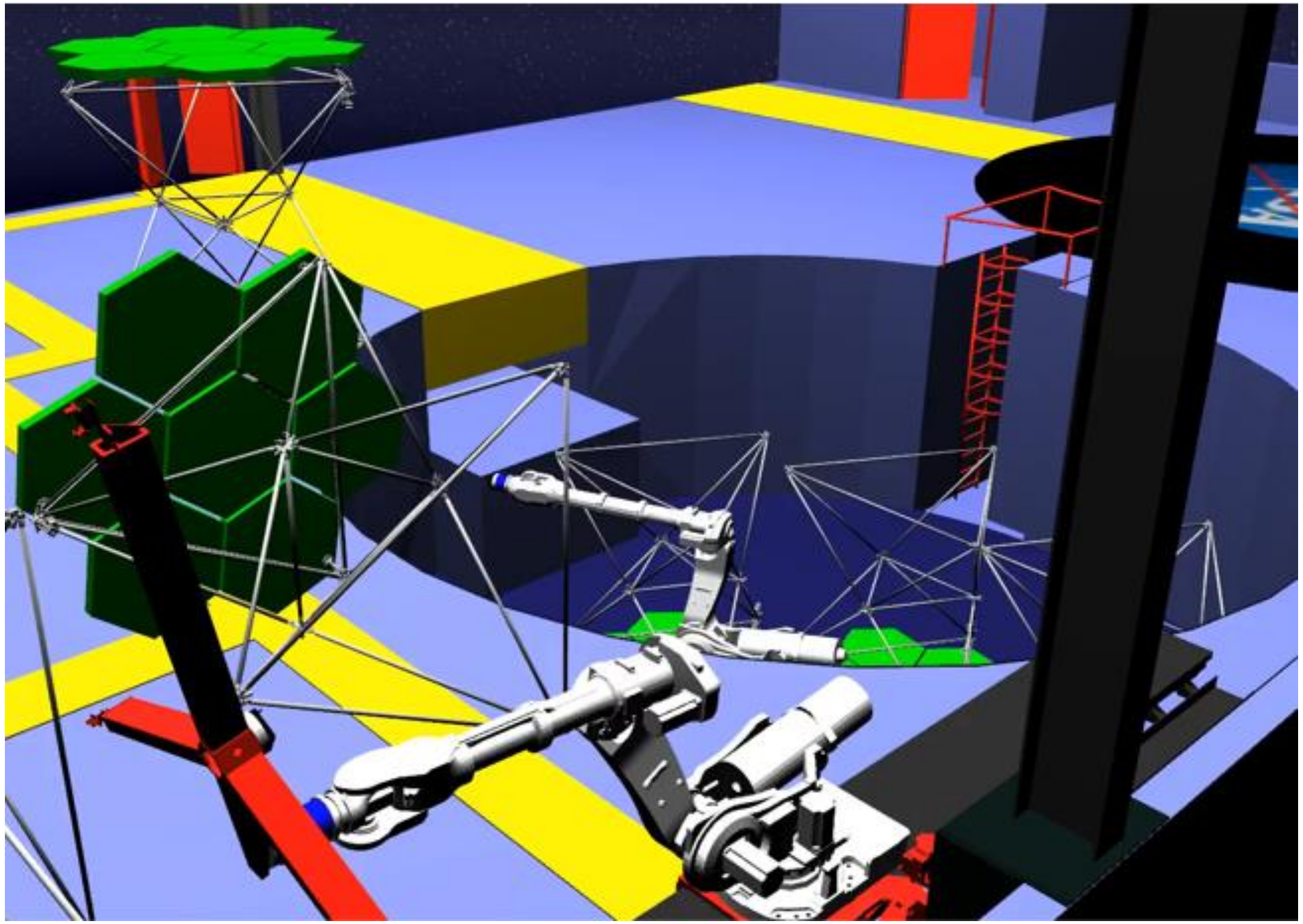


- Iterative process
  - A path is formed by repeatedly sampling point in robot configuration space until goal pose is reached
  - Each point is checked for validity (collisions, other constraints)
- Time is specified along the path according to max velocity and acceleration constraints
  - Create a trajectory from path

Planning Scene

Langley Research Center

- Planning scene represents all objects in the robot's worksite
- A-priori known objects are represented as meshes
- Dynamic and/or uncertain objects detected from external sensors can be added to scene via occupancy map representation



Manipulator Kinematics

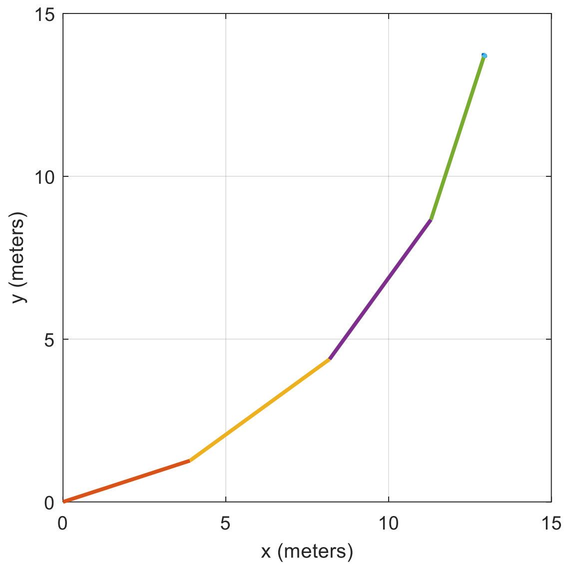
- End-effector variables a function of joint variables
- $x = f(q)$
- Differentiating shows us how end-effector variables change with joint variables
- $\delta x = \frac{df(q)}{dq}\delta q, \frac{df(q)}{dq} = J(q)$: manipulator Jacobian
- Our knowledge of x comes from sensor measurements of q, which may be noisy
- End-effector covariance related to joint covariance through Jacobian
- $C_x = J(q) C_q J^\top(q)$

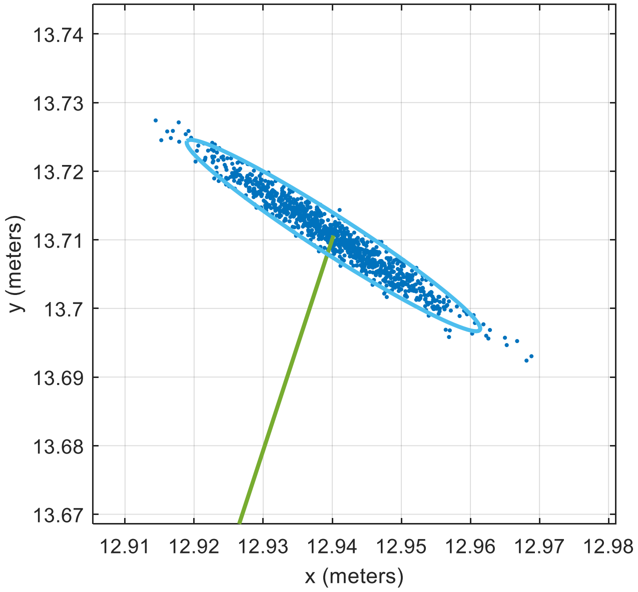
Manipulator Kinematics – Denavit Hartenberg Parameters

- End-effector variables as a function of 4 parameters per joint
- $x = f(d, \theta, r, \alpha)$
- $[Z_i] = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, [X_i] = \begin{bmatrix} 1 & 0 & 0 & r_{i,i+1} \\ 0 & \cos\alpha_{i,i+1} & -\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- $[T] = [Z_1][X_1][Z_2][X_2] \ldots [Z_{n-1}][X_{n-1}][Z_n][X_n]$
- Differentiating shows how end-effector variables change with joint variables
- $\delta x = \frac{df}{d\theta}\delta\theta, \frac{df}{d\theta} = J(\theta)$: manipulator Jacobian
- Our knowledge of x comes from measurements of d, θ, r, α, which may be noisy or contain other uncertainties
- End-effector covariance related to joint covariance through Jacobian
- $C_x = J(q) C_q J^\top(q)$

Uncertainty Ellipsoid

- Eigenvalues of covariance matrix correspond to the spread of data in direction of the corresponding eigenvector
- 2D example: $\frac{x_\lambda^2}{\lambda_{\max}} + \frac{y_\lambda^2}{\lambda_{\min}} = k^2$
- $k = \sqrt{-2\ln(1 - Pr)}$, where Pr is confidence threshold
- Setting $Pr = 0.99$ yields an ellipsoid that encloses 99% of the data
- **We want to ensure the capture envelopes relevant to our con-ops enclosed an ellipsoid with appropriate confidence threshold**





Kinematic Model Calibration

- Uncertainty in DH parameters can be mitigated through calibration

1. Take external measurements of real robot end-effector pose (Vicon, laser tracker, etc.)
2. Compare measurements to model output
3. Run optimization routine to find best fit DH parameters

$$\text{minimize } E(d,\theta,r,\alpha) = \left(x_{meas} - f(d,\theta,r,\alpha)\right)^2$$

- May need separate calibrations for different regions of the workspace due to gravitational sag
- Can use multiple calibration results to define uncertainty in DH parameters

Uncertainty Mitigation in Path Planning

- Path planning algorithm is sample-based
- Each sample checked for collisions and whether it satisfies goal state conditions
- If covariance ellipsoid is not enclosed within capture envelope, the sampled state is not a feasible goal state
- Note it's possible for the end-effector to be in the correct pose with too large uncertainty ellipsoid
- Including this extra goal check increases likelihood of mission success
- Alternatively, include metric for uncertainty in numerical inverse kinematics optimizer to find best goal state up front

<u>Path Planning Algorithm</u>

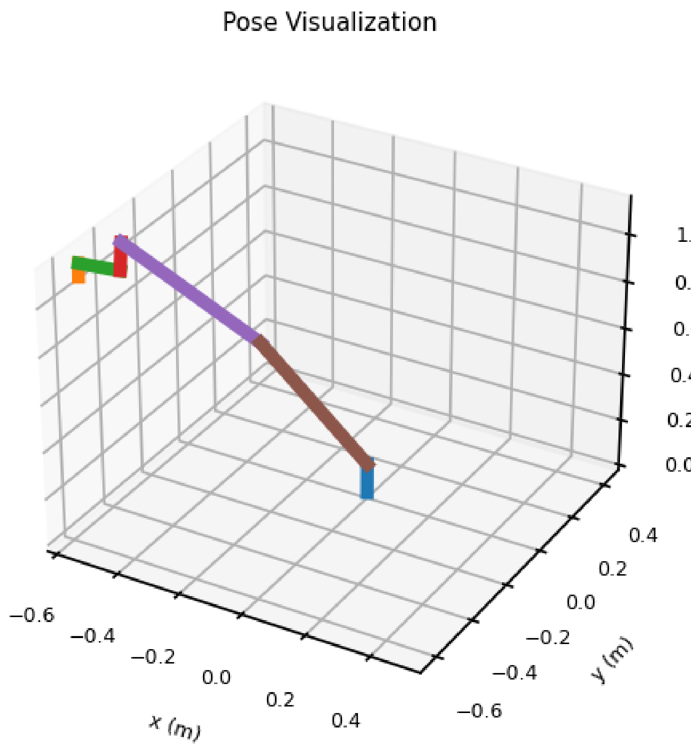
```
for n iterations:
    sample a joint space
    if sample is feasible:
        add sample to tree
        if sample is a goal state:
            goal <- sample
        rewire tree for optimization
```

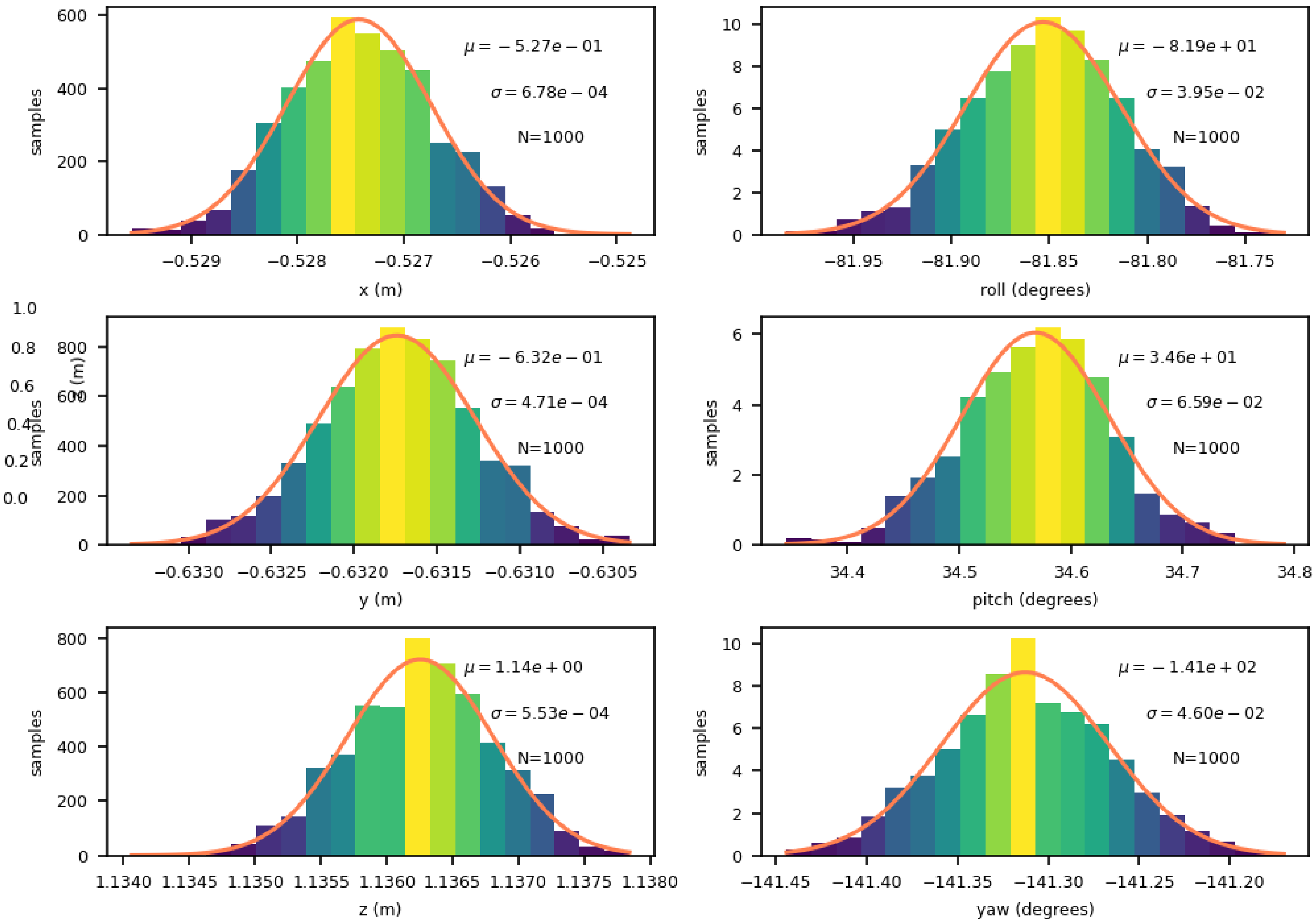
Monte Carlo Analysis

- Example: UR10 manipulator
  - AksIM encoder accuracy +/- 0.1 degrees
    - Interpret as 99% confidence interval (3 x std. dev.)
- Monte Carlo sampling of joint angles within kinematic model
- Plot histogram of results and fit normal distribution
- Perform for several nominal poses

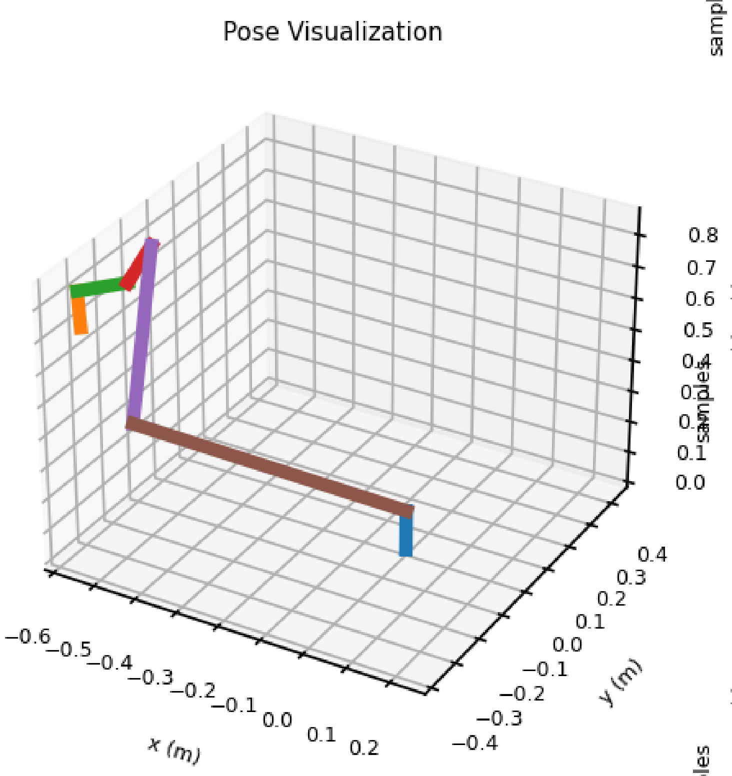
UR10				
Kinematics	theta [rad]	r [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1273	$\pi/2$
Joint 2	0	-0.612	0	0
Joint 3	0	-0.5723	0	0
Joint 4	0	0	0.16394	$\pi/2$
Joint 5	0	0	0.1157	$-\pi/2$
Joint 6	0	0	0.0922	0

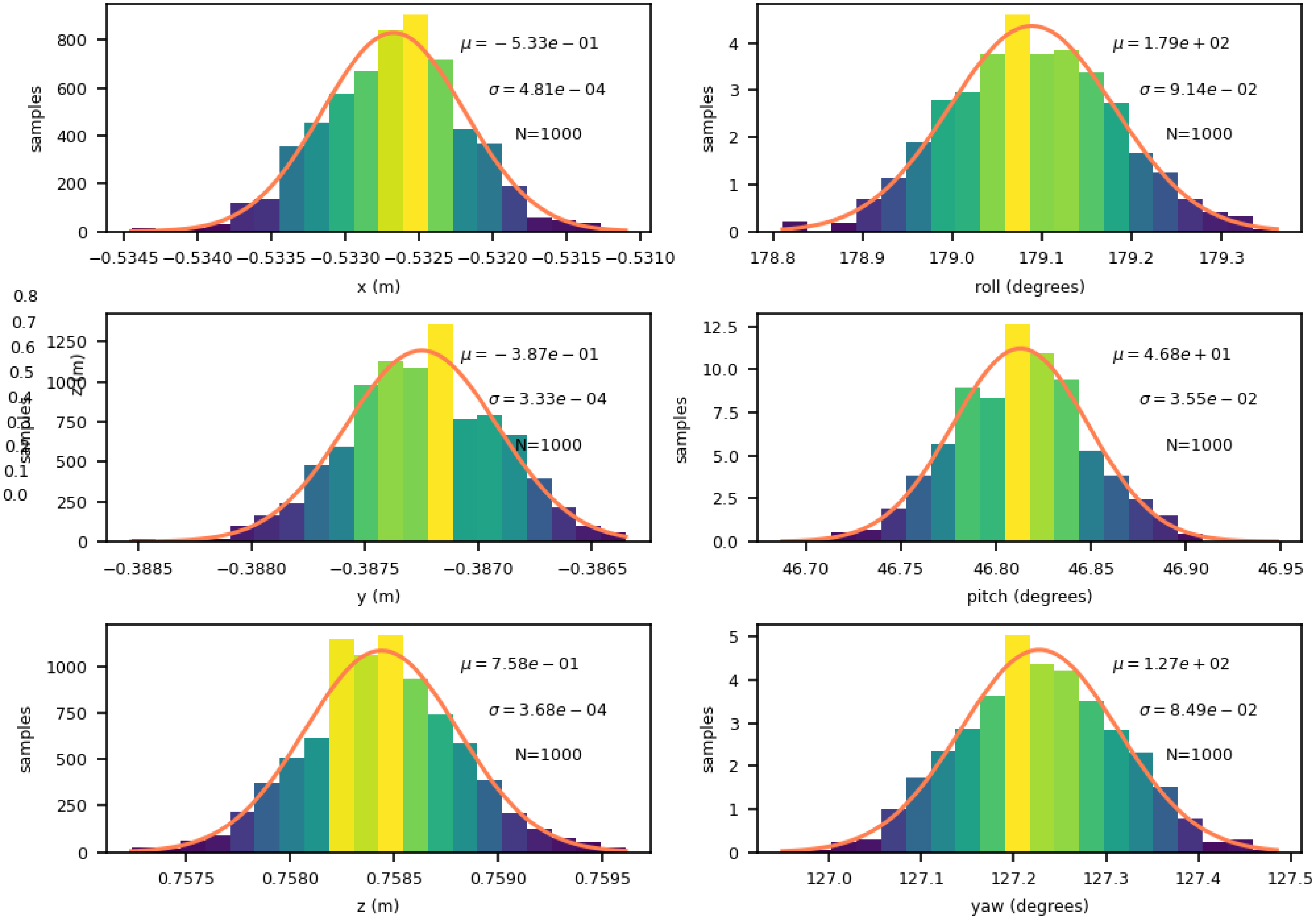
Pose 1


Pose Visualization
x (m)
y (m)
z (m)
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.0
0.2
0.4
0.6
0.8
1.0

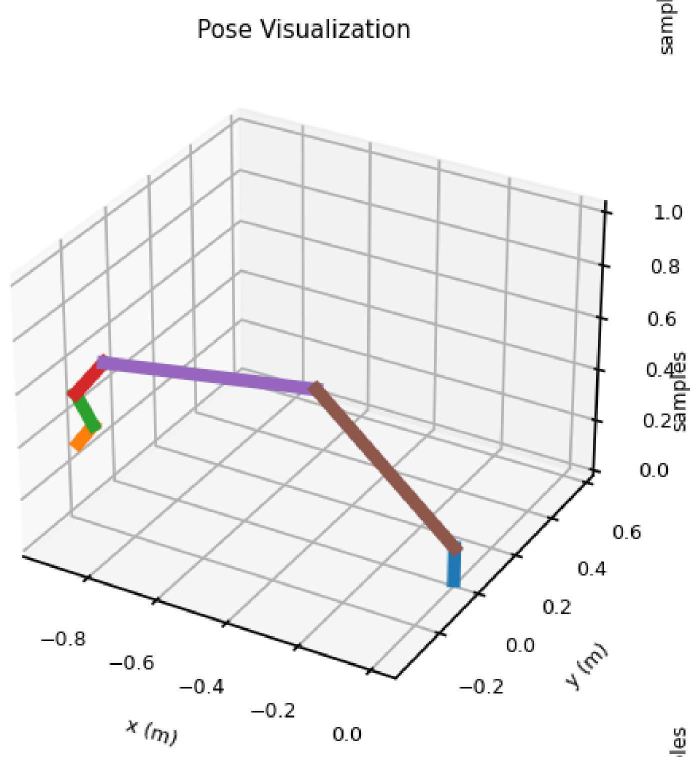

samples
x (m)
$\mu = -5.27e-01$
$\sigma = 6.78e-04$
N=1000
roll (degrees)
$\mu = -8.19e+01$
$\sigma = 3.95e-02$
N=1000
y (m)
$\mu = -6.32e-01$
$\sigma = 4.71e-04$
N=1000
pitch (degrees)
$\mu = 3.46e+01$
$\sigma = 6.59e-02$
N=1000
z (m)
$\mu = 1.14e+00$
$\sigma = 5.53e-04$
N=1000
yaw (degrees)
$\mu = -1.41e+02$
$\sigma = 4.60e-02$
N=1000

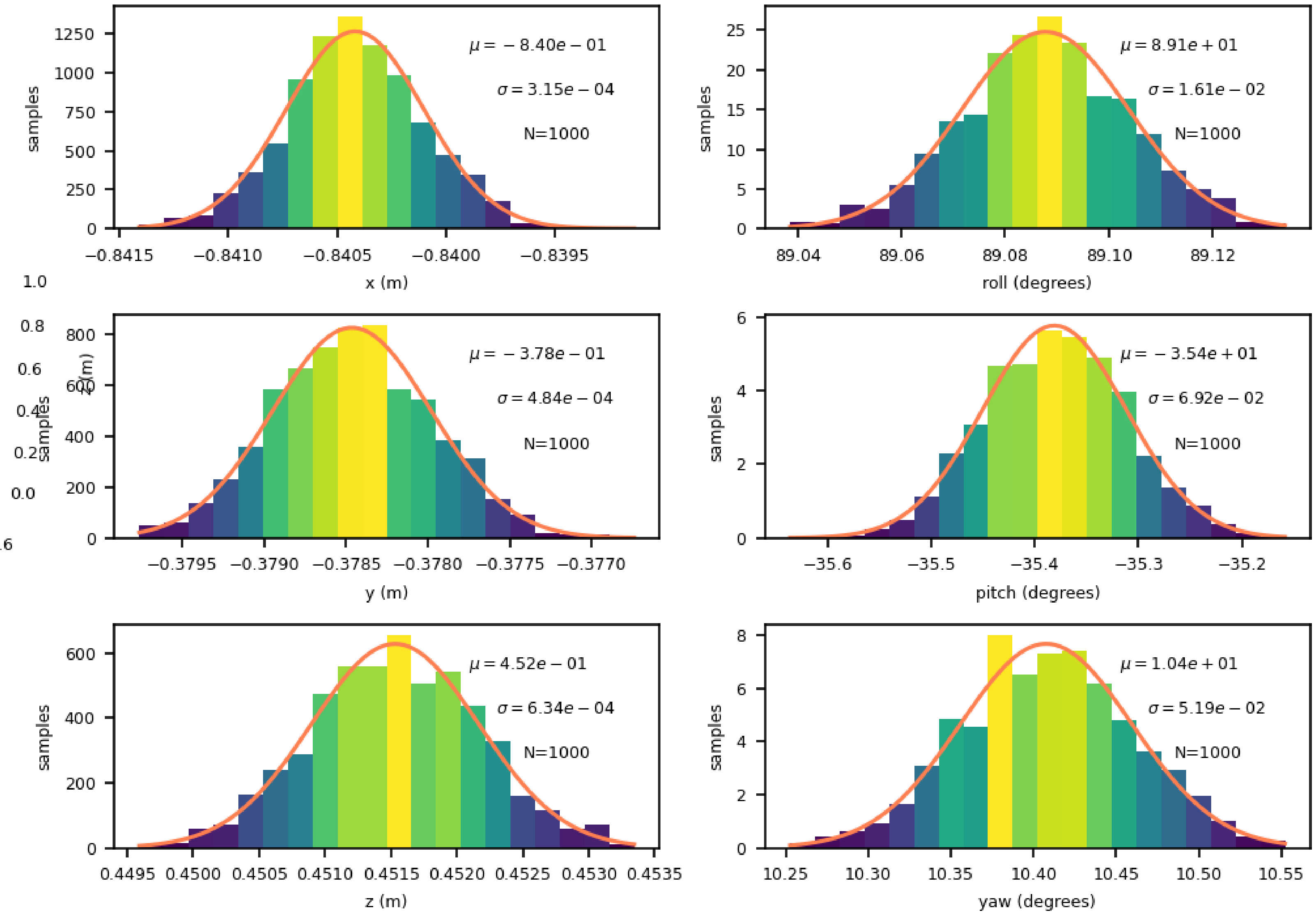
Pose 2


Pose Visualization
x (m)
y (m)
z (m)
−0.6
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


samples
x (m)
$\mu = -5.33e-01$
$\sigma = 4.81e-04$
N=1000
roll (degrees)
$\mu = 1.79e+02$
$\sigma = 9.14e-02$
N=1000
y (m)
$\mu = -3.87e-01$
$\sigma = 3.33e-04$
N=1000
pitch (degrees)
$\mu = 4.68e+01$
$\sigma = 3.55e-02$
N=1000
z (m)
$\mu = 7.58e-01$
$\sigma = 3.68e-04$
N=1000
yaw (degrees)
$\mu = 1.27e+02$
$\sigma = 8.49e-02$
N=1000

Pose 3


Pose Visualization
x (m)
y (m)
z (m)
−0.8
−0.6
−0.4
−0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0


samples
x (m)
$\mu = -8.40e-01$
$\sigma = 3.15e-04$
N=1000
roll (degrees)
$\mu = 8.91e+01$
$\sigma = 1.61e-02$
N=1000
y (m)
$\mu = -3.78e-01$
$\sigma = 4.84e-04$
N=1000
pitch (degrees)
$\mu = -3.54e+01$
$\sigma = 6.92e-02$
N=1000
z (m)
$\mu = 4.52e-01$
$\sigma = 6.34e-04$
N=1000
yaw (degrees)
$\mu = 1.04e+01$
$\sigma = 5.19e-02$
N=1000

Minimizing End Effector Pose Uncertainty

$\delta x = J(q)\delta q$

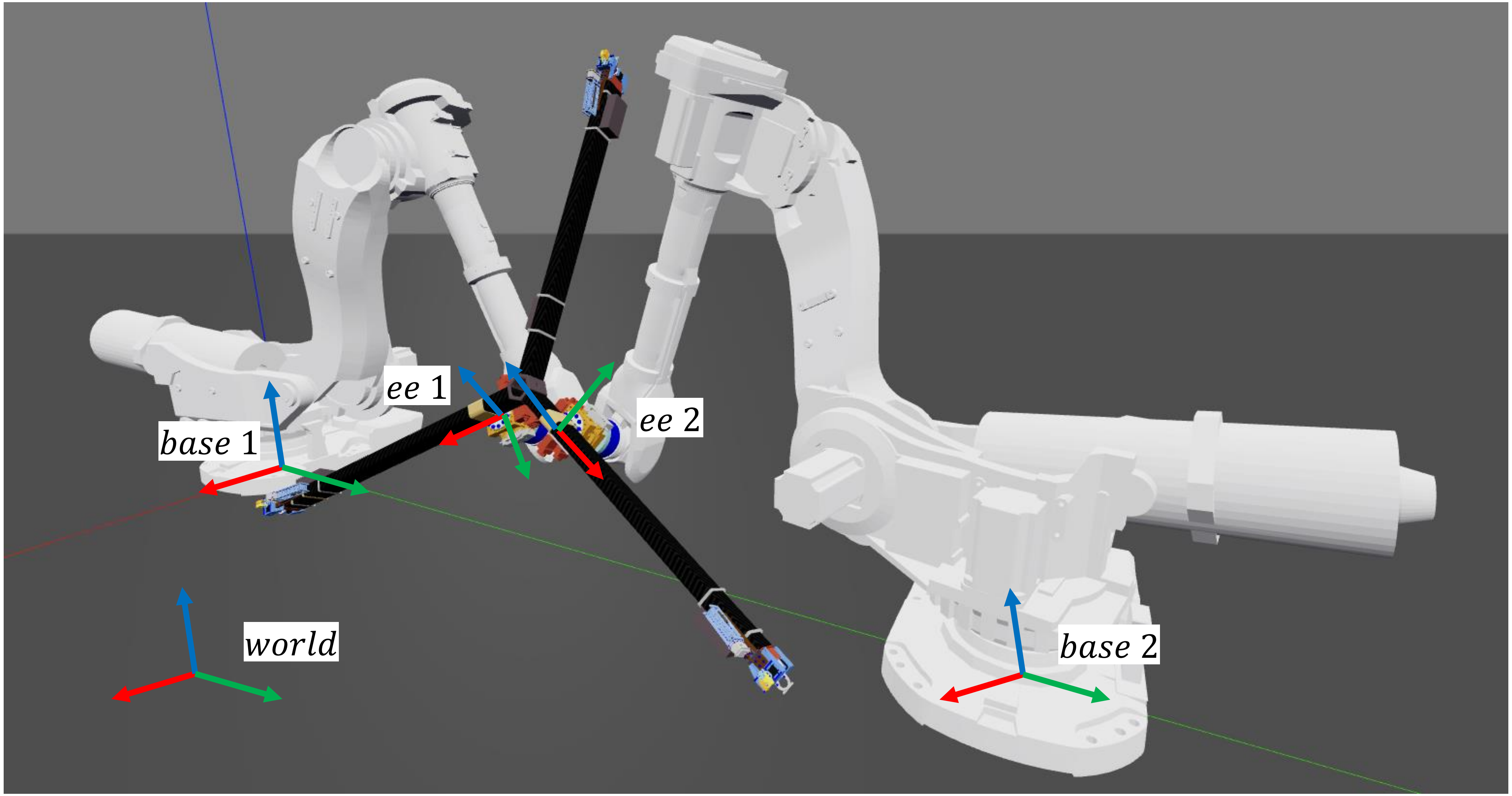
Then: $||\delta x|| \leq ||J(q)||_F ||\delta q||$

Where: $||J(q)||_F = \sqrt{Trace(J(q)^T J(q))}$: Frobenius Norm

Decreasing the Frobenius Norm decreases the bounds on uncertainty

$\boldsymbol{\delta x}$ is minimized by minimizing $||\boldsymbol{J(q)}||_{\boldsymbol{F}}$

Coordinate Frames



PASS PI: Bill.Doggett@nasa.gov
Autonomy Tech Lead:
john.r.cooper@nasa.gov

Handoff Pose Separation

At handoff, the end effectors have a known difference in pose, which can be expressed as a 4x4 SE(3) matrix:

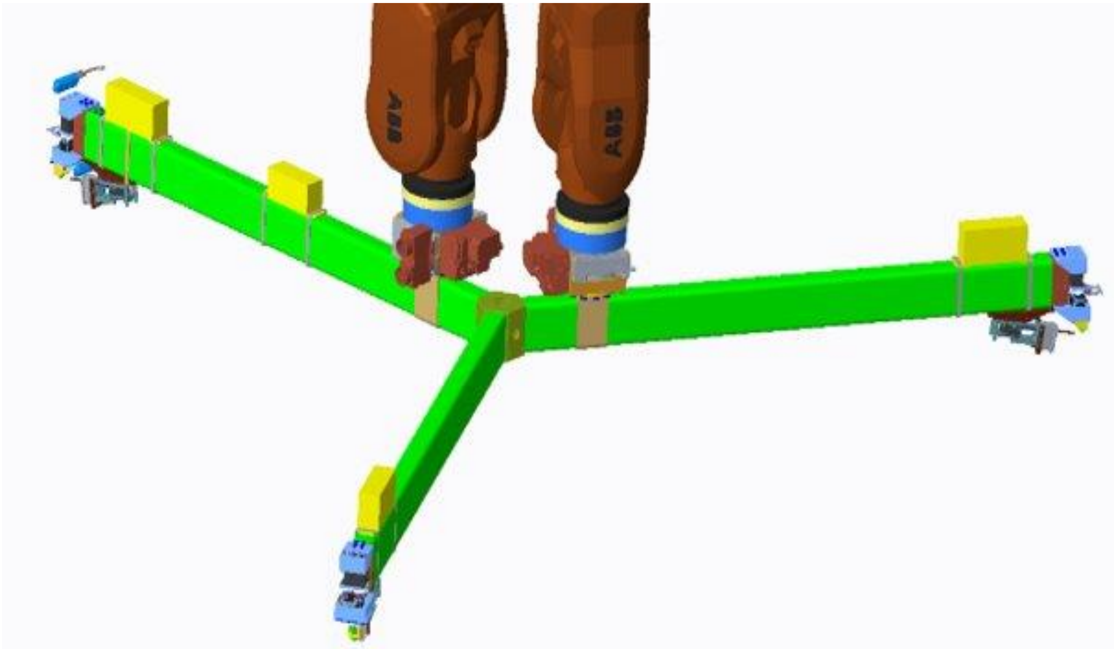
$$SE(3) = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where:

t: translation coordinates

R: elements of the rotation matrix

Let T^*_{e12} be a transformation matrix from the pose of ee1 to the pose of ee2 in the ee1 frame



The handoff transformation matrix is defined by the grapple tool design:

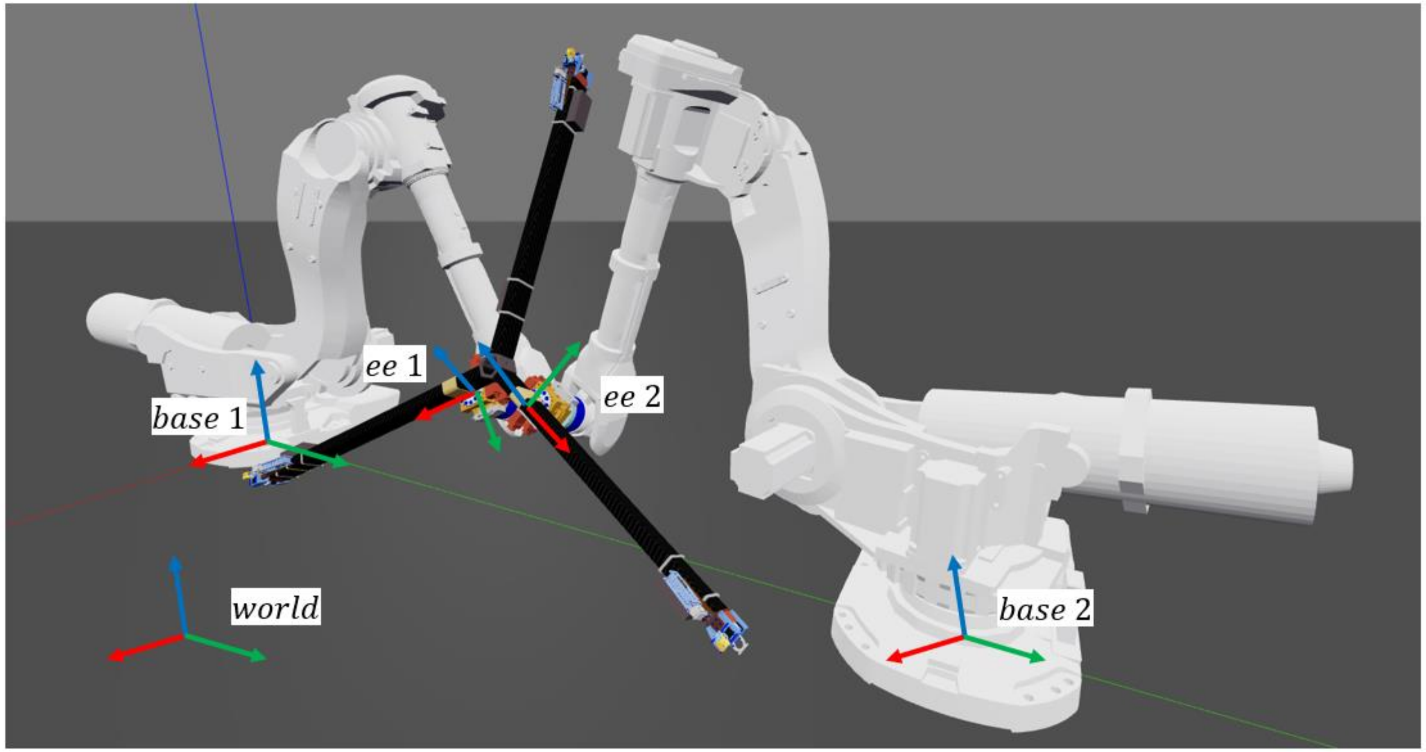
$$T^*_{e12} = \begin{bmatrix} -.5 & -.866 & 0 & -.375 \\ .866 & -.5 & 0 & .217 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Optimization Problem Formulation

$$T_{e1} = f_1(q_1): world \rightarrow ee\ 1$$

$$T_{e2} = f_2(q_2): world \rightarrow ee\ 2$$

$$T_{e12} = {T_{e1}}^{-1} T_{e2}: ee\ 1 \rightarrow ee\ 2$$



$$\min c(q_1, q_2) = \|J_1(q_1)\|_F + \|J_2(q_2)\|_F$$
$$s.t.\ {T_{e1}}^{-1} T_{e2} = T^*_{e12}$$
$$d_{min}(q_1, q_2) > 0$$

d_{min}: a function that returns the minimum distance between each robot and all other objects in the scene to prevent obstacles

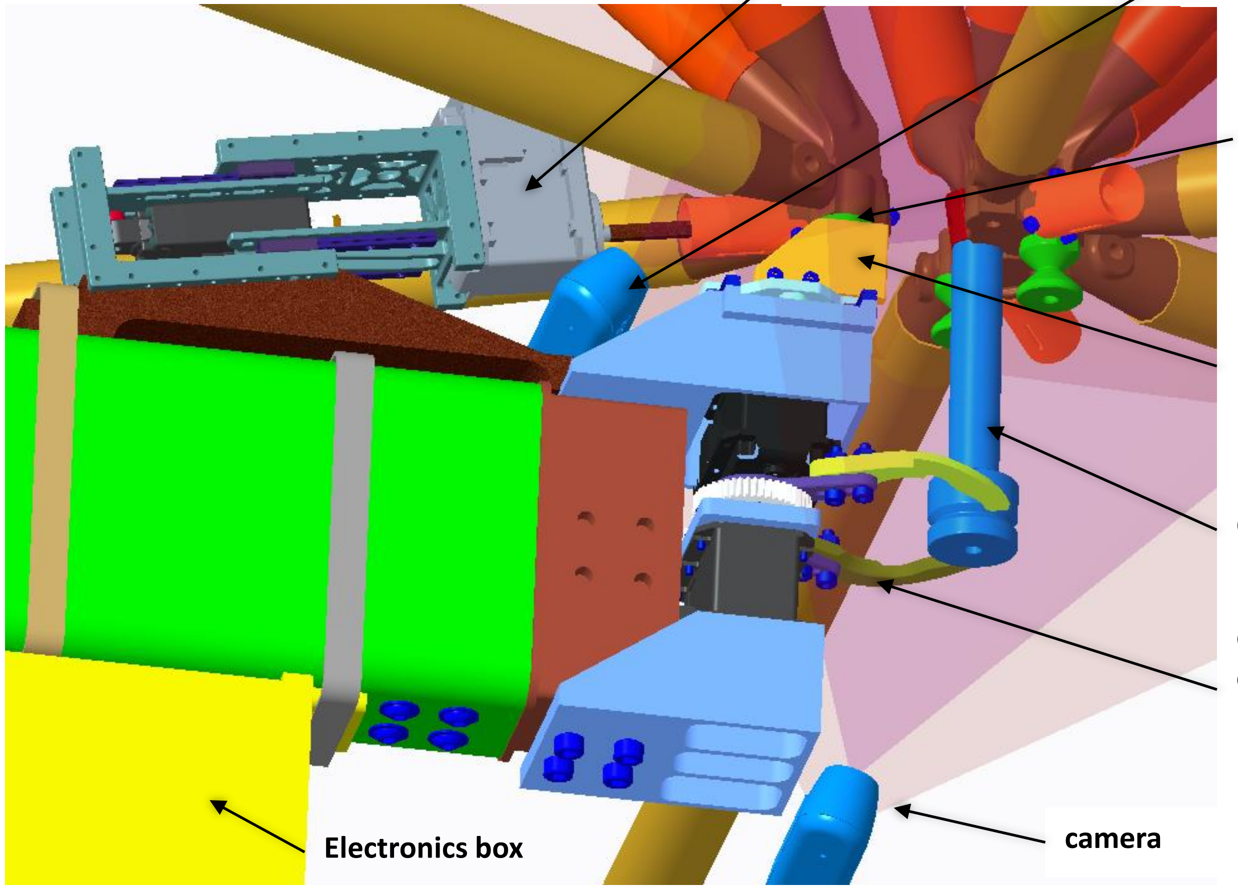
Object Pose Estimation Overview

- Algorithm that estimates pose of objects for robot to interact with from camera image and point cloud
  - Use convolutional neural network (CNN) to segment object image
  - Match object pixels to points
  - Use optimization to align CAD model with point cloud to get pose
- Used as a visual-servo sensor
- Need to understand uncertainty in these estimates and relate to capture envelopes
- Source of uncertainty:
  - Sensor noise
  - CNN performance
  - Cost function in optimization

These can each be tweaked if analysis shows uncertainty is too large

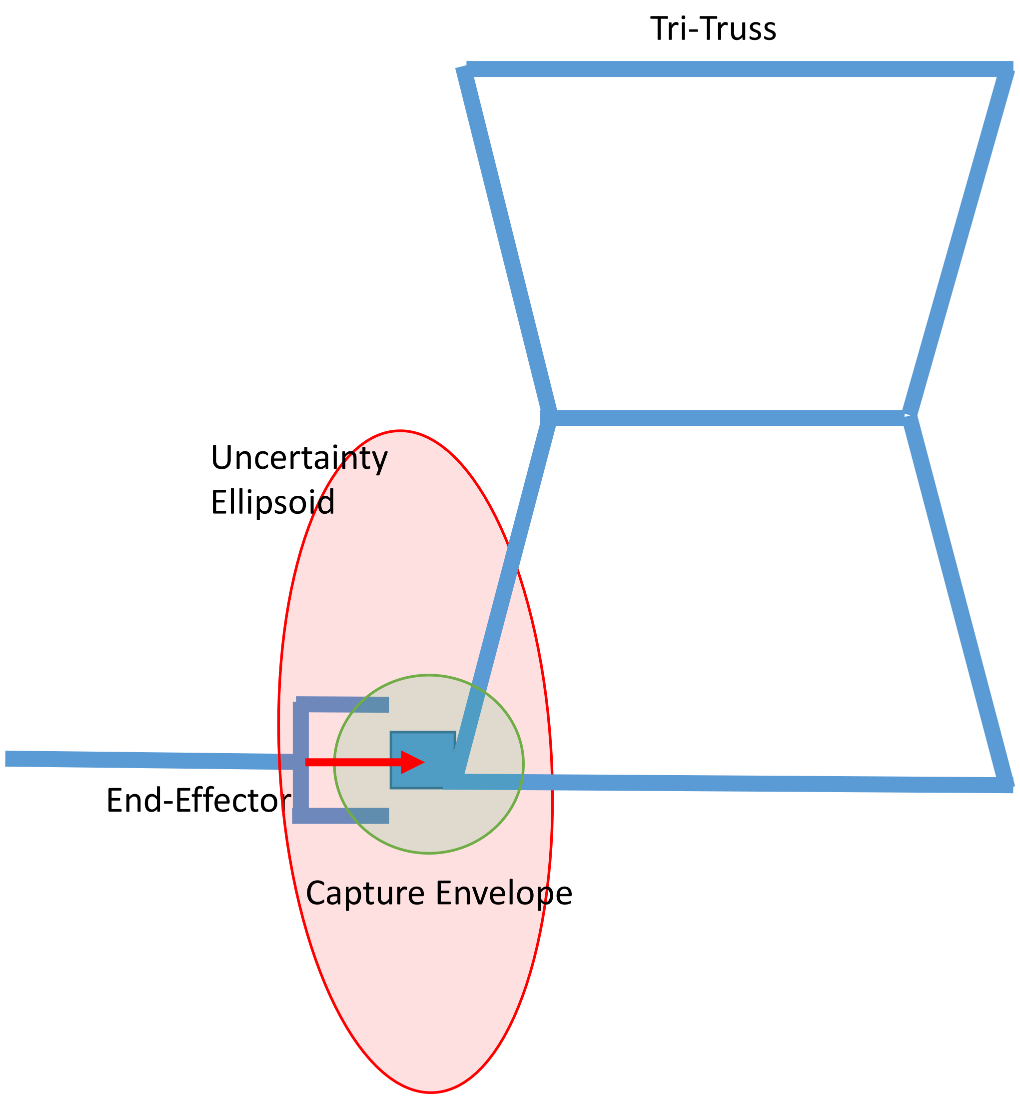
Capture Con-ops

- Grapple tool must be positioned such that capture fingers can grasp capture post and post fingers can grasp grapple post
- Each finger mechanism has a dedicated camera/LiDAR sensor package for pose estimation

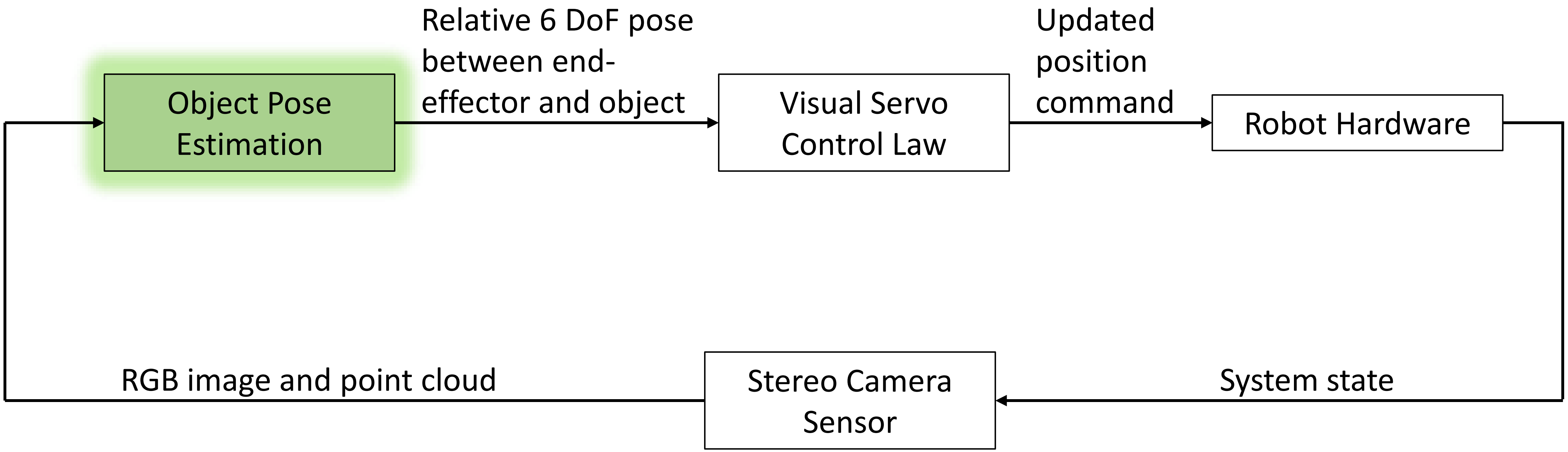


Grapple Con-Ops

1. Manipulator moves to "pre-pose" capture point
2. Object pose estimation algorithm provides relative pose between end-effector and capture target
3. Object pose estimation vector used to generate offset command for end-effector

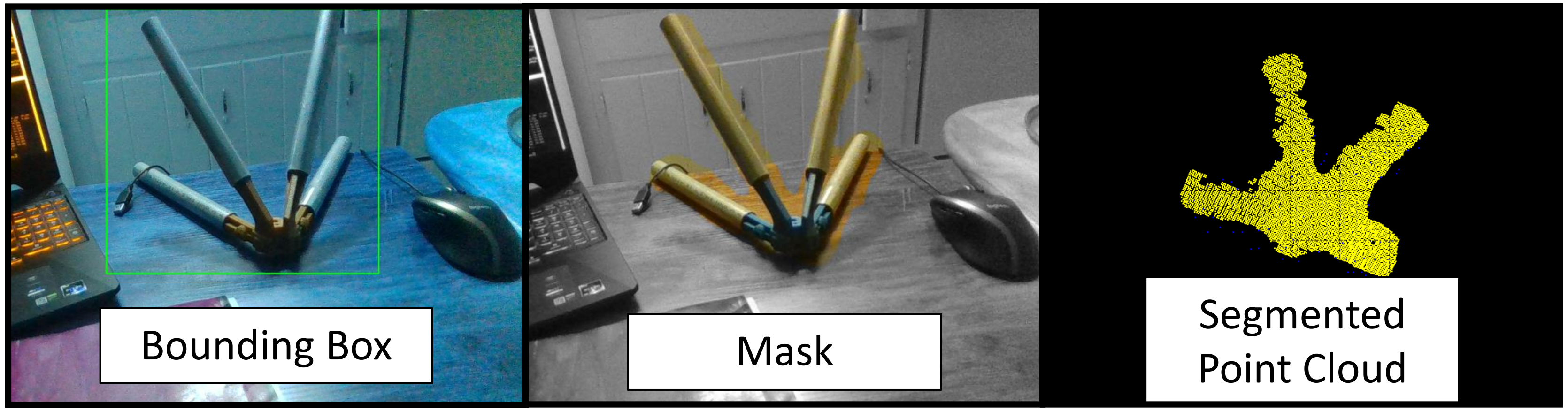


Visual Servo Control Loop

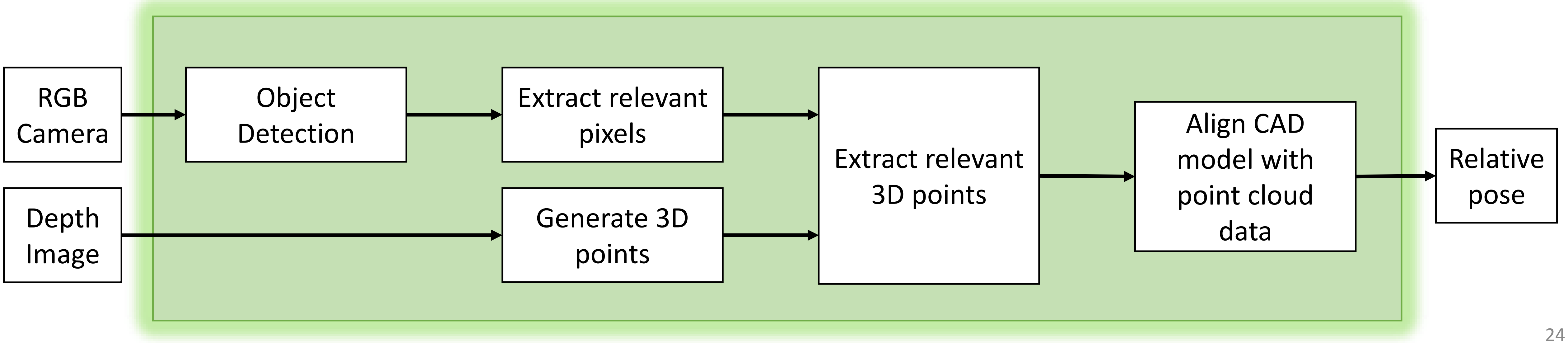


Pose Estimation Pipeline

Intel Realsense D405 depth camera

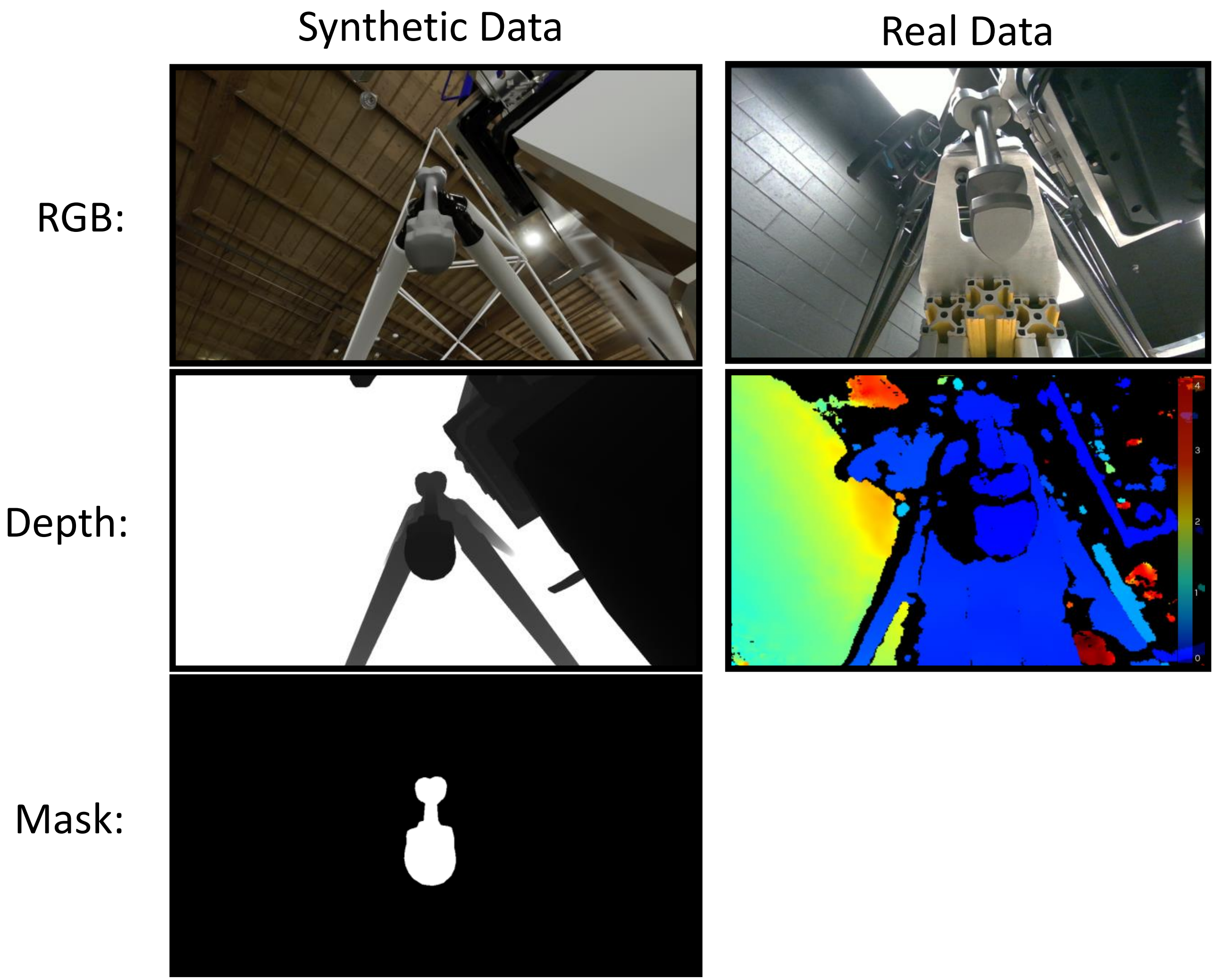


Pose Estimation Block Diagram



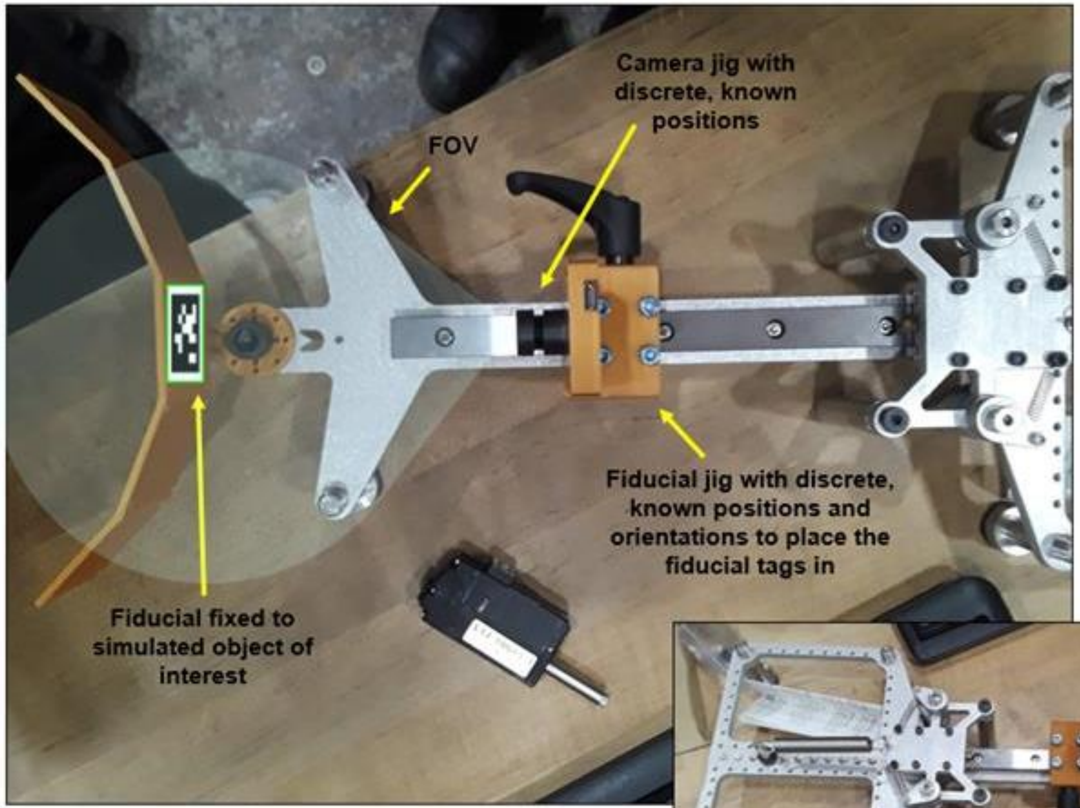
Testing Approach

- Generate a large dataset of synthetic images and point clouds with ground truth for relative pose
  - Import CAD models into rendering software
  - Generate point cloud from mesh
  - Control pose of object and camera
  - Add noise to image and point cloud
    - Real noise characteristics can be measured from physical sensor
- Independent variables to define different test cases:
  - Workspace region
  - Lighting conditions
  - Material properties
  - Target object
- Evaluate error between estimated pose and true pose



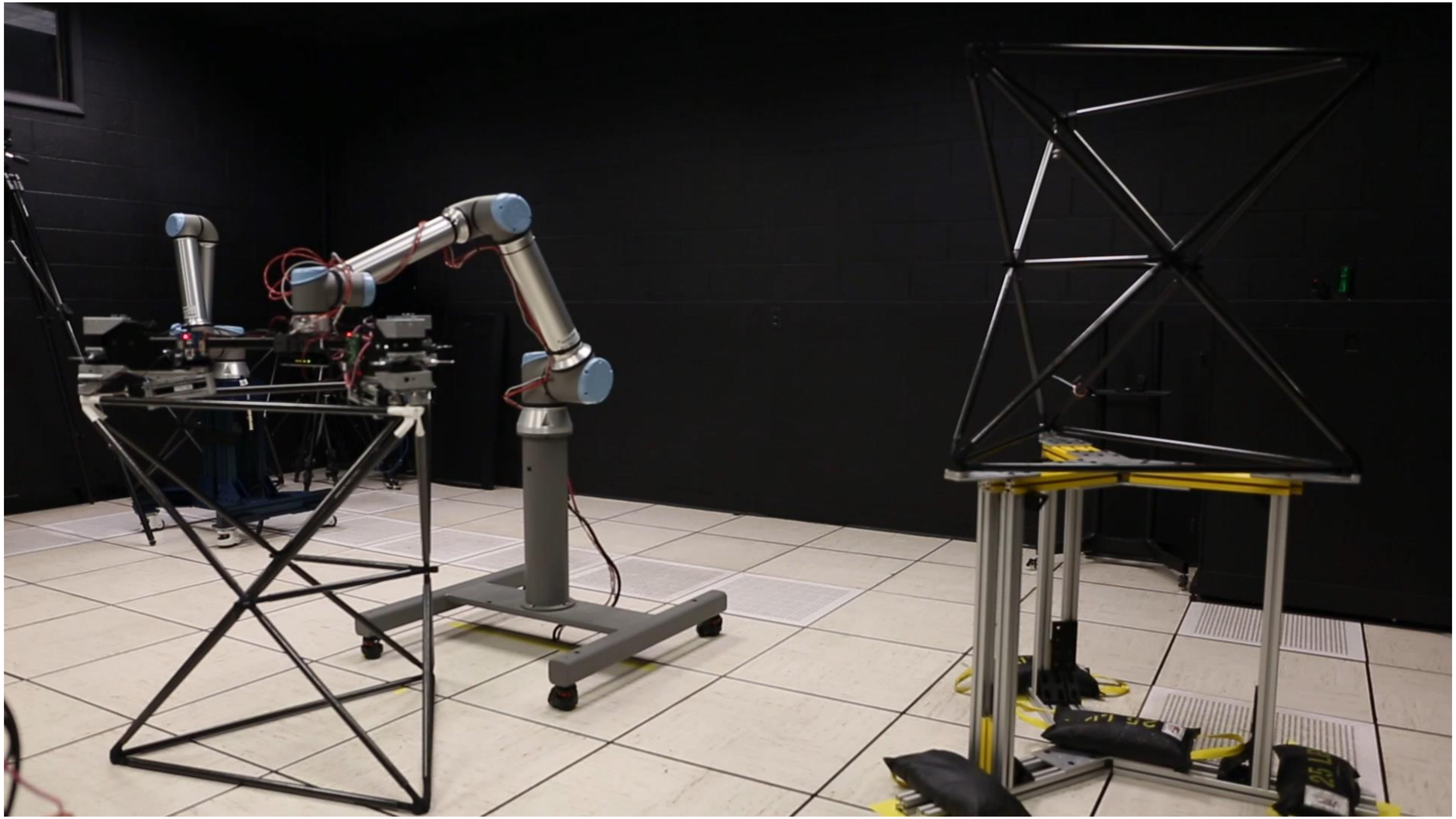
Verification of Synthetic Data

- Need to verify algorithm performs similarly on real data
- Build a test rig with discrete mounting points for both camera and test article
- Create a set of synthetic data that matches all test rig combinations
- Compare pose estimate on synthetic data and real data



Similar Test Rig

Truss Placement Hardware Test



Summary

- Many sources of uncertainty
  - Modeling
  - Sensor noise
  - Machine learning accuracy
- Mitigations
  - Use uncertainty quantification to design capture envelopes
  - Choose goal poses with minimum sensitivity