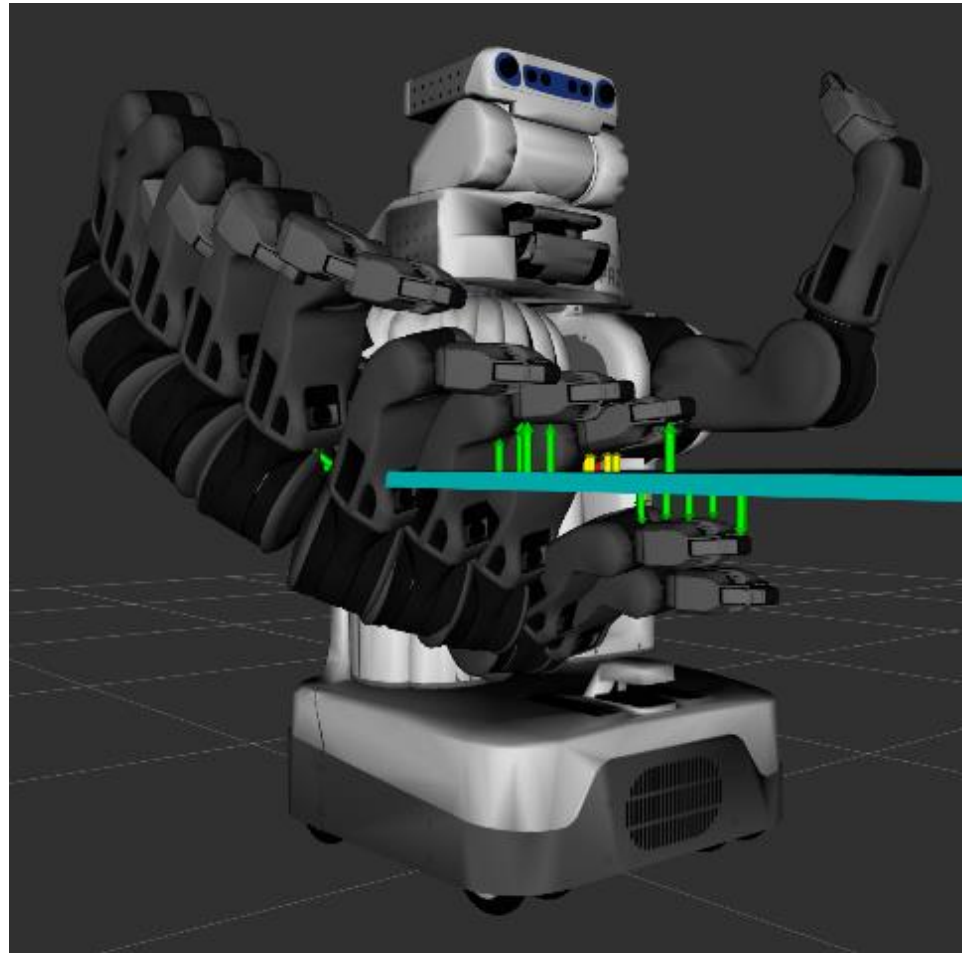
Optimization Motion Planning with TrajOpt and Tesseract for Industrial Applications

Levi Armstrong and Jonathan Meyer
Southwest Research Institute

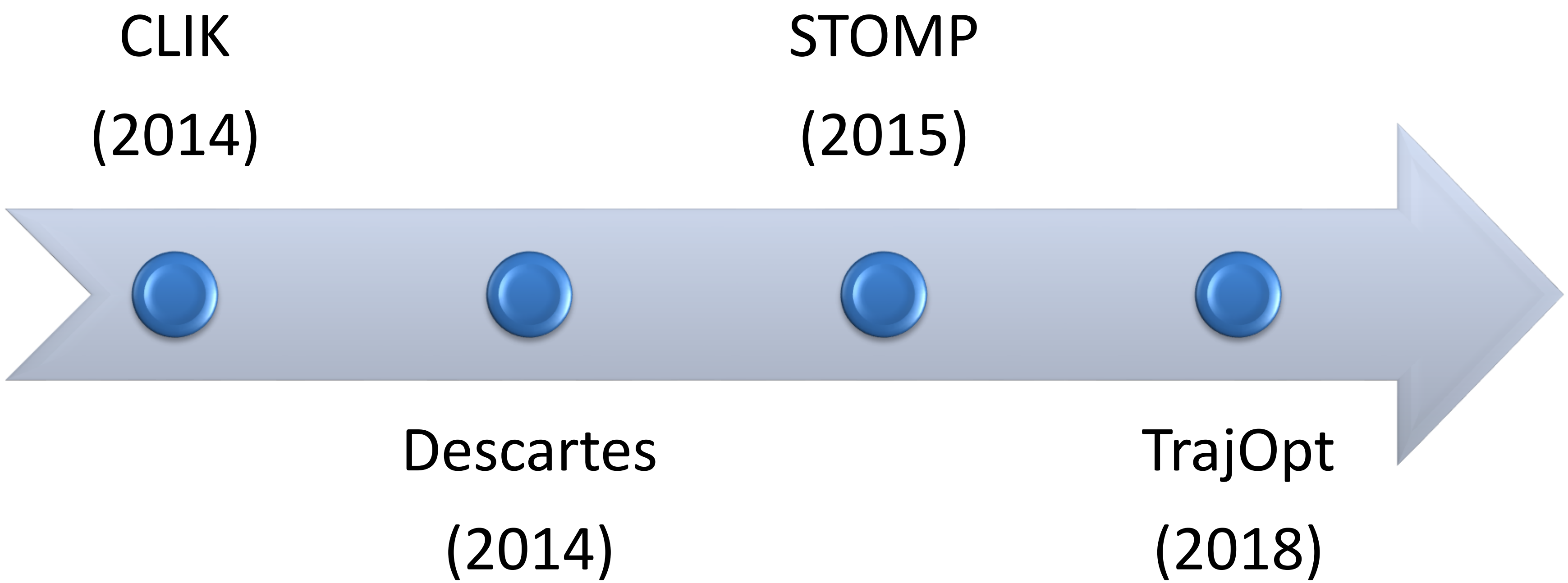
30 September, 2018

Outline

- Background
- TrajOpt
  - History
  - Motivation
  - Capabilities
  - Future
- Tesseract
  - Motivation
  - Capabilities
  - Future
- "Hybrid" Planning

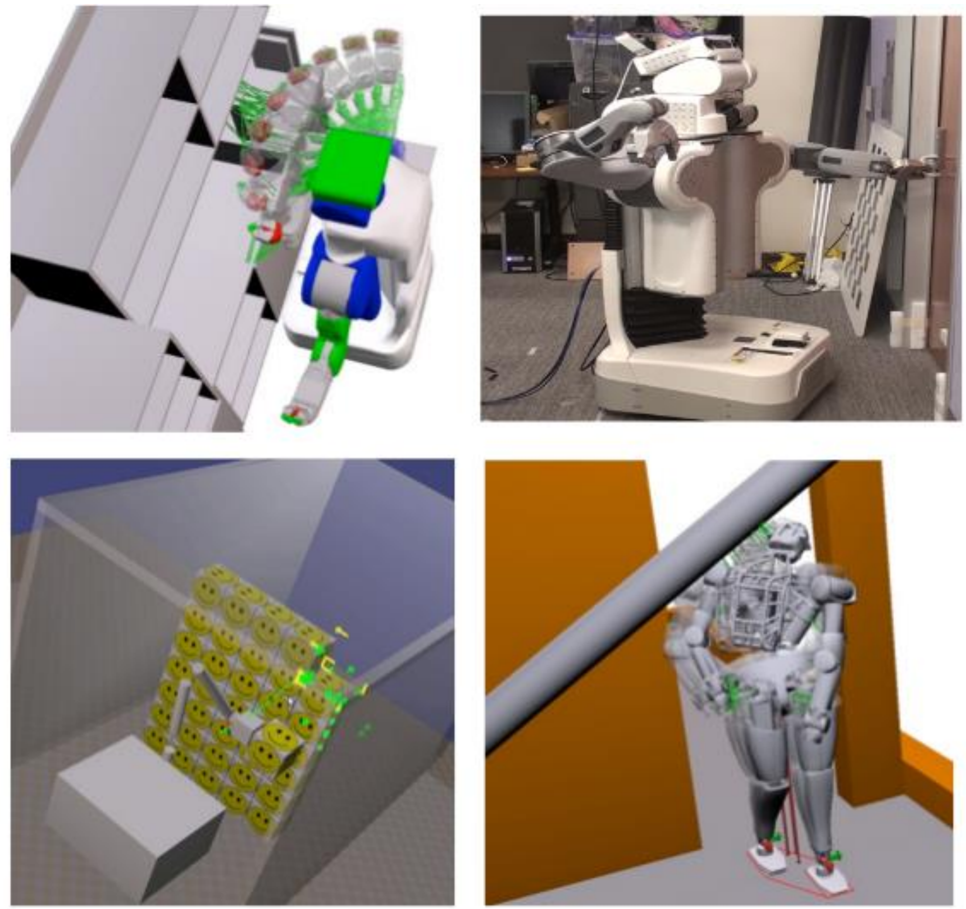


Background



TrajOpt: History

- UC Berkeley, John Shulman
- Sequential Convex Optimization
- Solves non-convex optimization problems



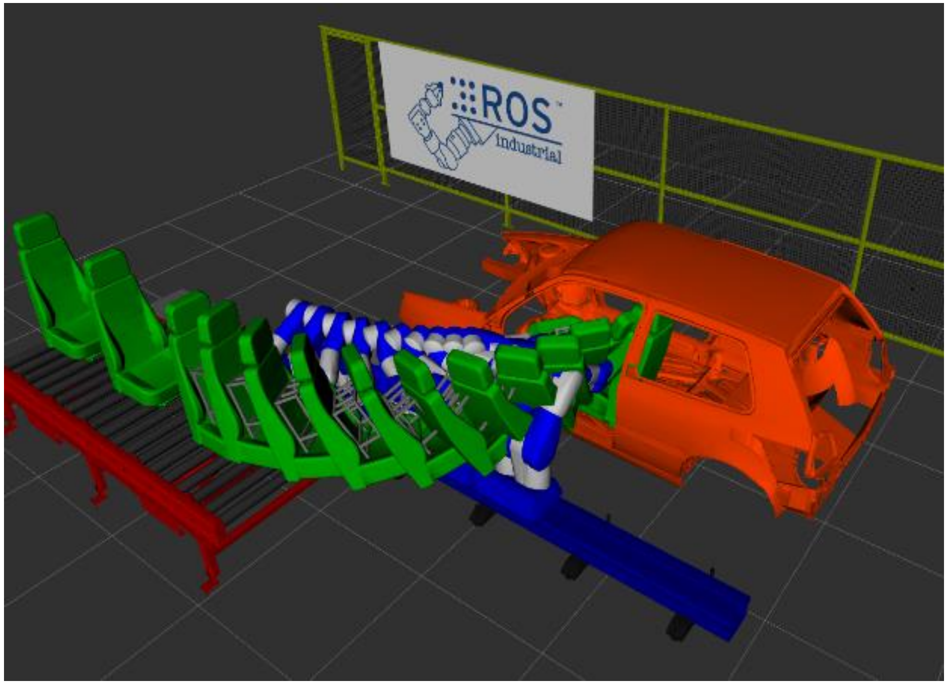
"Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization"
John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow and Pieter Abbeel

TrajOpt: History

- Southwest Research Institute launched an internal IR&D project to integrate the existing motion planner TrajOpt (Trajectory Optimization for Motion Planning) into ROS.
- The integration of TrajOpt necessitated new capabilities that spawned the creation of several new packages: ***trajopt_ros*** and ***tesseract***.

Motivation

- Current Industrial Problems
- Process Planning
  - Cartesian Planning with process constraints and leverage redundant degrees of freedom
  - Leverage secondary constraints
- Human expected paths
  - Reduce the Pucker Factor
  - Smoothness
- Improve robustness through plan reuse.
- Higher level framework for process definition



Complex Cartesian Path

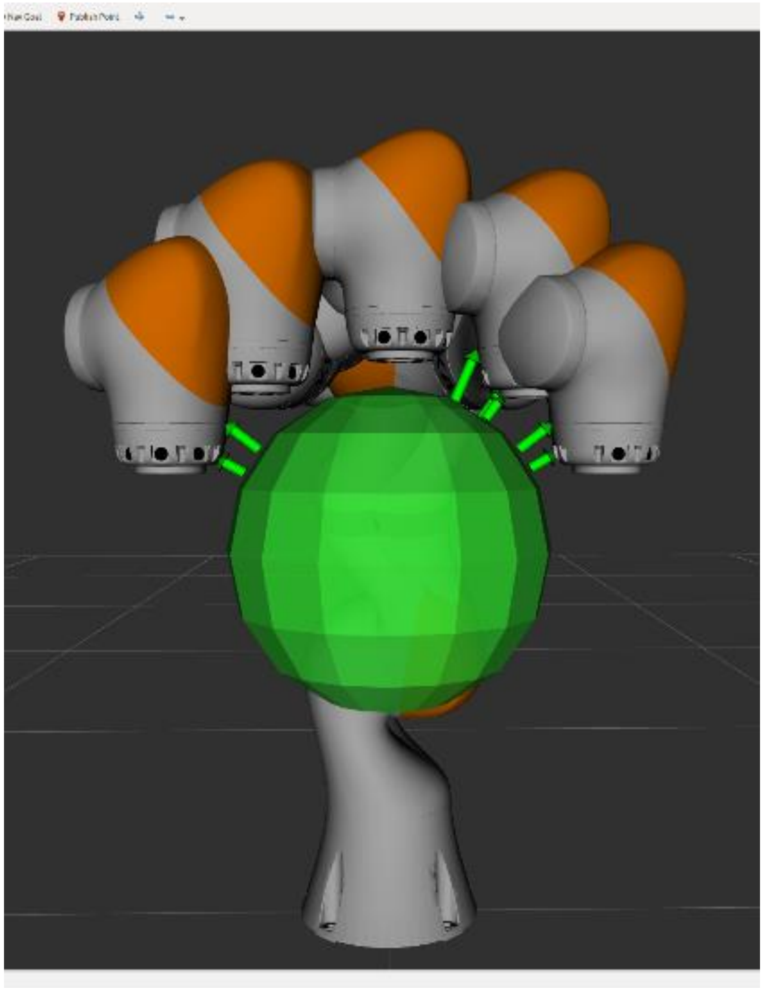
- 8DOF
  - windows\puzzle_piece.avi

Auxiliary Axis

- 10 DOF
  - windows\puzzle_piece_with_positioner.avi

TrajOpt: Conversion

- Package ***trajopt_ros***
  - Direct Port of Core
  - Swapped OSG Viewer with Rviz
  - Swapped kinematics using OpenRave with ***tesseract*** kinematics (KDL).
  - ***tesseract_collision***
  - Removed Commented Code
  - Exposed optimization parameters



TrajOpt: Capabilities

- Existing Costs/Constraints
  - Cartesian Pose
    - Semi-Constrained
    - Fully-Constrained
  - Cartesian Velocity
  - Collision
    - Discrete
    - Continuous
  - Joint Position
  - Joint Velocity
- Existing Costs/Constraints
  - Freespace motion planning
  - Cartesian motion planning
  - Combination of both Freespace and Cartesian motion planning
  - Each Cost/Constraint is time-step specific
- New
  - Joint Acceleration
  - Joint Jerk
  - Conical

Tesseract: Motivation

- TrajOpt Requires
  - Convex hull collision checking
    - Accurate minimum translation vector
  - Access to Kinematics and Dynamics
  - Parameter Limits
    - Joint Limits
- Industrial Automation Focus
  - Tool Changes
  - Multi Robot Planning
  - Scene Building
  - Kinematic chains and arbitrary trees
  - Integrated Time Parameterization

Tesseract: Motivation

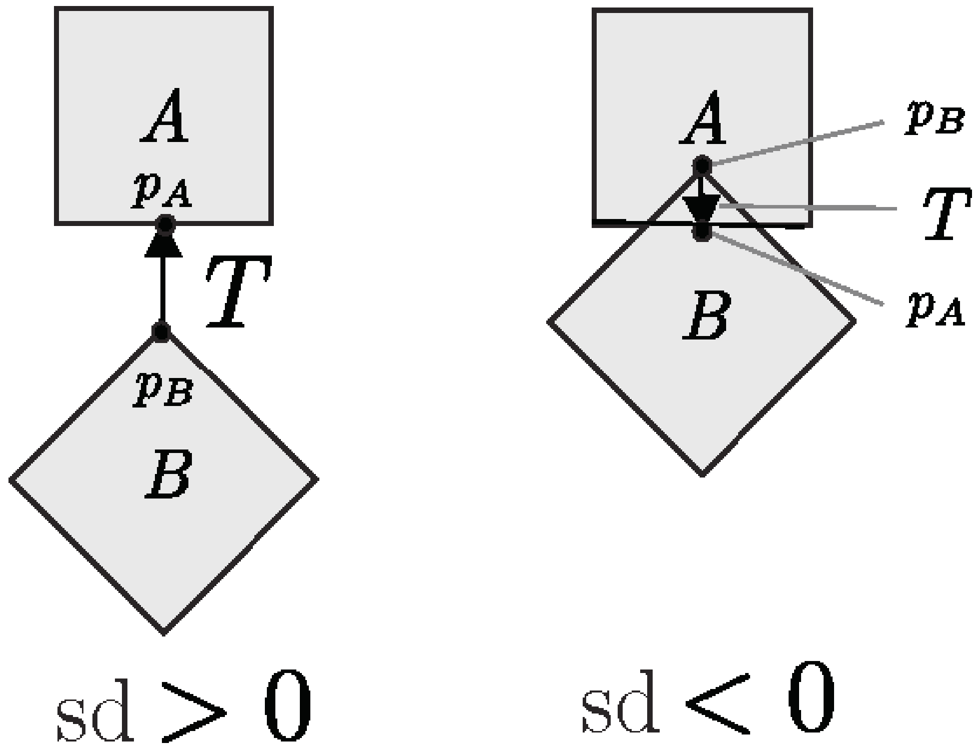
- Testbed:
  - Environment
  - Collision Checking
  - Kinematics
  - Dynamics
- Fully understand the capabilities of TrajOpt
- Single Environment
- Continuous Collision Checking

Tesseract: Capabilities

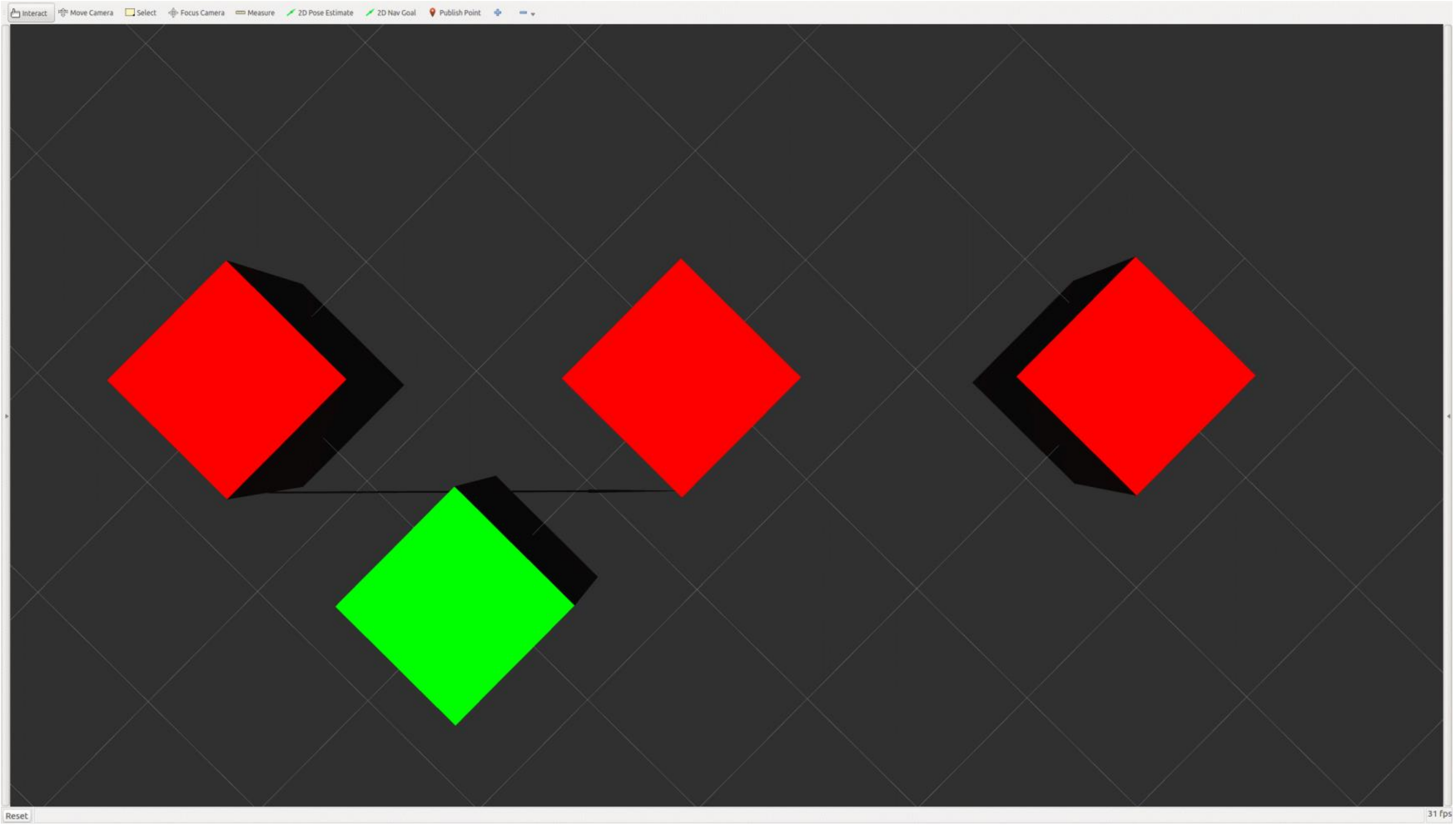
- ***tesseract_core***
  - Contains platform agnostic interfaces and data structures to be used.
- ***tesseract_ros***
  - ROS implementation of the interfaces identified in the tesseract_core package, currently leverages Orocos/KDL libraries.
- ***tesseract_collision***
  - ROS implementation of a Bullet collision library. It includes both continuous and discrete collision checking for convex-convex and convex-concave shapes.
- ***tesseract_msgs***
  - ROS message types used by Tesseract.
- ***tesseract_rviz***
  - ROS visualization plugins for Rviz for both the environment state and trajectories.
- ***tesseract_monitoring***
  - Monitoring the active environment state and publishing contact information.
  - Environment monitor (A direct copy of MoveIt! API)
- ***tesseract_planning***
  - Contains interface bridges between Tesseract Environment and motion planners OMPL and TrajOpt.

Tesseract: Collision

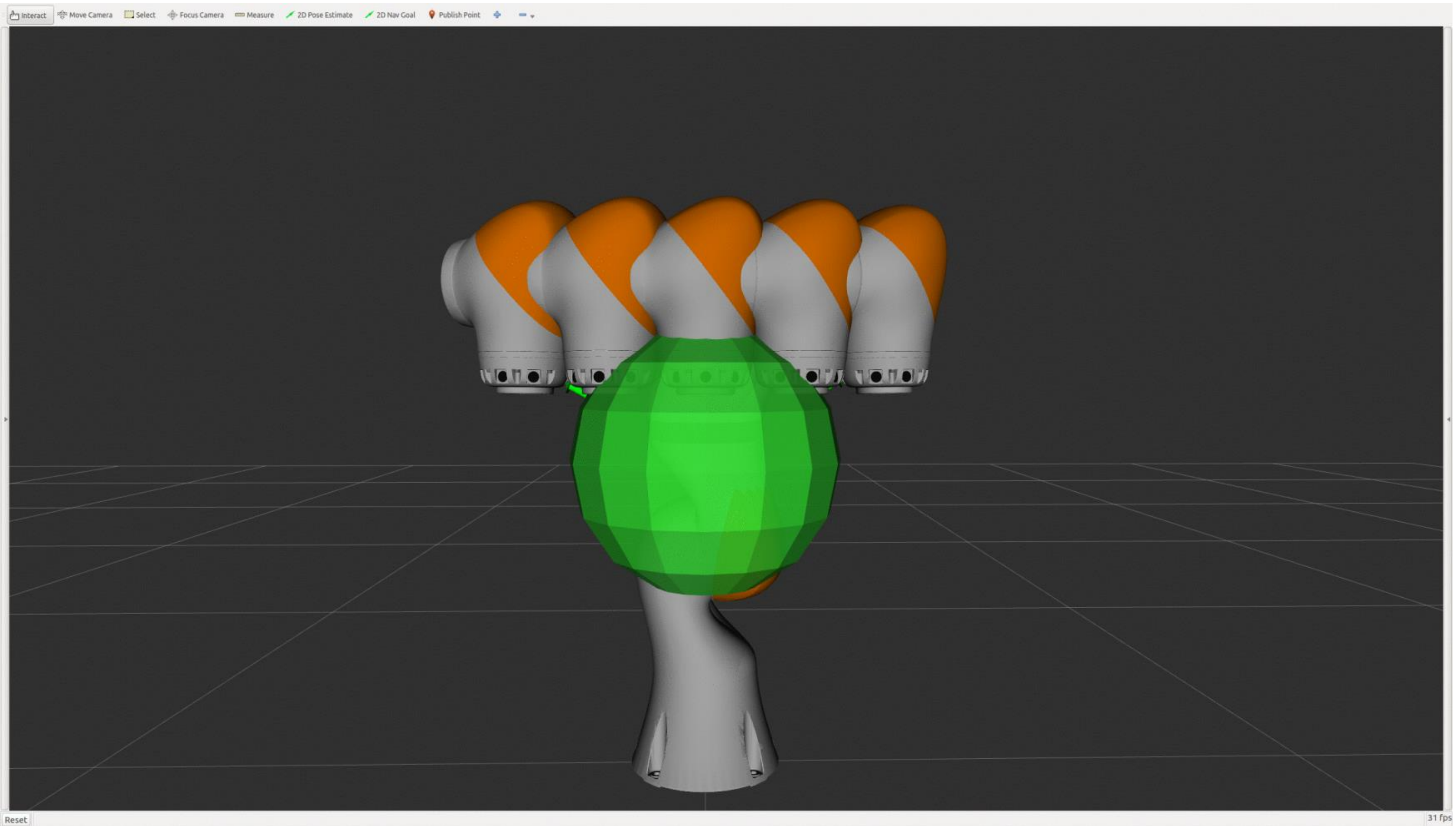
- Single Environment
- Profiled Code
  - Affine3d
  - Environment Construction for every check.
- Non-ThreadSafe
  - Clone Method
- Individual Object Distance Limits
- Implementations
  - Bullet
    - Discrete
    - Continuous (Cast)
  - FCL
    - Discrete (Experimental)



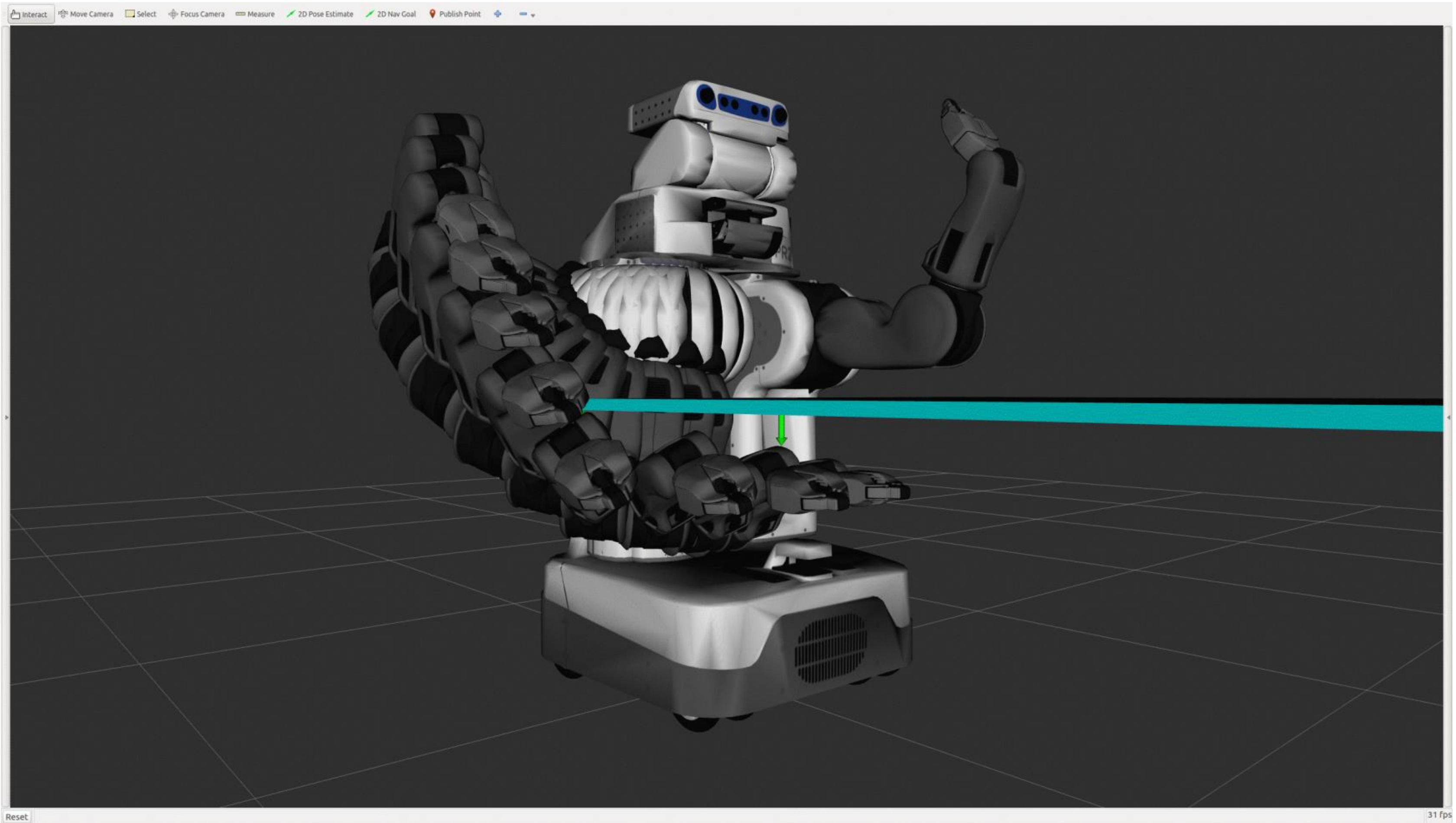
Continuous Collision Checking



Glass Up Right



Free Space


Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Reset
31 fps

Hybrid Planning

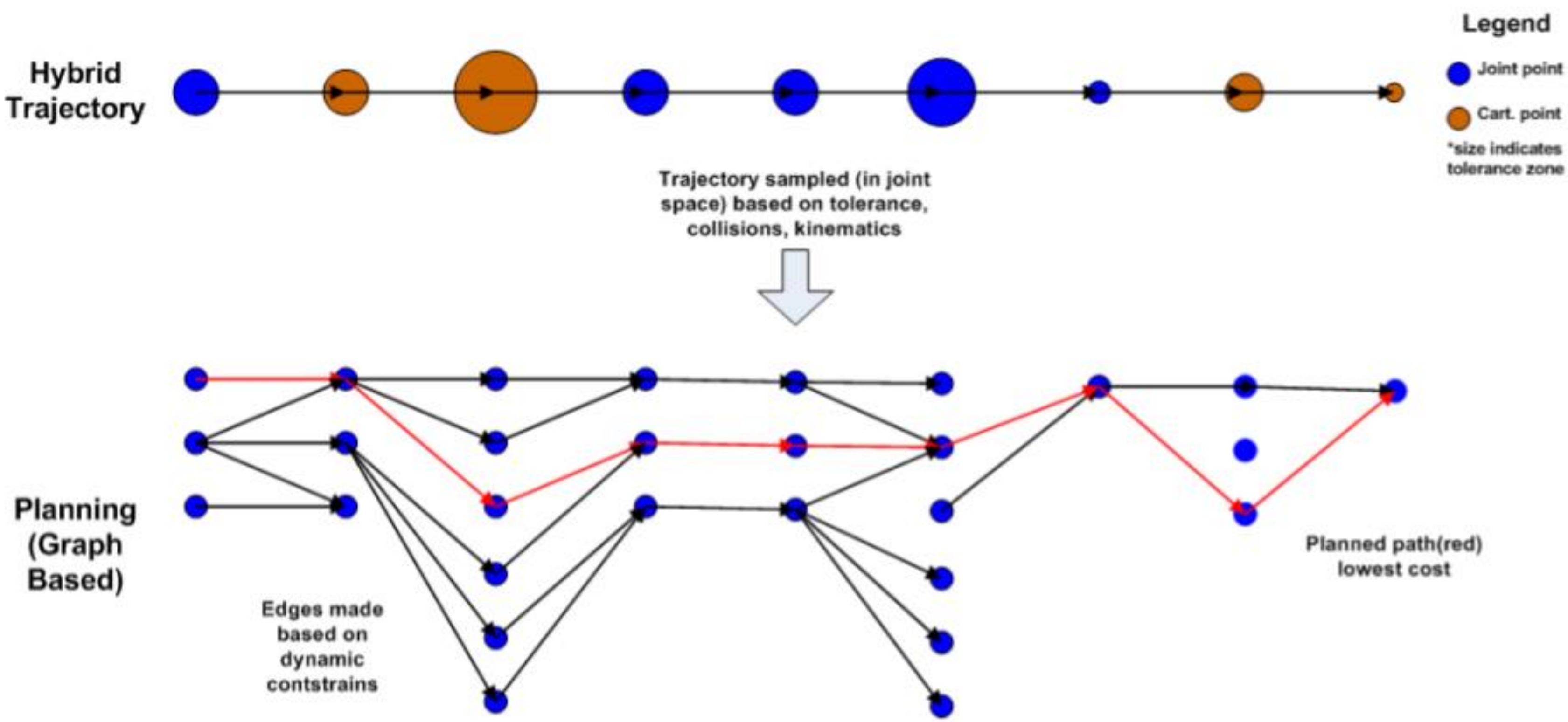
- What can we do with Trajopt/Tesseract?
  - Lots!
  - Optimization opens new doors!
- A quick note on under-constrained path planning
  - DoF Actuators > DoF of Problem

- Optimization good for:
  - High DOF
  - “Dynamic” Costs
- Where does it fall short?
  - Local minima
  - Initial condition
  - Trillions of parameters

Naïve Seed

- windows\barrel_opt.avi

Descartes



Hybrid Trajectory
Legend
Joint point
Cart. point
*size indicates tolerance zone
Trajectory sampled (in joint space) based on tolerance, collisions, kinematics
Planning (Graph Based)
Edges made based on dynamic contstrains
Planned path(red) lowest cost

Hybrid Approach

- windows\barrel_descartes_optv2.avi

Expand

- windows\gantry.avi

Questions

- Core Packages
  - https://github.com/ros-industrial-consortium/tesseract
  - https://github.com/ros-industrial-consortium/trajopt_ros
- Examples
  - https://github.com/Jmeyer1292/hybrid_planning_experiments
- Original Implementation
  - http://rll.berkeley.edu/trajopt/doc/sphinx_build/html/

Contact Information

Levi Armstrong

Group Leader

Southwest Research Institute
6220 Culebra Road
San Antonio, Texas 78238

Phone: (210) 522-3801
Email: levi.armstrong@swri.org

www.swri.org

www.ros-i.org

Contact Information

Jonathan Meyer

Github: @jmeyer1292

Email: jmeyer1292@gmail.com