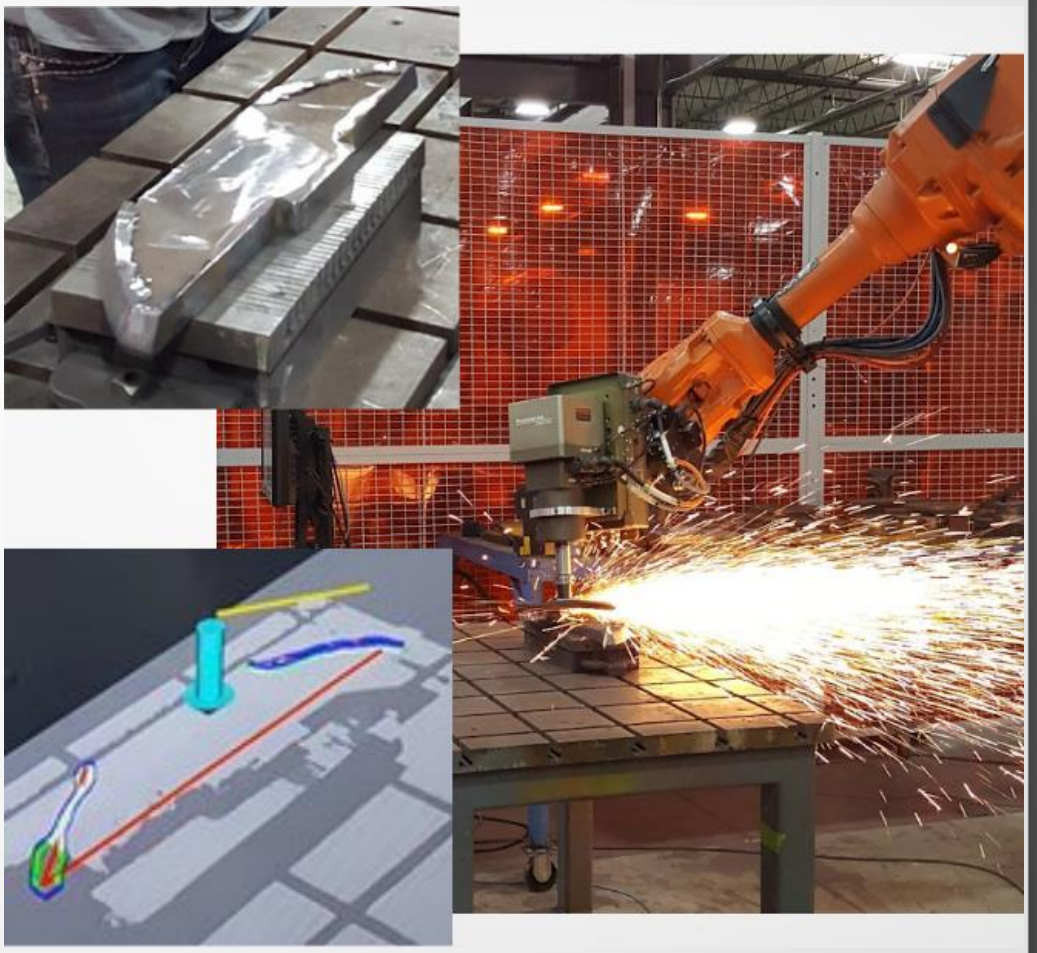
Tesseract Update

- Overview
- Recent Development
  - Leverage Plugins
  - Task Composer
  - Tesseract Qt
- Future/Current Development
  - Tesseract Planning Refactoring
  - Tesseract Qt (Sensors, Physics, etc.)

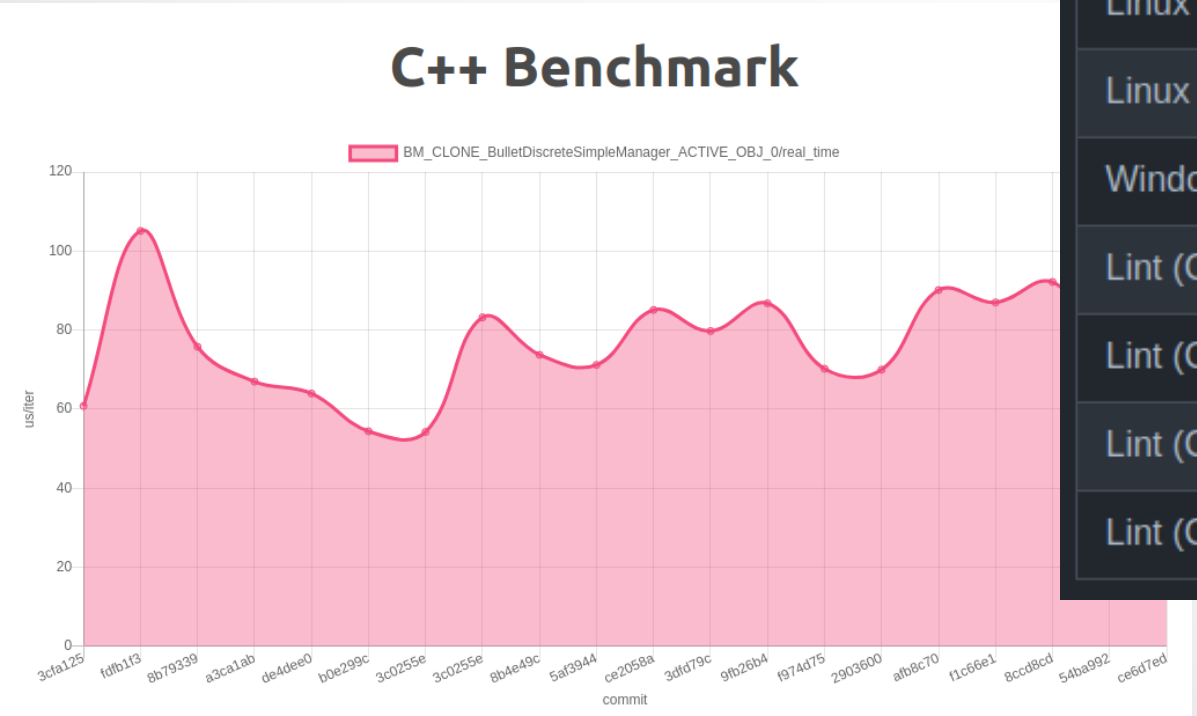
Tesseract Overview

- Tesseract is developed to meet industrial automation needs around functionality, quality, performance, reliability and traceability.
- Repositories
  - tesseract
  - tesseract_planning
  - tesseract_qt
  - tesseract_ros
  - tesseract_ros2
  - tesseract_python
- https://github.com/tesseract-robotics



Quality & Performance

- Tools leveraged
  - Clang-Tidy & CppCheck
  - Google Test & Benchmarks
  - Code Coverage
    - Goal > 90%



codecov 90%

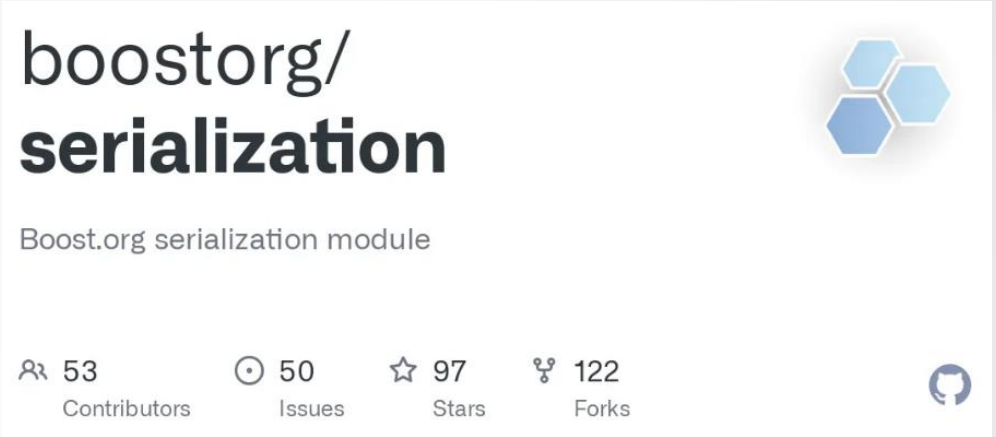
python 3.7+

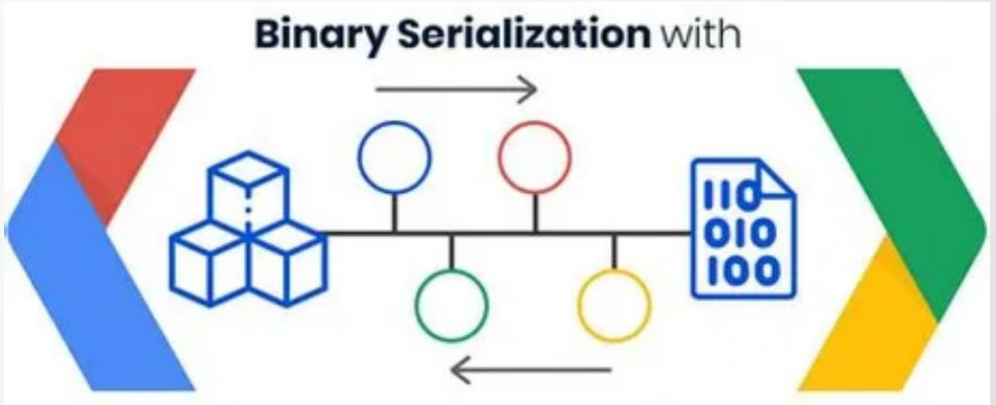
pypi v0.1.8

Platform	CI Status
Linux (Focal)	Focal-Build passing
Linux (Bionic)	Bionic-Build passing
Windows	Windows-Noetic-Build passing
Lint (Clang-Format)	Clang-Format passing
Lint (CMake-Format)	CMake-Format passing
Lint (Clang-Tidy)	Clang-Tidy passing
Lint (CodeCov)	CodeCov passing

Traceability

- On deployed systems you need a reliable way to troubleshoot the system when an issue is reported.
- In order to address this Tesseract leverages
  - Boost Serialization
  - Design Features
    - Environment Command History
    - Task Composer Info
    - Etc.
  - These allow taking snapshots of the system to enable full reconstruction offline for trouble shooting





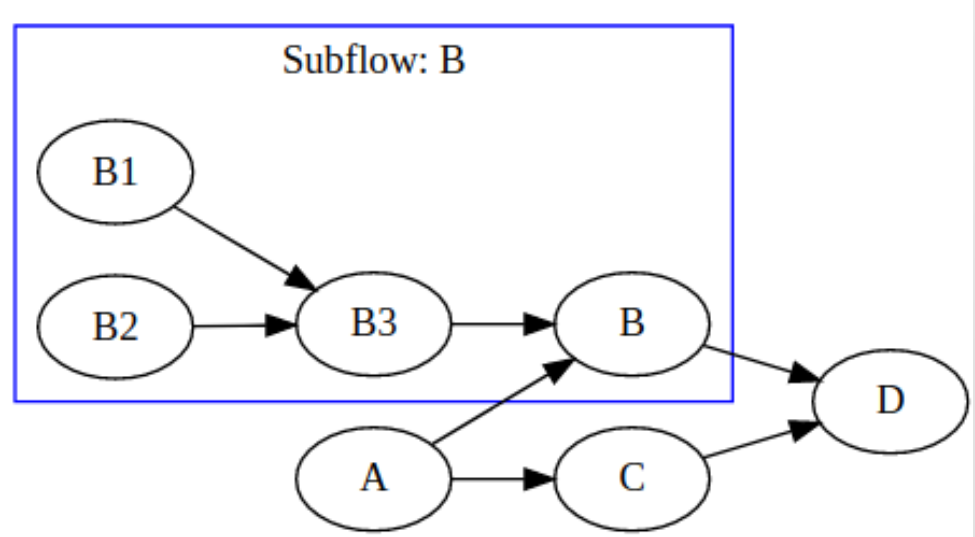
Plugins

- Current
  - Tesseract Kinematics
    - Forward
    - Inverse
  - Tesseract Collision
    - Discrete
    - Continuous
- Future
  - Task Composer Graph
  - Task Composer Node
  - Task Composer Executor
  - Profiles
  - Motion Planners

Task Composer

- A generic interface for create and composing task graphs.
- Task Composer Node
  - Task (Preforms Work)
  - Graph
- Task Composer Executor
  - Process Node
  - Default Implementation: Taskflow
- Task Composer Data Storage
  - Can hold anything!
- Full Boost Serialization Support

- Task/Graphs Implementations
  - Motions Planning
  - Time Parameterization
  - Raster Planning
  - Collision Checking
  - Etc.

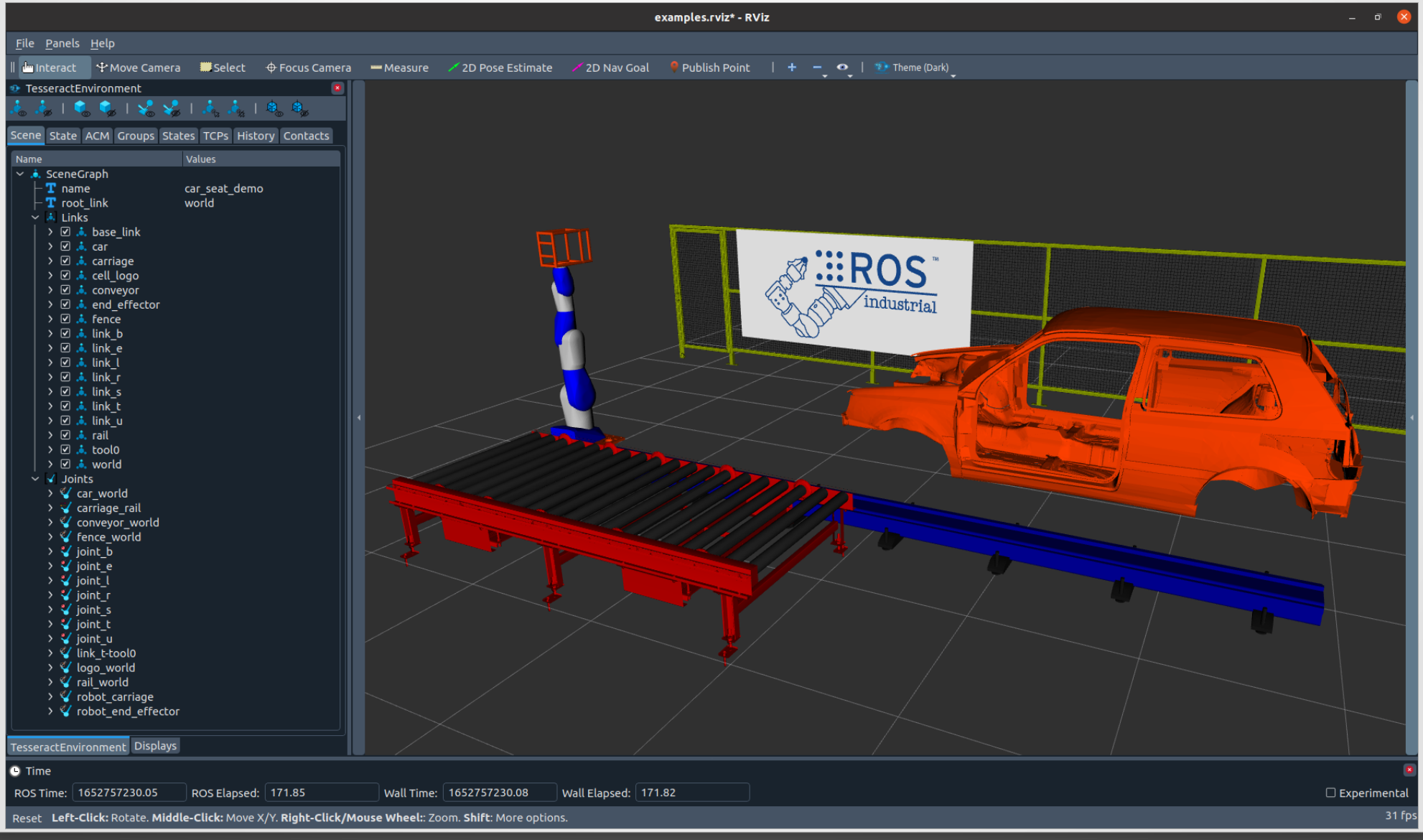


Tesseract Qt

Tesseract Qt

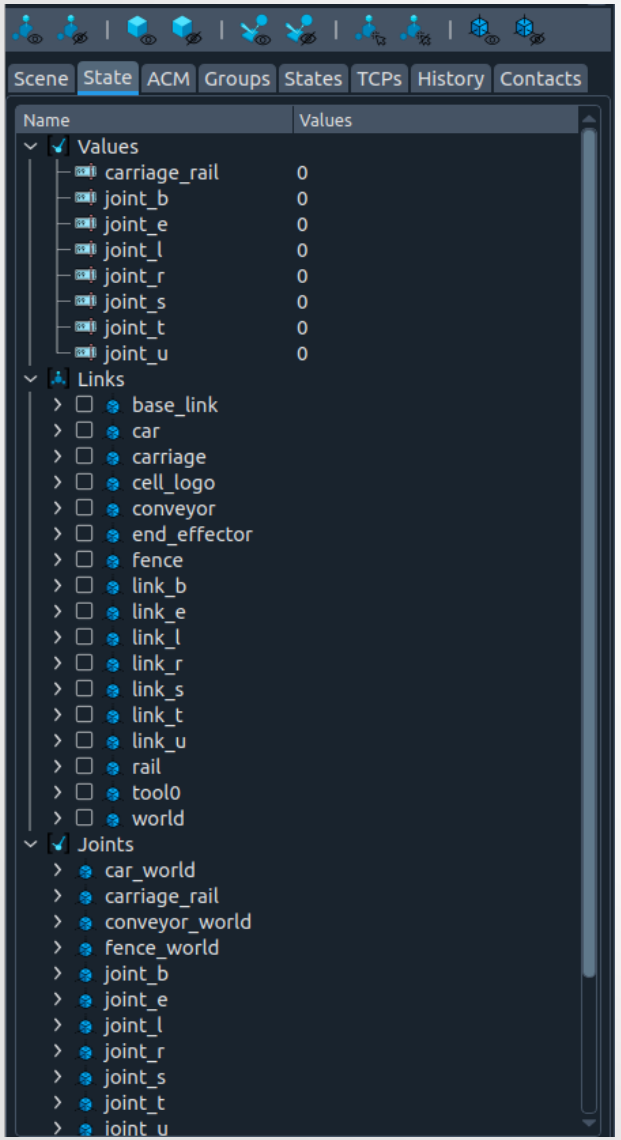
- A collection of Qt Widgets and Data Models for a variety of Tesseract Components (All Under Active Development)
  - Scene Graph
  - Scene State
  - Allowed Collision Matrix
  - Kinematic Groups
  - Kinematic States
  - Tool Center Points
  - Command History
  - Manipulation
  - Joint Trajectory
  - Tool Path
  - Rendering
  - SRDF Editor

Scene Graph Widget

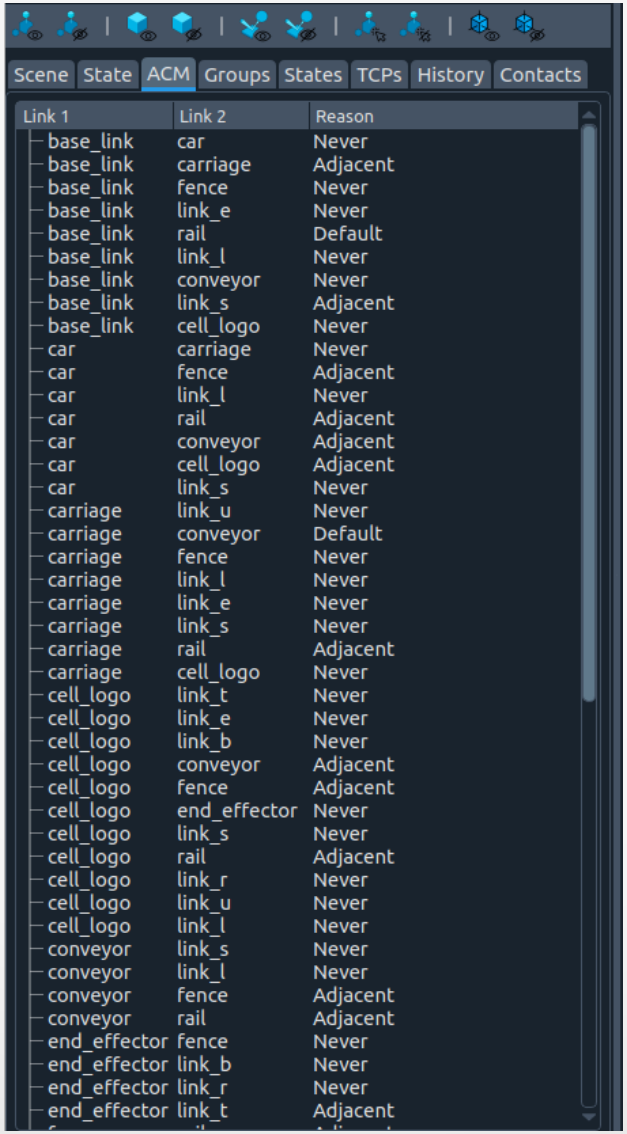


Additional Widgets

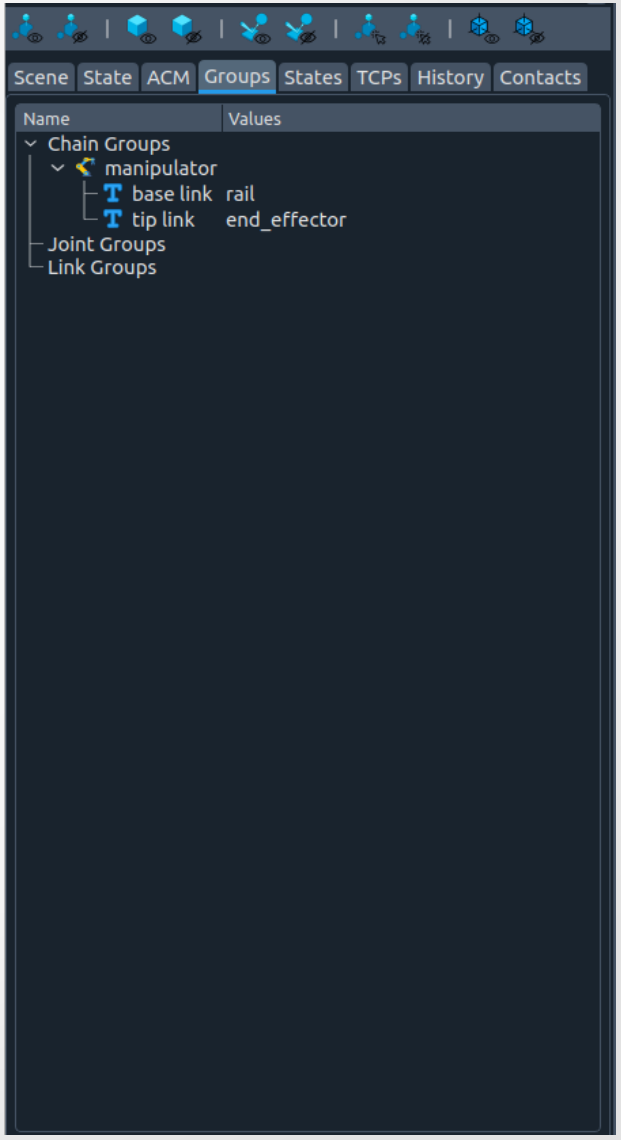
Scene State Widget



ACM Widget

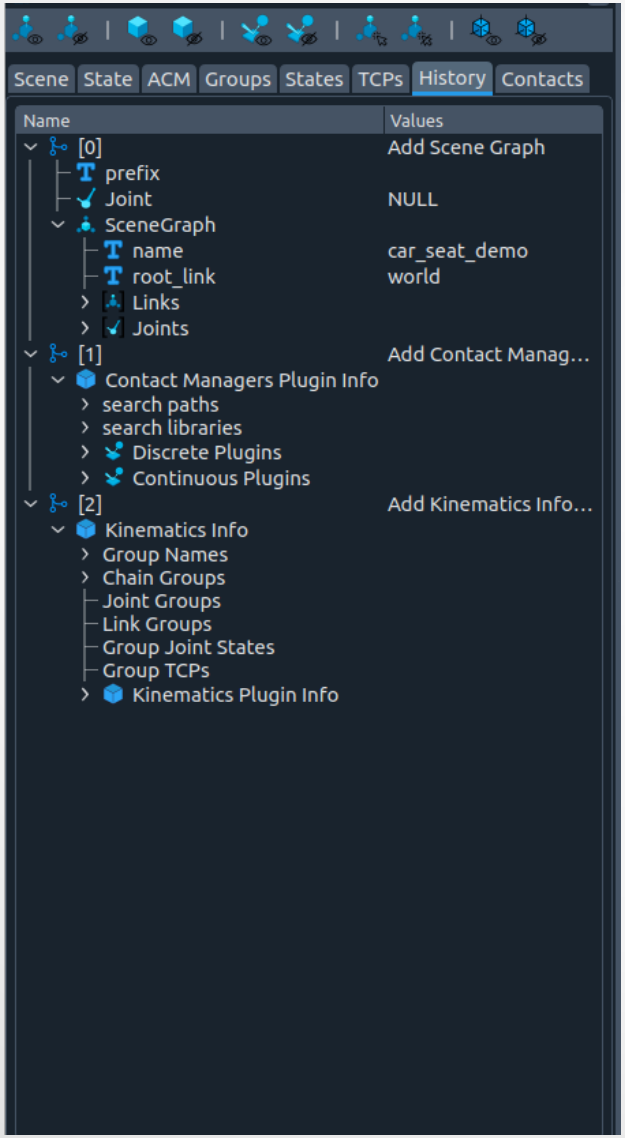


Kinematic Groups Widget

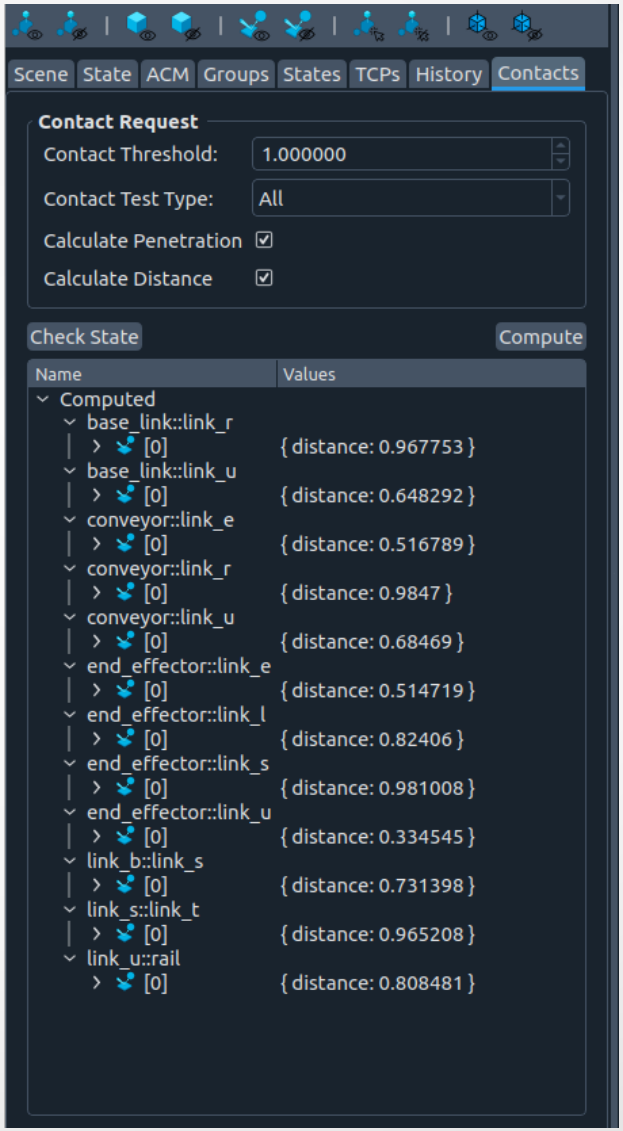


Additional Widgets

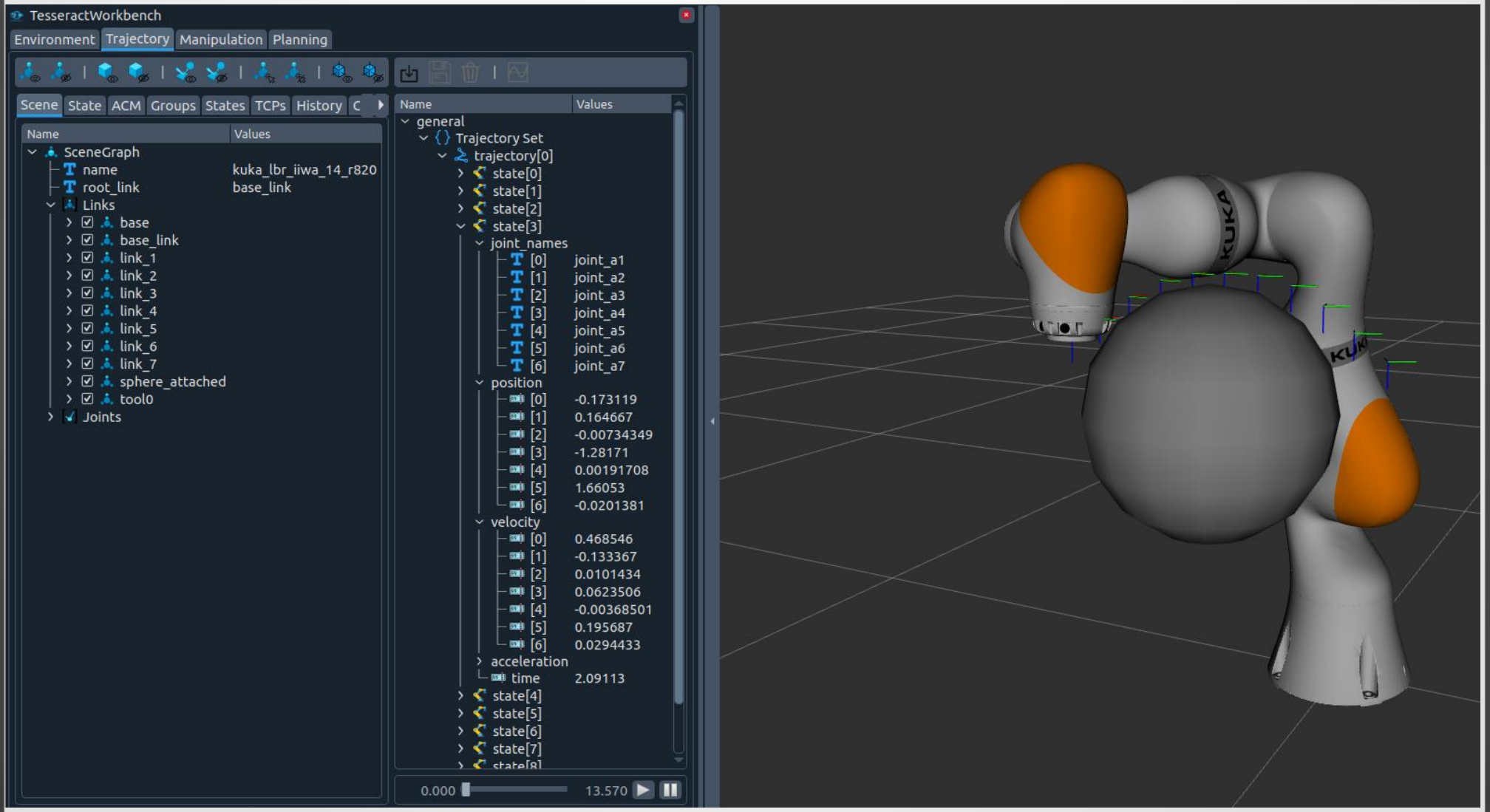
Command History Widget



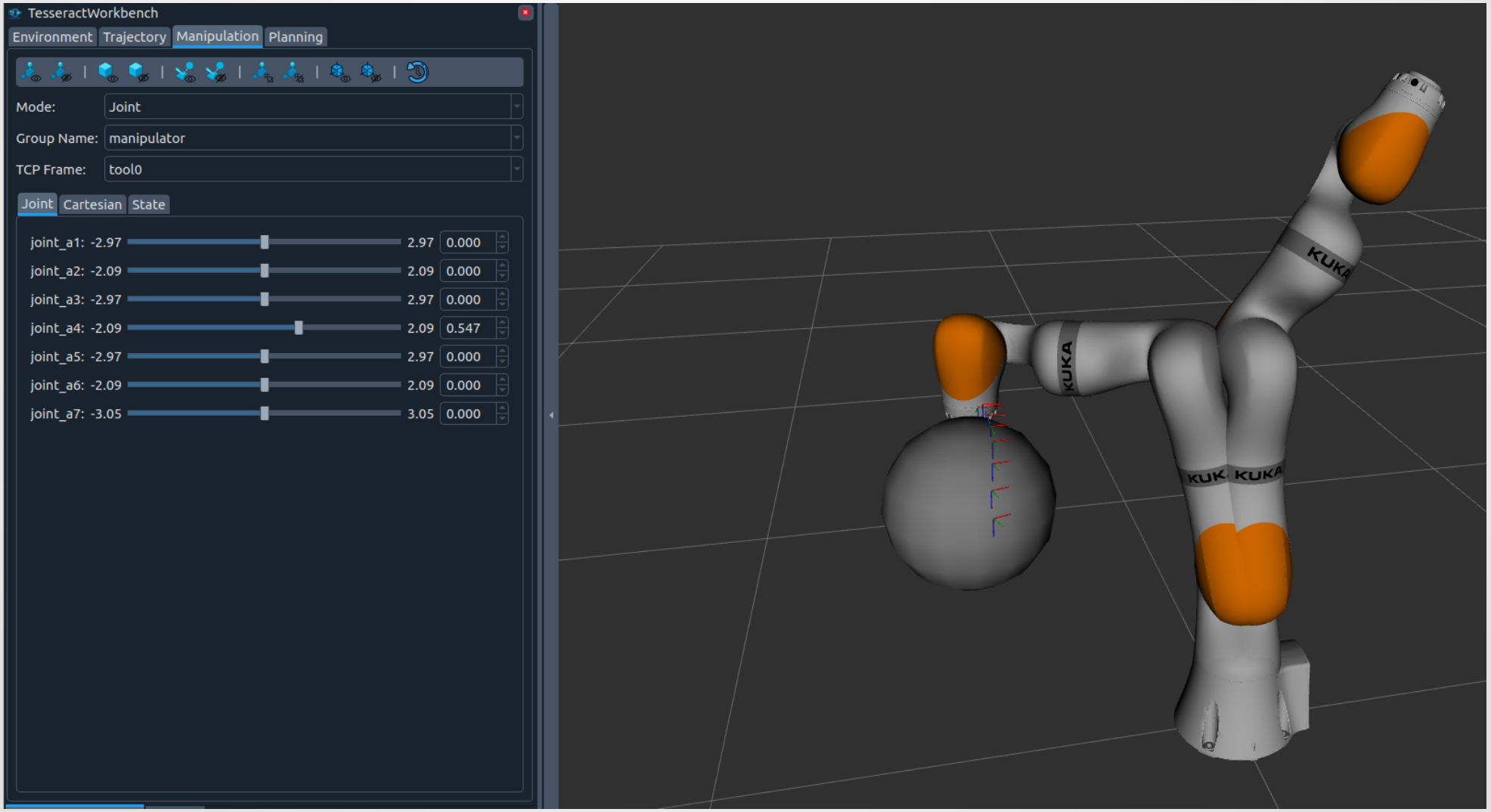
Contact Results Widget



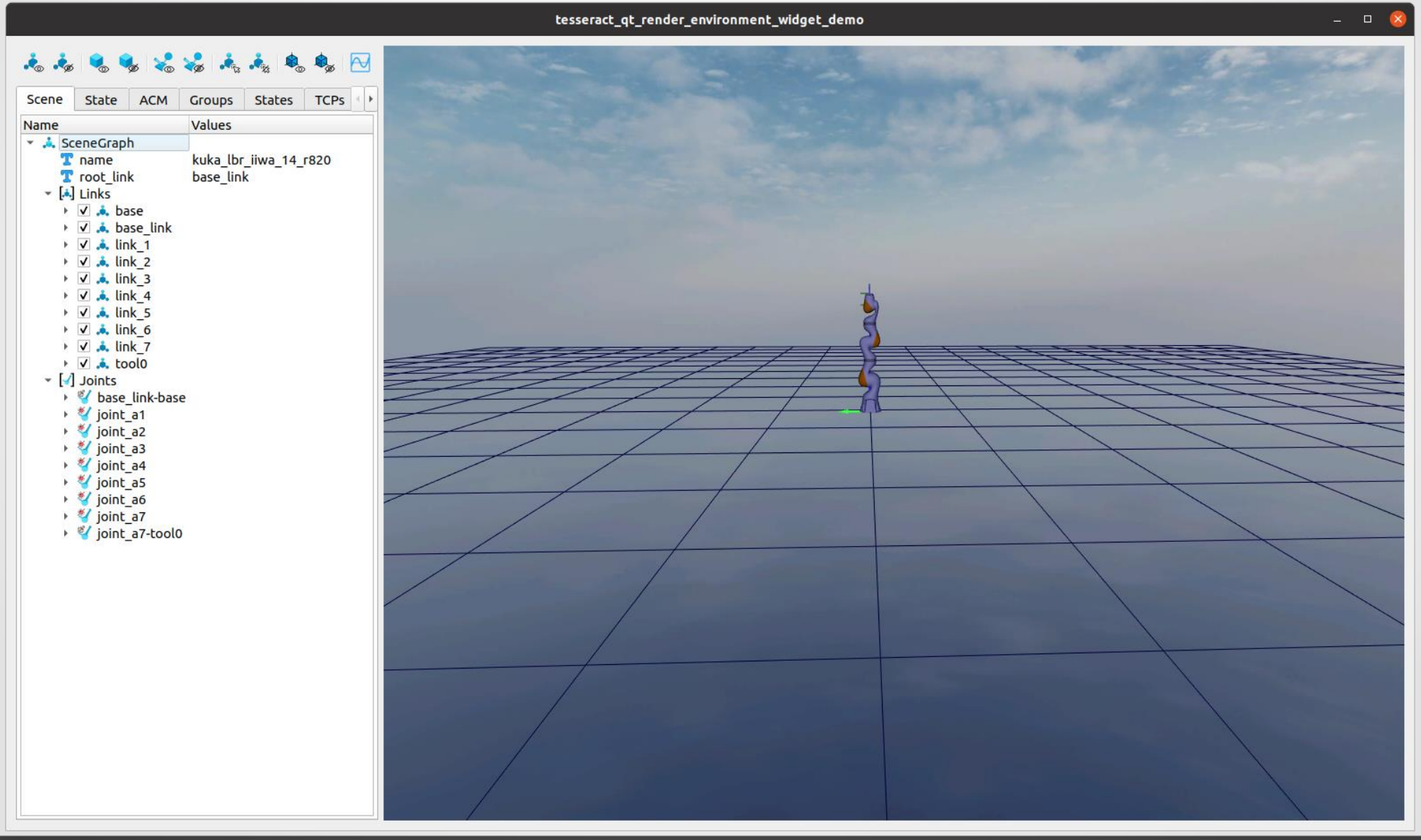
Joint Trajectory Widget



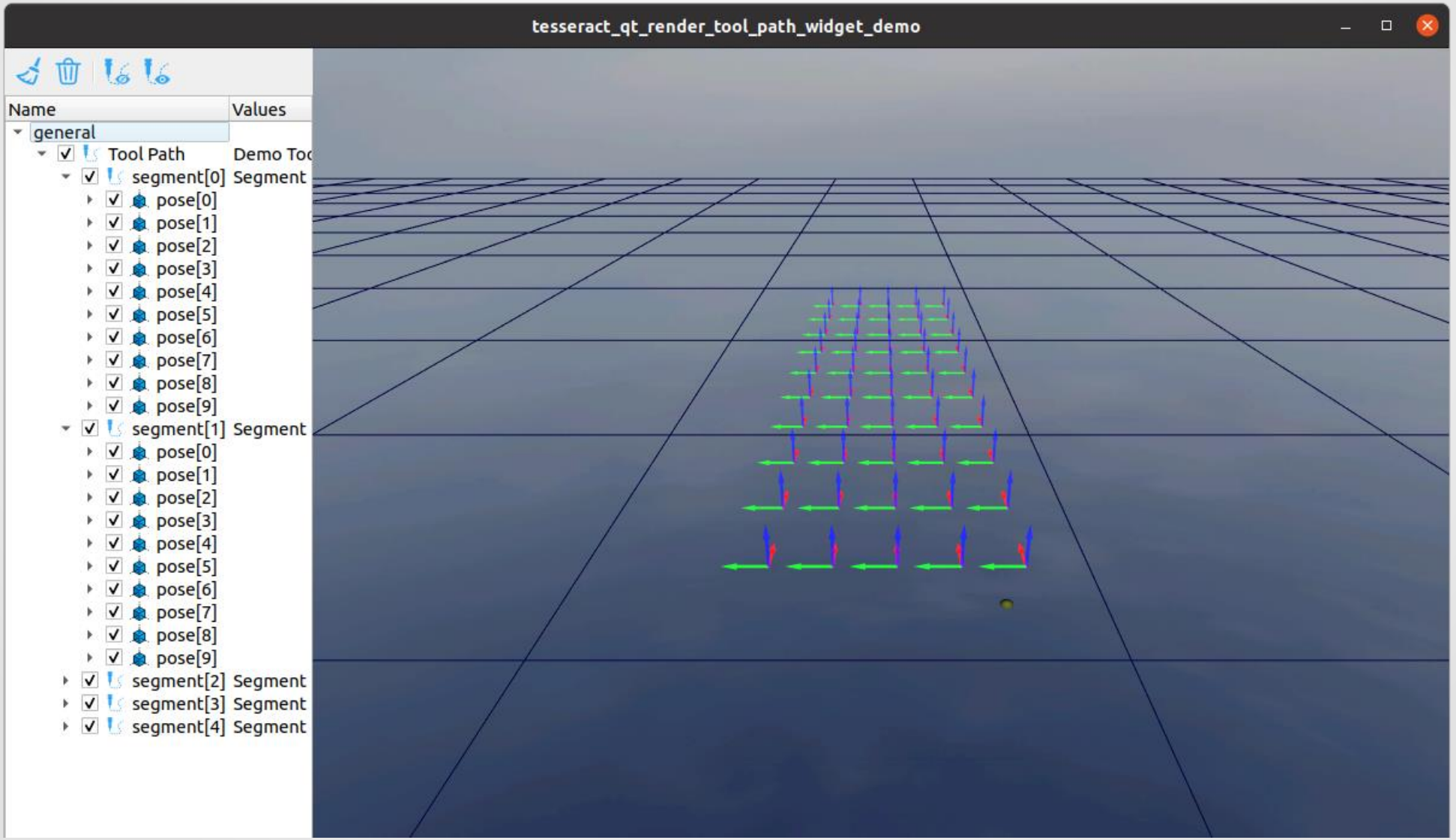
Manipulation Widget



Render Widget, Leverages Ignition



Tool Path Widget



Tesseract Planning Refactoring

Michael Ripperger
Levi Armstrong

Overview

- Challenges
  - Planning profiles do not share a common interface
  - Profile dictionary is type-specific, so a user always has to know what type a profile is to add it to the dictionary
  - Must build tesseract_planning from source to introduce custom profiles, motion planners, task graphs, etc.
- Changes
  - Create common planner profile interface
  - Leverage Plugins to allow users to create custom profiles more easily
  - Leverage Plugins to allow user to create custom Task Composer Graphs
  - Add serialization support

Conclusion

- 1.0 release in next six months
  - Tesseract Planning Refactor
  - Full Boost Serialization
  - Full Plugin Support
  - Code Coverage > 90%
    - tesseract
      - Complete
    - tesseract_planning
      - In-Work
  - Complete SRDF Editor
- Extracting functionality that applies to broader community
  - boost_plugin_loader