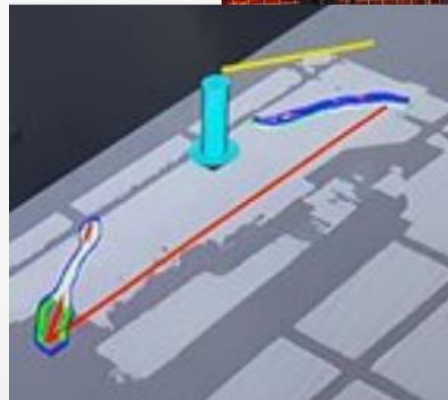
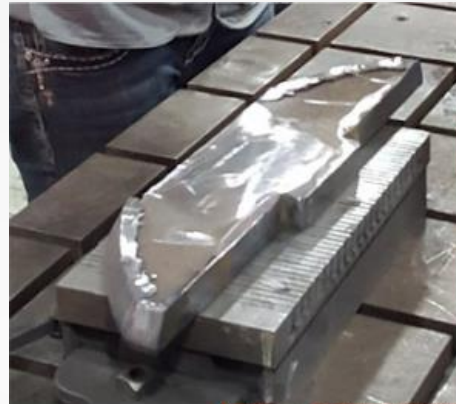


# Tesseract Update

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

# Tesseract Overview

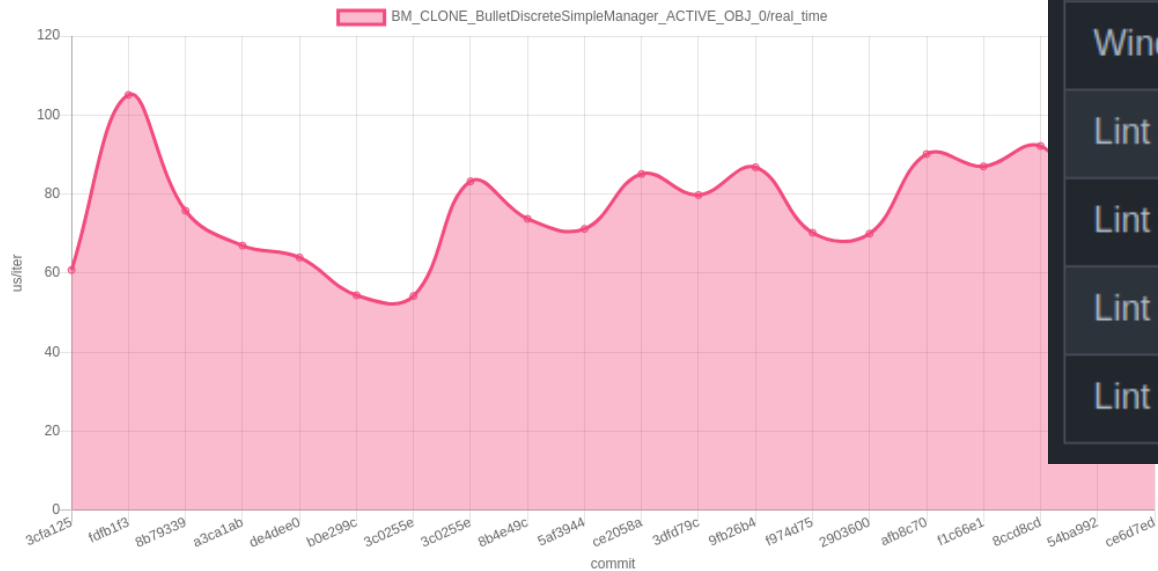
- Tesseract is developed to meet industrial automation needs around functionality, quality, performance, reliability and traceability.
- Repositories
  - tesseract
  - tesseract\_planning
  - tesseract Qt
  - tesseract\_ros
  - tesseract\_ros2
  - tesseract\_python
- <https://github.com/tesseract-robotics>



# Quality & Performance

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

## C++ Benchmark



codecov	90%
python	3.7+
pypi	v0.1.8
Platform	CI Status
Linux (Focal)	Focal-Build <span>passing</span>
Linux (Bionic)	Bionic-Build <span>passing</span>
Windows	Windows-Noetic-Build <span>passing</span>
Lint (Clang-Format)	Clang-Format <span>passing</span>
Lint (CMake-Format)	CMake-Format <span>passing</span>
Lint (Clang-Tidy)	Clang-Tidy <span>passing</span>
Lint (CodeCov)	CodeCov <span>passing</span>

# 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

## boostorg/ serialization



Boost.org serialization module

53

Contributors

50

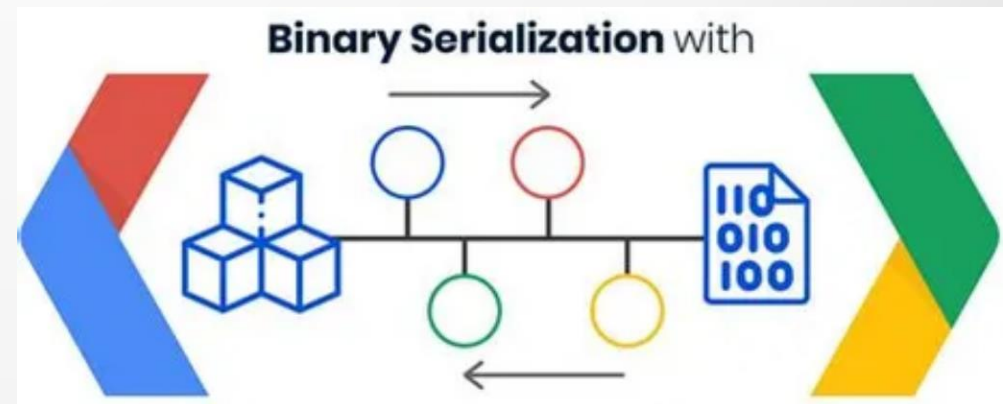
Issues

97

Stars

122

Forks



# Plugins

- Current

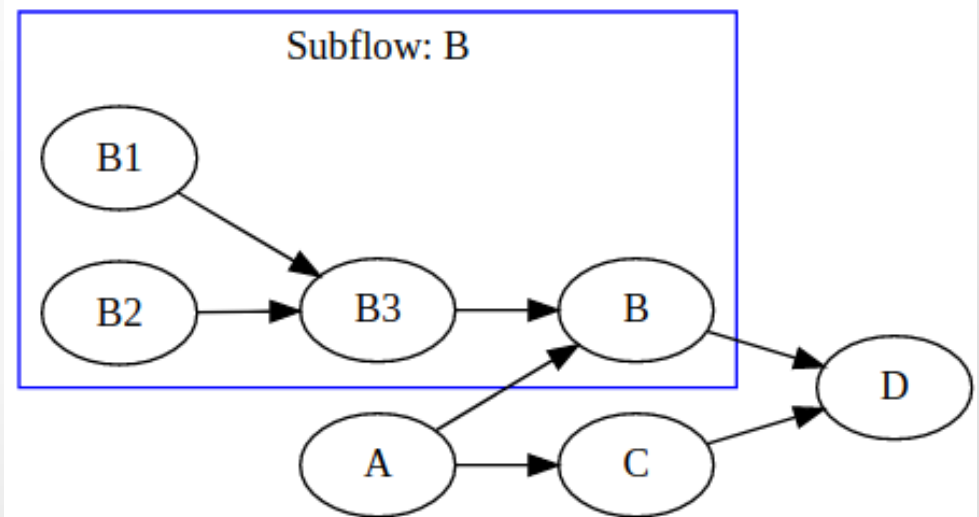
- Tesseract Kinematics
  - Forward
  - Inverse
- Tesseract Collision
  - Discrete
  - Continuous

- 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

The screenshot displays the RViz interface for a simulation titled "examples.rviz\* - RViz". The main window shows a 3D scene with a red conveyor belt, a blue robot arm, and an orange car body. A yellow fence in the background features a "ROS industrial" logo. The left sidebar contains the "Scene Graph" widget, which lists the scene's hierarchical structure:

Name	Values
SceneGraph	
name	car_seat_demo
root_link	world
Links	
base_link	
car	
carriage	
cell_logo	
conveyor	
end_effector	
fence	
link_b	
link_e	
link_l	
link_r	
link_s	
link_t	
link_u	
rail	
tool0	
world	
Joints	
car_world	
carriage_rail	
conveyor_world	
fence_world	
joint_b	
joint_e	
joint_l	
joint_r	
joint_s	
joint_t	
joint_u	
link_t-tool0	
logo_world	
rail_world	
robot_carriage	
robot_end_effector	

At the bottom of the interface, the "Time" panel shows the following data:

ROS Time:	ROS Elapsed:	Wall Time:	Wall Elapsed:
1652757230.05	171.85	1652757230.08	171.82

Additional information includes a "Reset" button, a legend for mouse actions (Left-Click: Rotate, Middle-Click: Move X/Y, Right-Click/Mouse Wheel: Zoom, Shift: More options), an "Experimental" checkbox, and a frame rate of "31 fps".

# Additional Widgets

## Scene State Widget

The Scene State Widget displays a tree view of scene elements and their values. The interface includes a toolbar with icons for Scene, State, ACM, Groups, States, TCPs, History, and Contacts. The main area is divided into two columns: Name and Values.

Name	Values
Values	
carriage_rail	0
joint_b	0
joint_e	0
joint_l	0
joint_r	0
joint_s	0
joint_t	0
joint_u	0
Links	
base_link	
car	
carriage	
cell_logo	
conveyor	
end_effector	
fence	
link_b	
link_e	
link_l	
link_r	
link_s	
link_t	
link_u	
rail	
tool0	
world	
Joints	
car_world	
carriage_rail	
conveyor_world	
fence_world	
joint_b	
joint_e	
joint_l	
joint_r	
joint_s	
joint_t	
joint_u	

## ACM Widget

The ACM Widget displays a table of link relationships and reasons. The interface includes a toolbar with icons for Scene, State, ACM, Groups, States, TCPs, History, and Contacts. The main area is divided into three columns: Link 1, Link 2, and Reason.

Link 1	Link 2	Reason
base_link	car	Never
base_link	carriage	Adjacent
base_link	fence	Never
base_link	link_e	Never
base_link	rail	Default
base_link	link_l	Never
base_link	conveyor	Never
base_link	link_s	Adjacent
base_link	cell_logo	Never
car	carriage	Never
car	fence	Adjacent
car	link_l	Never
car	rail	Adjacent
car	conveyor	Adjacent
car	cell_logo	Adjacent
car	link_s	Never
carriage	link_u	Never
carriage	conveyor	Default
carriage	fence	Never
carriage	link_l	Never
carriage	link_e	Never
carriage	link_s	Never
carriage	rail	Adjacent
carriage	cell_logo	Never
cell_logo	link_t	Never
cell_logo	link_e	Never
cell_logo	link_b	Never
cell_logo	conveyor	Adjacent
cell_logo	fence	Adjacent
cell_logo	end_effector	Never
cell_logo	link_s	Never
cell_logo	rail	Adjacent
cell_logo	link_r	Never
cell_logo	link_u	Never
cell_logo	link_l	Never
conveyor	link_s	Never
conveyor	link_l	Never
conveyor	fence	Adjacent
conveyor	rail	Adjacent
end_effector	fence	Never
end_effector	link_b	Never
end_effector	link_r	Never
end_effector	link_t	Adjacent

## Kinematic Groups Widget

The Kinematic Groups Widget displays a tree view of kinematic groups. The interface includes a toolbar with icons for Scene, State, ACM, Groups, States, TCPs, History, and Contacts. The main area is divided into two columns: Name and Values.

Name	Values
Chain Groups	
manipulator	
base link	rail
tip link	end_effector
Joint Groups	
Link Groups	

# Additional Widgets

## Command History Widget

The Command History Widget displays a list of commands executed in the software. The interface includes a menu bar with options: Scene, State, ACM, Groups, States, TCPs, History (selected), and Contacts. The main area is a table with two columns: Name and Values.

Name	Values
[0]	Add Scene Graph
prefix	
Joint	NULL
SceneGraph	
name	car_seat_demo
root_link	world
Links	
Joints	
[1]	Add Contact Manag...
Contact Managers Plugin Info	
search paths	
search libraries	
Discrete Plugins	
Continuous Plugins	
[2]	Add Kinematics Info...
Kinematics Info	
Group Names	
Chain Groups	
Joint Groups	
Link Groups	
Group Joint States	
Group TCPs	
Kinematics Plugin Info	

## Contact Results Widget

The Contact Results Widget displays contact request parameters and results. The interface includes a menu bar with options: Scene, State, ACM, Groups, States, TCPs, History, and Contacts (selected). The main area is divided into two sections: Contact Request and Check State.

**Contact Request**

Contact Threshold: 1.000000  
Contact Test Type: All  
Calculate Penetration:   
Calculate Distance:

**Check State** Compute

Name	Values
Computed	
base_link::link_r	
[0]	{ distance: 0.967753 }
base_link::link_u	
[0]	{ distance: 0.648292 }
conveyor::link_e	
[0]	{ distance: 0.516789 }
conveyor::link_r	
[0]	{ distance: 0.9847 }
conveyor::link_u	
[0]	{ distance: 0.68469 }
end_effector::link_e	
[0]	{ distance: 0.514719 }
end_effector::link_l	
[0]	{ distance: 0.82406 }
end_effector::link_s	
[0]	{ distance: 0.981008 }
end_effector::link_u	
[0]	{ distance: 0.334545 }
link_b::link_s	
[0]	{ distance: 0.731398 }
link_s::link_t	
[0]	{ distance: 0.965208 }
link_u::rail	
[0]	{ distance: 0.808481 }

# Joint Trajectory Widget

TesseractWorkbench

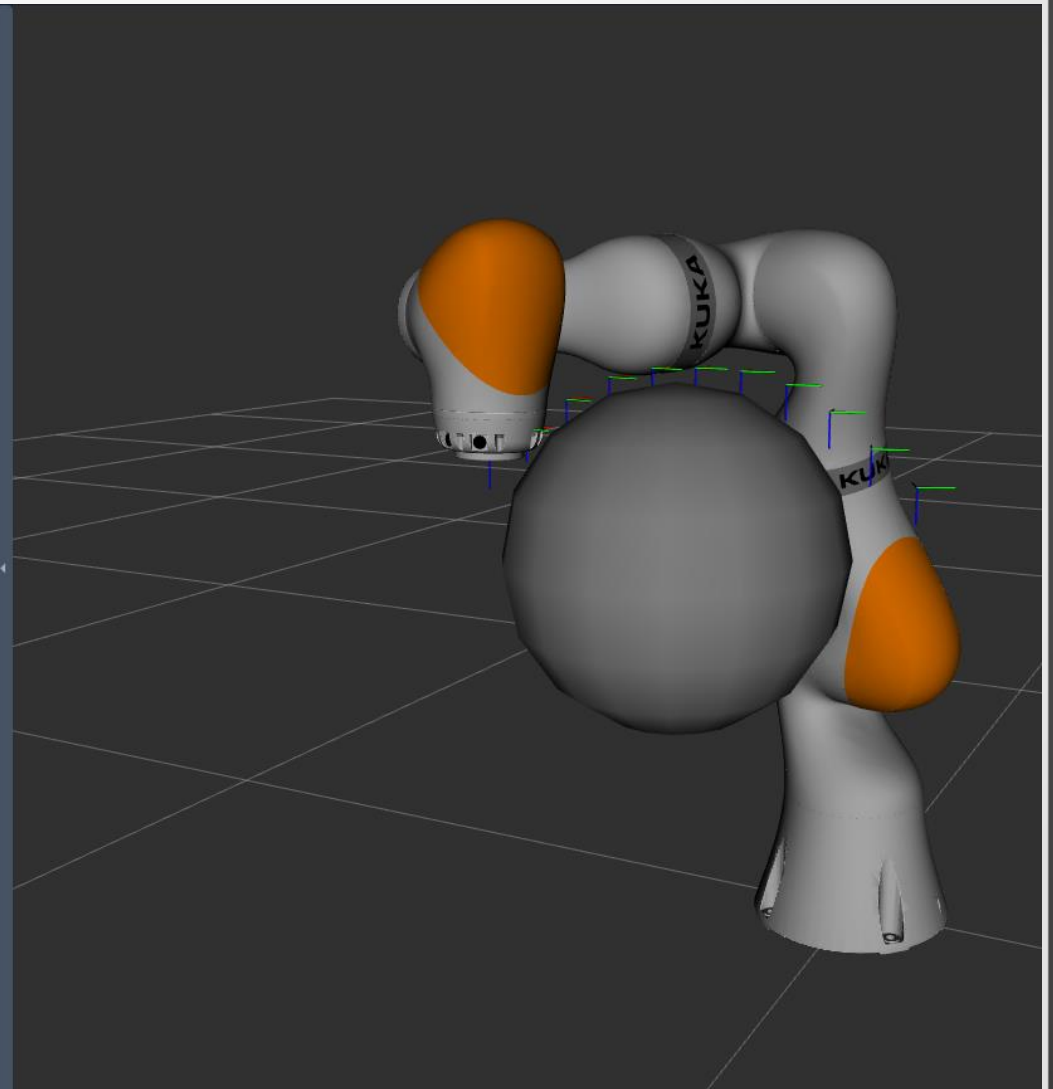
Environment Trajectory Manipulation Planning

Scene State ACM Groups States TCPs History C

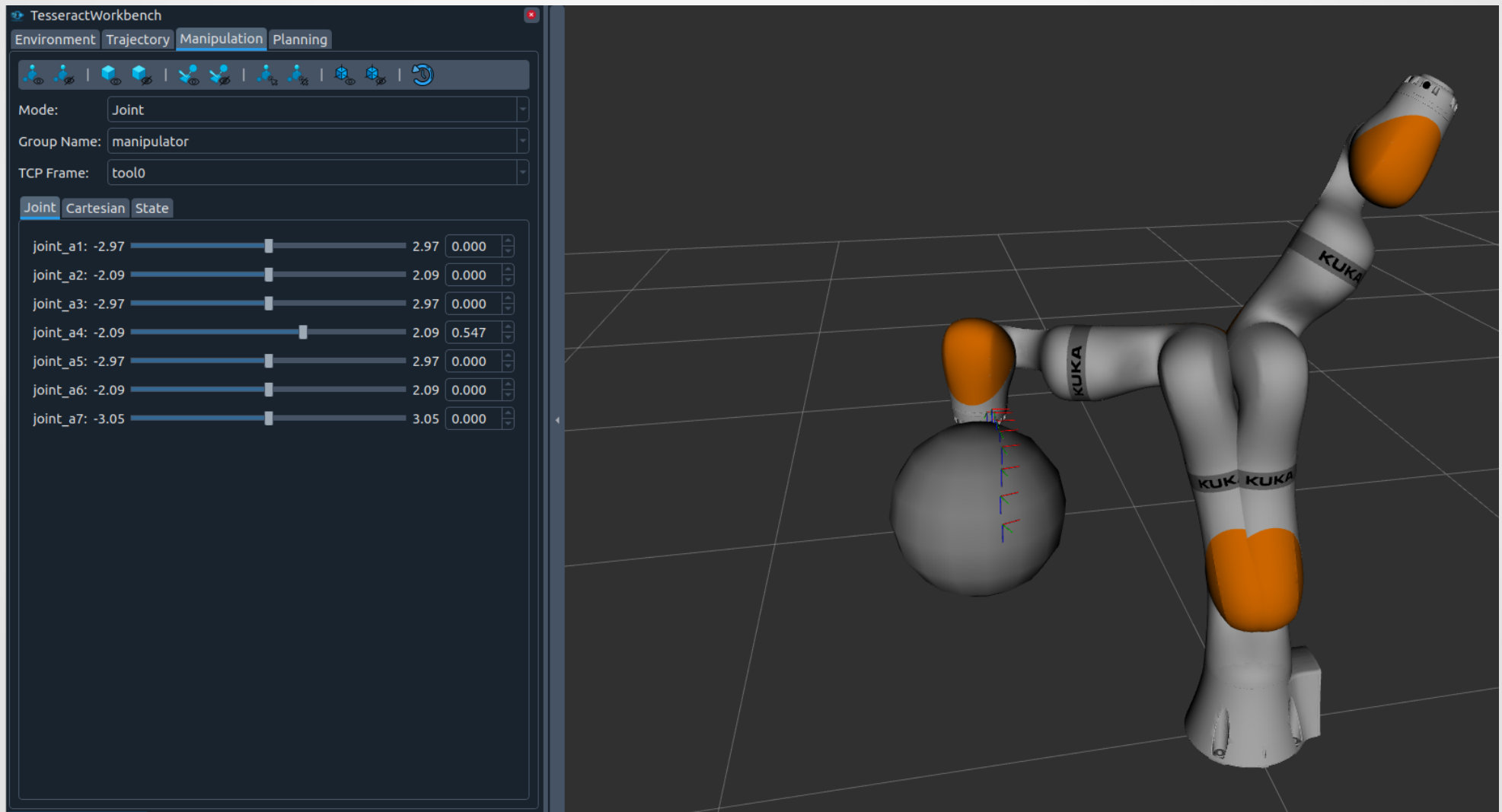
Name	Values
SceneGraph	
name	kuka_lbr_iiwa_14_r820
root_link	base_link
Links	
base	
base_link	
link_1	
link_2	
link_3	
link_4	
link_5	
link_6	
link_7	
sphere_attached	
tool0	
Joints	

Name	Values
general	
Trajectory Set	
trajectory[0]	
state[0]	
state[1]	
state[2]	
state[3]	
joint_names	
[0]	joint_a1
[1]	joint_a2
[2]	joint_a3
[3]	joint_a4
[4]	joint_a5
[5]	joint_a6
[6]	joint_a7
position	
[0]	-0.173119
[1]	0.164667
[2]	-0.00734349
[3]	-1.28171
[4]	0.00191708
[5]	1.66053
[6]	-0.0201381
velocity	
[0]	0.468546
[1]	-0.133367
[2]	0.0101434
[3]	0.0623506
[4]	-0.00368501
[5]	0.195687
[6]	0.0294433
acceleration	
time	2.09113
state[4]	
state[5]	
state[6]	
state[7]	
state[8]	

0.000 13.570



# Manipulation Widget



The image displays the TesseractWorkbench Manipulation Widget interface. The left panel shows the control interface, and the right panel shows a 3D visualization of a KUKA robot arm.

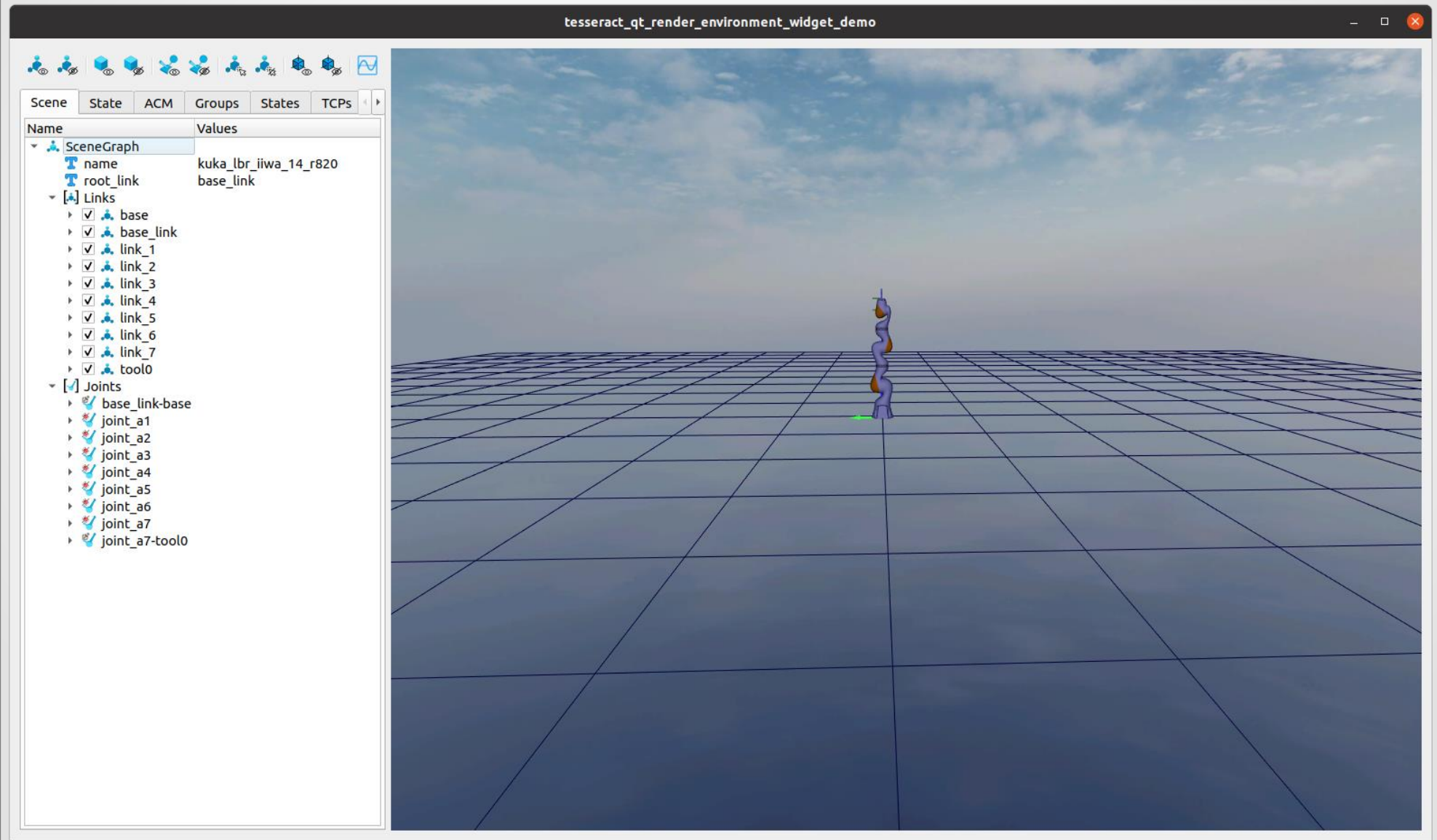
**Control Panel (Left):**

- Environment: Trajectory **Manipulation** Planning
- Mode: Joint
- Group Name: manipulator
- TCP Frame: tool0
- Tab: Joint Cartesian State
- joint\_a1: -2.97 [Slider] 2.97 0.000
- joint\_a2: -2.09 [Slider] 2.09 0.000
- joint\_a3: -2.97 [Slider] 2.97 0.000
- joint\_a4: -2.09 [Slider] 2.09 0.547
- joint\_a5: -2.97 [Slider] 2.97 0.000
- joint\_a6: -2.09 [Slider] 2.09 0.000
- joint\_a7: -3.05 [Slider] 3.05 0.000

**3D Visualization (Right):**

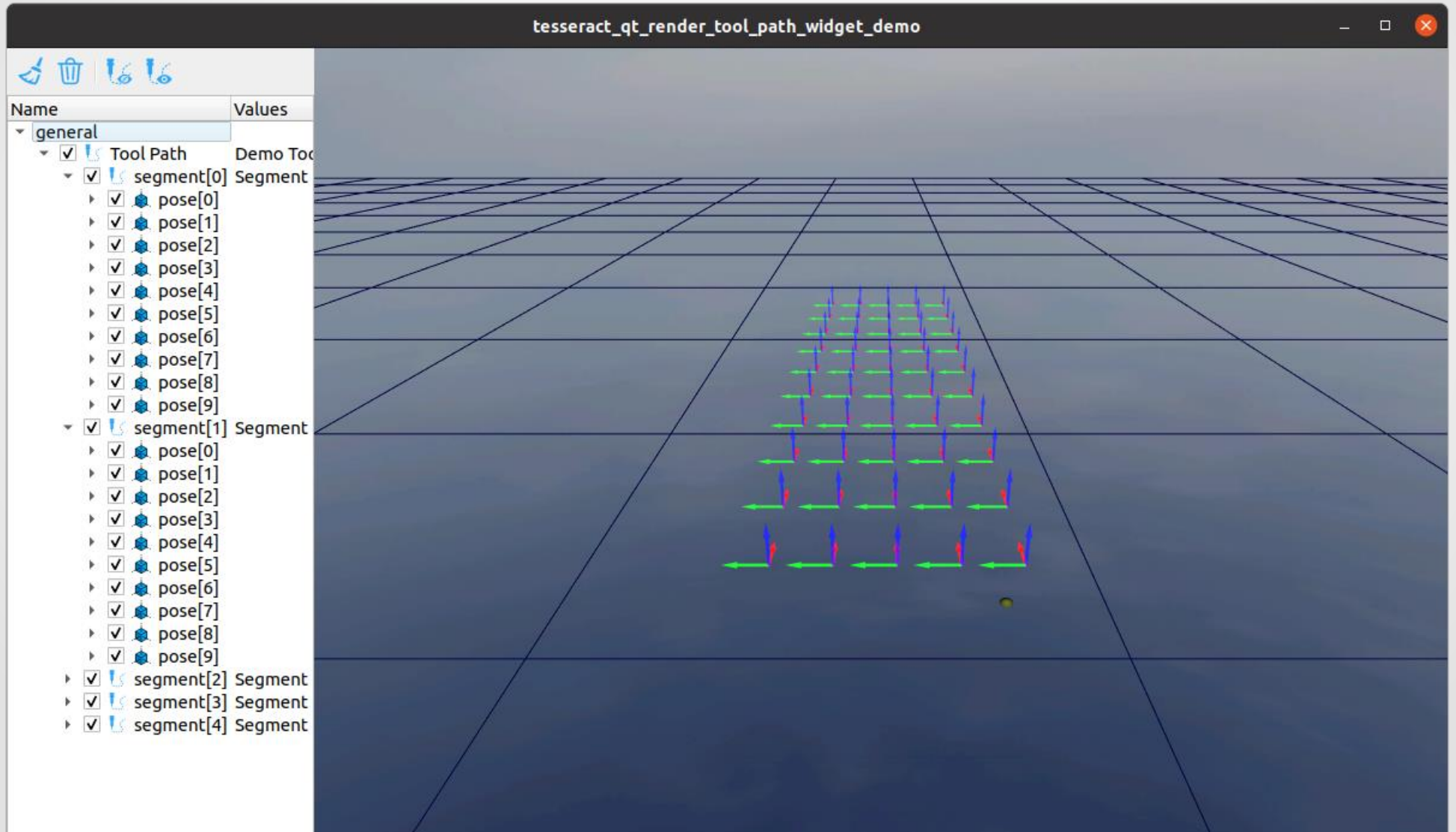
A 3D model of a KUKA robot arm is shown in a virtual environment. The robot arm is white with orange accents on the joints. The base is labeled "KUKA". The end effector is a spherical tool. The robot arm is positioned over a grey sphere, which is likely a target or a tool. The background is a dark grey grid.

# 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