



ROS-Industrial Advanced Developer's Training Class

March 2023

Southwest Research Institute





Advanced Topic:

Motion Planning with Tesseract

Southwest Research Institute





Types of Motion Plans

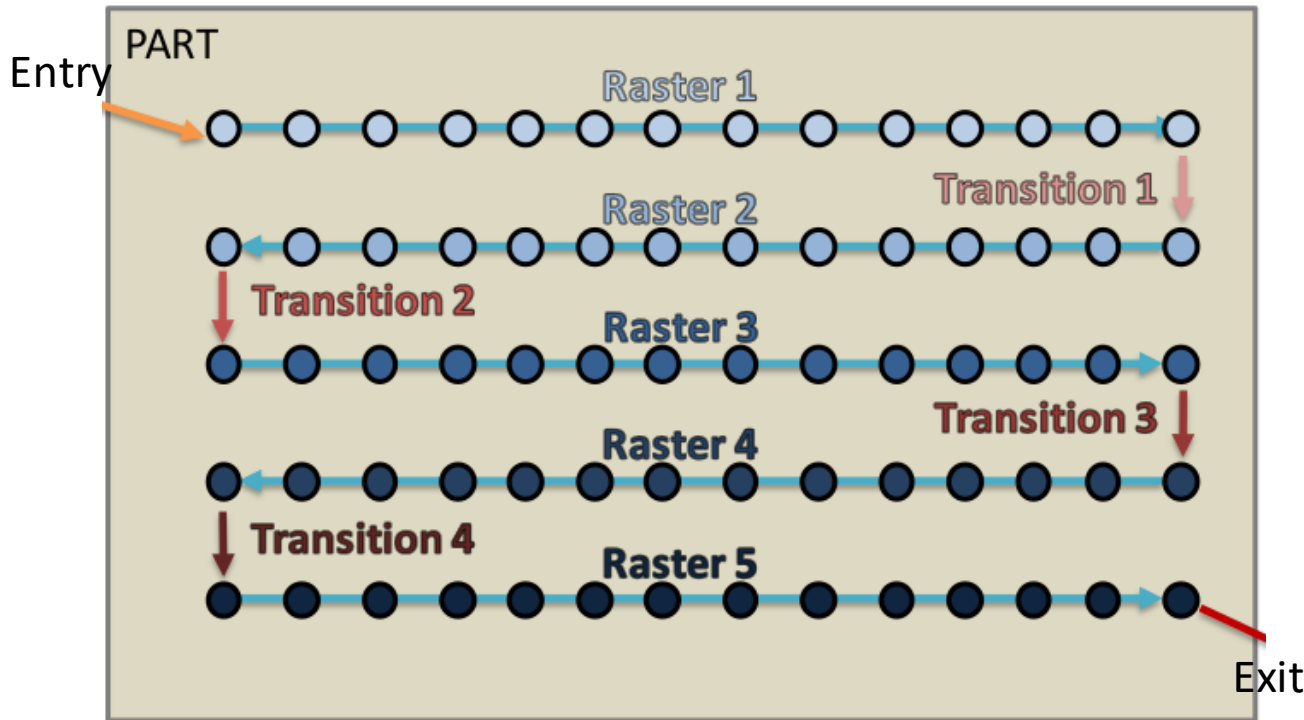


Freespace	Process	Combined
Motion plans between far-spaced start and end points	Motion plans optimize robot pose between under-constrained waypoints	Motion plans that can be segmented into portions that are freespace motions and others that are process motions
Example: Moving from a generic "start pose" to the upper righthand corner of a surface for painting	Example: A continuous line mapped around the edge of a piece to be welded	Example: Moving from a generic "start pose" to the edge of a jig-held part and then welding the edge at a known angle





Toolpath Plan Example



Definitions:

- **Raster** – A series of specified Cartesian waypoints to be executed (without breaking)
- **Transition** – A freespace move between rasters
- **Entry/Exit** – A freespace move from/to a position away from the part





What is Tesseract?



- A motion planning library
- Designed to be light weight, limiting the number of dependencies, mainly only using standard libraries
- Core packages are ROS agnostic and have full python support.
- <https://github.com/tesseract-robotics/tesseract>
- <https://tesseract-docs.readthedocs.io/en/latest/index.html>





Tesseract Core Packages



- **tesseract** – Manages the major component Environment, Forward Kinematics, Inverse Kinematics and loading from various data.
- **tesseract_command_language** – Contains a generic command language to support motion and process planning like industrial teach pendants
- **tesseract_collision** – Contains a common interface for collision checking. It includes both continuous and discrete collision checking for convex-convex, convex-concave and concave-concave shapes.
- **tesseract_common** – Contains common functionality needed by most of the packages.
- **tesseract_environment** – Contains the Tesseract Environment which provides functionality to add,remove,move and modify links and joint.
- **tesseract_geometry** – Contains geometry types used by Tesseract including primitive shapes, mesh, convex hull mesh, octomap and signed distance field.
- **tesseract_kinematics** – Contains a common interface for Forward and Inverse kinematics for Chains, Trees and Graphs.
- **tesseract_scene_graph** – Contains the scene graph which is the data structure used to manage the connectivity of objects in the environment. It provides addition functionality for adding, removing and modifying Links and Joints along with search implementation.
- **tesseract_support** – Contains support data used for unit tests and examples.
- **tesseract_visualization** – Contains visualization utilities and libraries.

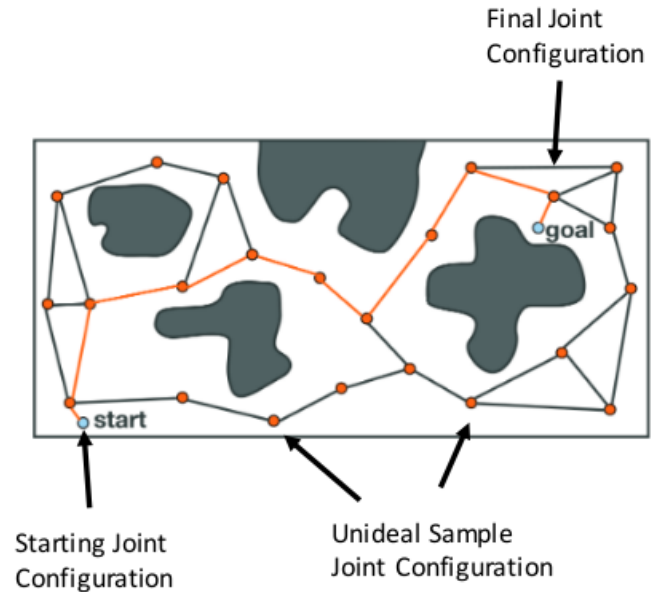




Motion Planner: OMPL



- Open Motion Planning Library
- Randomly sample valid joint states then solve for a sequence
- Most often used:
 - RRT
 - Build a tree along different potential joint configurations
 - RRT Connect
 - Build a tree from each side and try to connect them

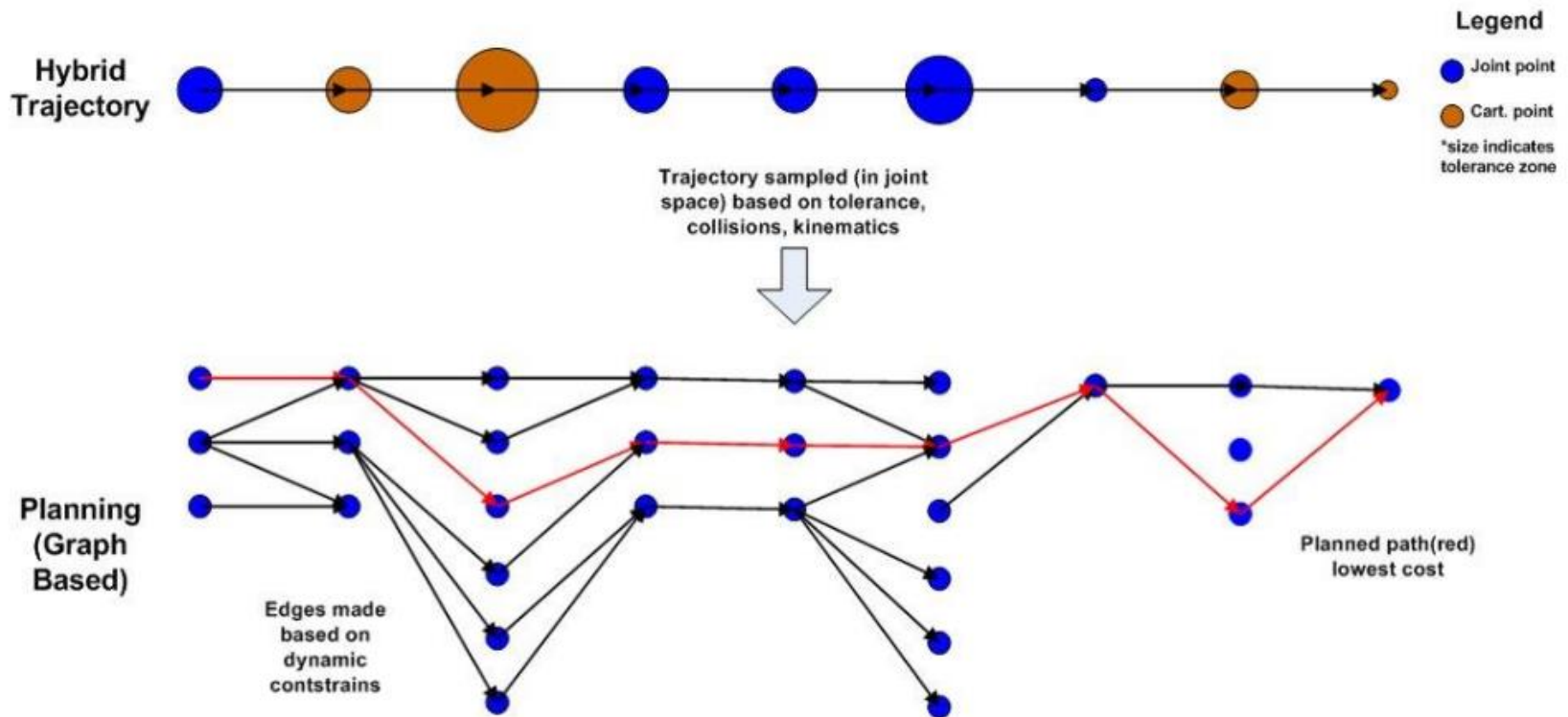




Motion Planner: Descartes



- Sample "all" possible solutions then graph search for the best trajectory





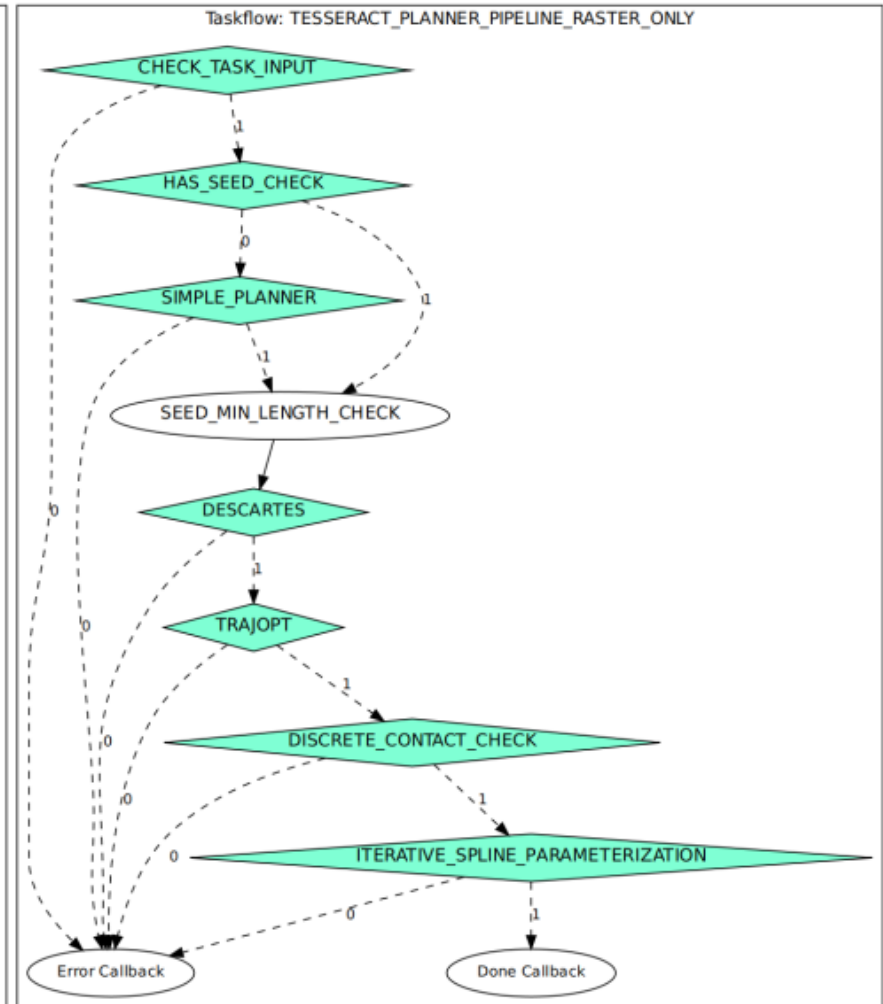
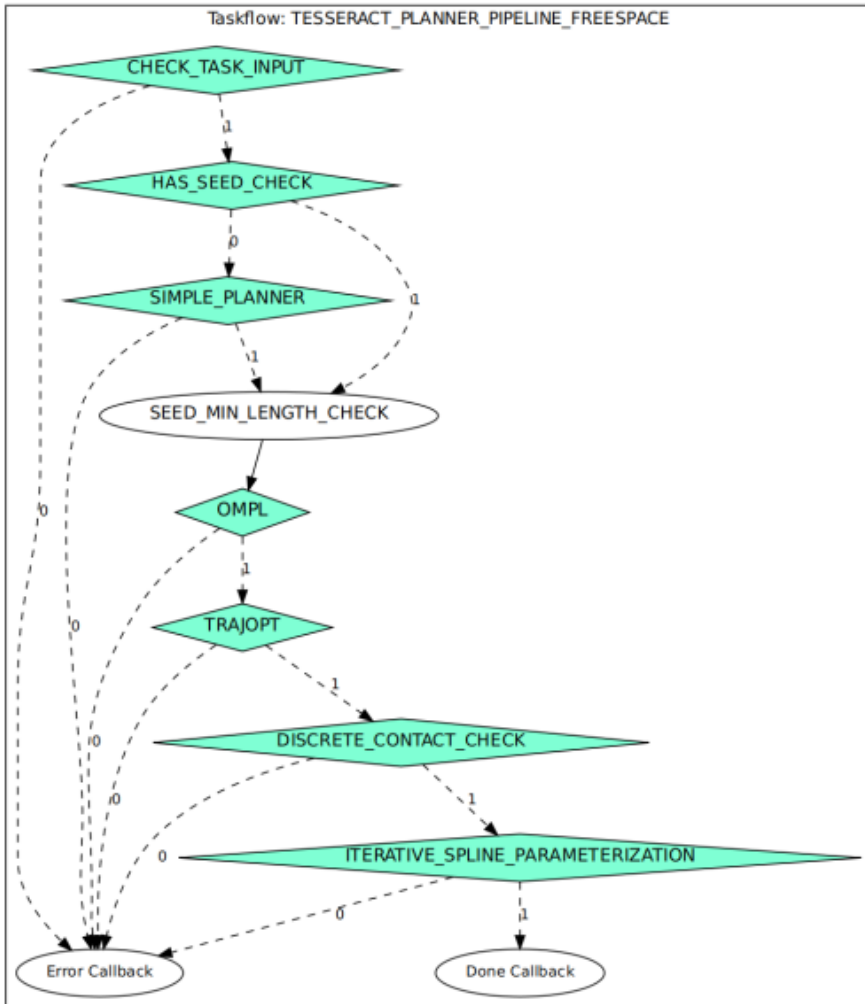
Motion Planner: TrajOpt



- Seed trajectory based on weighted cost functions
- Example costs:
 - Distance from collision
 - Joint limits
 - Velocity/Acceleration/Jerk
 - Encourage/Discourage DOF usage
- Constraints are simply infinite costs
- Often used to smooth trajectories



Motion Planning Taskflow





Your Task



- Goal: Complete the motion planning pipeline for a Scan 'n Plan application.
- Based off of a previous SwRI demo for Automate.
- Fill in custom planning profiles and add them to the planning server and taskflow.





Code Highlights



- **Workcell_plugins.yaml** - Contains all kinematic and contact manager plugins
- **Planner_profiles.hpp** - Contains all custom planner profiles and configurations
- **Planner_server.hpp** - Loads all custom planner profiles. `CreateProgram()` method takes the toolpath rasters and connects them with transitions
- **Taskflow_generators.hpp** - Creates the taskflow graphs for planning transition and freespace motions





Exercise 8.0



- Exercise 8.0 - <https://industrial-training-master.readthedocs.io/en/foxy/source/session5/Motion-Planning-with-Tesseract.html>

