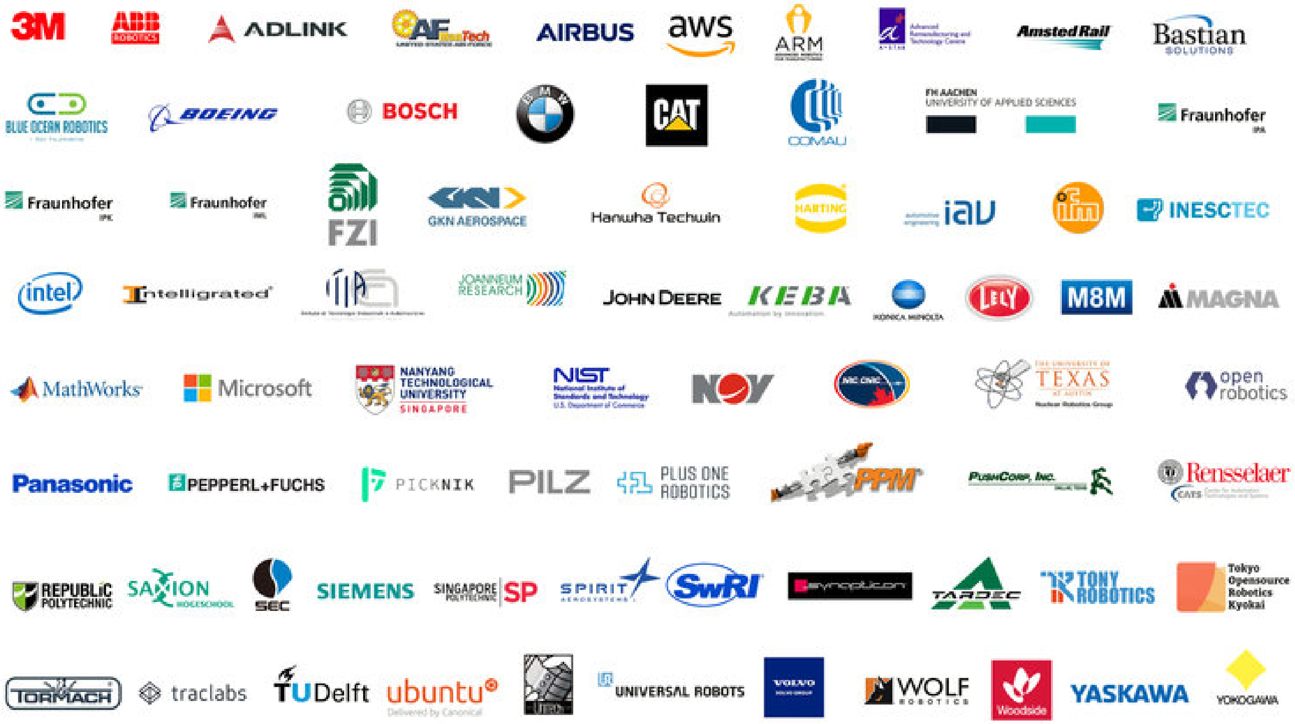
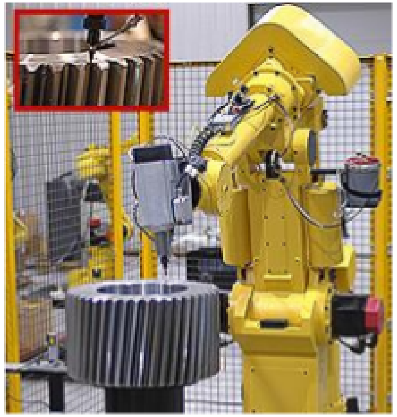
ROS-Industrial Consortium

- ROS-Industrial is an open-source project that extends the advanced capabilities of ROS software to manufacturing.
- www.rosindustrial.org
- www.github.com/ros-industrial

ROS® industrial

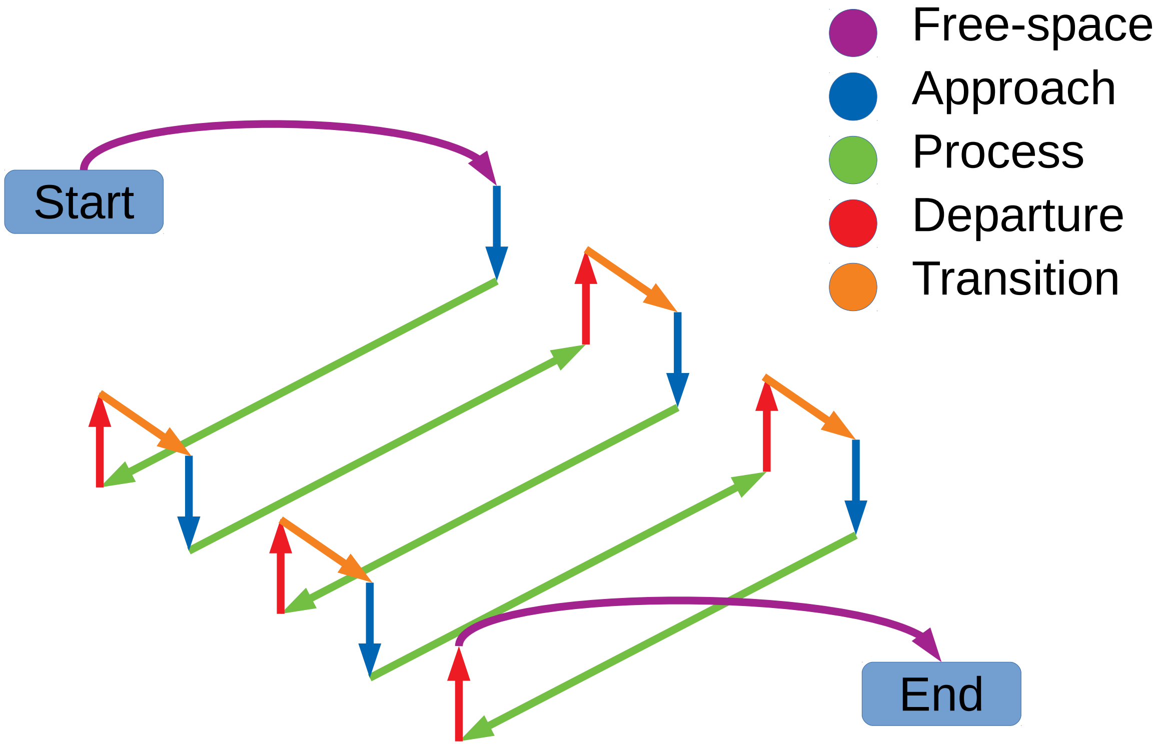


SwRI Research



Process Planning

- Most industrial processes can be represented by raster paths
  - i.e. painting, sanding, grinding
- Raster paths composed of waypoints
  - Joint, joint-toleranced, Cartesian, dynamic Cartesian
- Waypoints contain relevant planning data
- Composed of several trajectories
  - Process segments
  - Approach/departure
  - Transitions between segments
  - Free-space or cartesian



Start

End

● Free-space

● Approach

● Process

● Departure

● Transition

Hybrid Planning

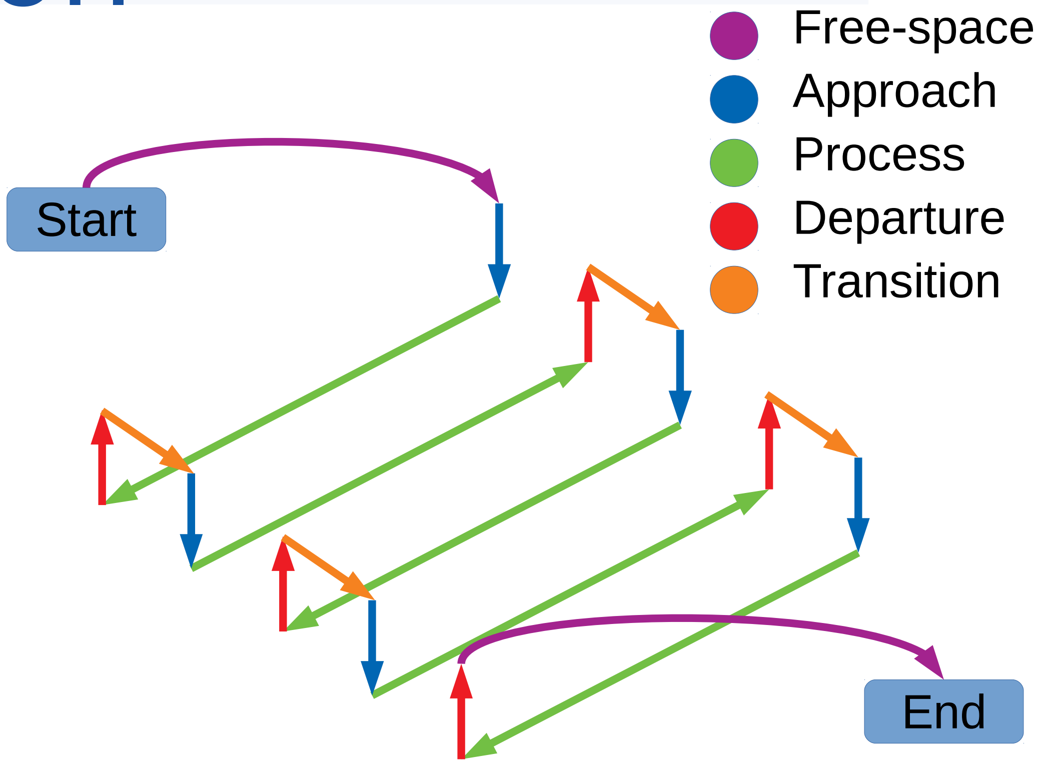
- Most industrial processes are defined as Cartesian paths → need Cartesian path planner
  - Descartes
- Find globally minimal cost solution*
  - *within sampling resolution
- Can we get more globally optimal solution?
  - TrajOpt!
  - Apply additional costs/constraints
    - Smooth velocity/acceleration/jerk
    - Avoid collision
    - Process constraints (tool orientation, etc.)
    - Maintain configuration, avoid robot controller safety zones

Process Planning Improvements

- Planning edge case failures
  - Waypoints are unreachable
  - Segments are infeasible
- How can we make raster planning more reliable?
- Behavior trees to make plans flexible
  - Drop unreachable waypoints from segment edges
  - Drop “non-critical” waypoints
  - Drop entire segments
  - Split segments → configuration change during transition

Process Execution

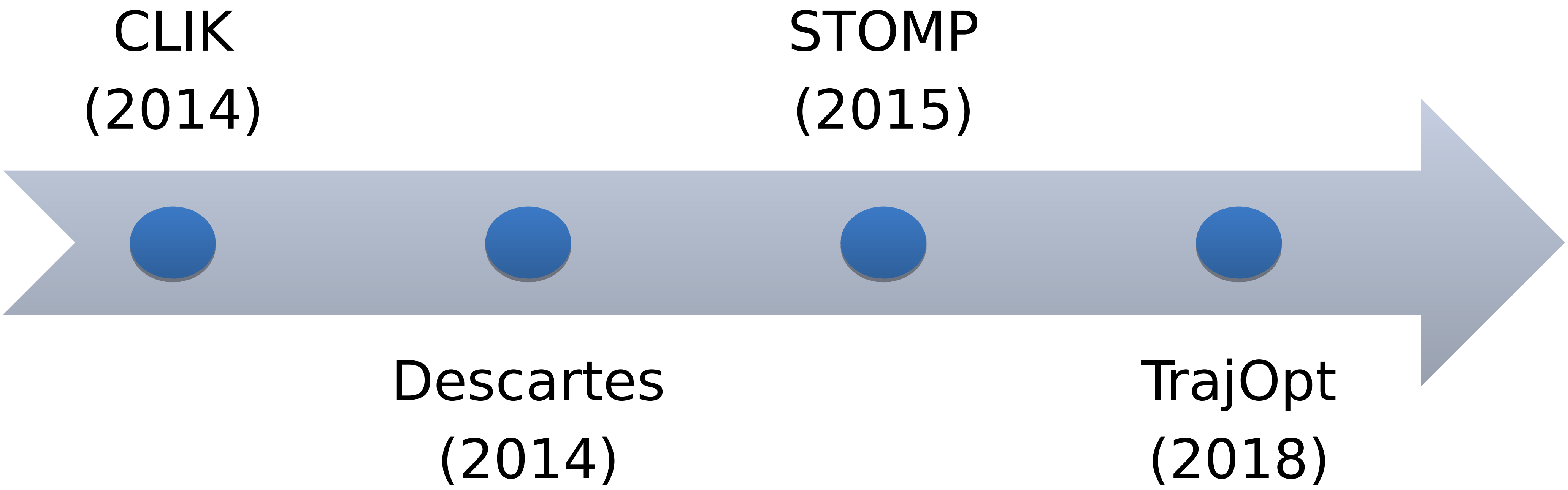
- Process execution can be interrupted
  - Tool faults, maintenance, tool media refill, tool change, etc.
- How can we avoid dynamically re-planning an escape and return from a process trajectory?
- Process paths as connected graph
  - Represent the various paths as graph edges and start/end states as graph nodes
  - Execute trajectories piece-wise as moves from node to node
  - Search the graph for the fastest way to safe state on interruption
  - Dynamic planning if necessary as last resort



Process Execution

- Graph based implementation
  - Create graph of nodes (i.e. named states) and edges (trajectories)
    - Boost graph library
  - Create a FIFO stack of the nominal process path
  - Execute trajectories off stack
  - On interruption
    - Query the graph for the trajectory/trajectories from the current state to the “safe state”
    - Combine resulting trajectories into one
    - Push the trajectory and its reverse onto the stack

SwRI Motion Planning History



CLIK
(2014)

Descartes
(2014)

STOMP
(2015)

TrajOpt
(2018)

TrajOpt/Tesseract Evolution

- January 2018
  - Awarded 4 month IR&D to evaluate the viability of TrajOpt for industrial applications.
  - It was not planned to be open-sourced, but would be evaluated on completion if it should be.
  - The original implementation used Bullet2 and OpenRAVE.
  - Started with replacing OpenRAVE and Bullet with MoveIt.
- Spent about 2 months getting familiar with TrajOpt, Bullet, and OpenRAVE by porting it to MoveIt.
- Initially it was partially working with MoveIt, but required significant changes to support the same functionality as the original implementation.
- Since this was not the focus of the IR&D it was decided to create a minimal interface using ROS to fully evaluate the viability of TrajOpt for industrial application currently in the pipeline.

Tesseract Evolution

- The evaluation showed promising results but required hybrid planning to improve the success rate.
- This was not intended to compete with MoveIt it was to address intimidate need driven by client demands around the Scan-N-Plan functionality.

- Also, for internal client projects it is desirable to be able to break things and not have to worry about backwards compatibility.