



MoveIt

Workshop 2019 Macau

Sponsored by





Movelt Capabilities Roadmapping



Recap: Key New Features In MoveIt Ecosystem

- **MoveIt Task Constructor**
 - *Task Planning*
 - *Robert Haschke, Michael Görner*
- **MoveIt Grasps**
 - *Geometric-based grasp generation*
 - *Mike Lautman, Dave Coleman*
- **MoveIt Cpp**
 - *Advanced API for performance*
 - *Henning Kayser*
- **MoveIt JogArm**
 - *Realtime teleoperation planner*
 - *Andy Zelenak*
- **CHOMP Planning Adapter**
 - *Post-processing of OMPL-generated plans*
 - *Raghavender Sahdev*
- **Iterative Cubic Spline Algorithm**
 - *Smoother trajectory generation*
 - *Ken Anderson*
- **Time-Optimal Trajectory Parameterization**
 - *Follow path within bounds on accelerations & velocities*
 - *Michael Ferguson, Henning Kaiser*
- **Named Frames on Collision Objects**
 - *Subframes for placing objects*
 - *Felix von Drigalski*

Milestone 1

Straight Port to ROS 2

Fully migrate existing Movelt packages to ROS 2
Wrap up Acutronic's work porting core Movelt functionality
Leverage ROS 2:
Build system (ament), middleware, logging, parameters
Cleanup Movelt 2 codebase

Milestone 2

Realtime Support

Reactive, closed-loop control to sensor input
Visual servoing, octomap updates
Preempt motion if new collision detected
Separate global and local planner (hybrid planning)
Global planner (full collision checking): 30hz
Local planner (IK-based, field-based): 300hz
Zero-memory copy integration to controllers (ros_control)
Tighter integration to ros_control
Integrate pilz_industrial_motion

Movelt Survey Results

91% most excited about ROS 2 realtime control
55% reactive planning and closed loop control
48% better integration with lower level realtime control
48% planning with dynamics

Milestone 3

Fully Leverage ROS 2

Lifecycle management of Movelt nodes
Deterministic startup, reset, & shutdown sequences
Leverage ROS2 component nodes
Ability to run Movelt as single or multi-process
Replace pluginlib with components
Cleanup API
More generic and standalone interfaces

Movelt Survey Results

47% excited about component nodes

Future Milestones

Determinism

Out of box / default planners return reliable paths
Tune or replace OMPL, BIT*
Further optimize / smooth paths
Default use TOTG, TOPP time parameterization
Use post-processing optimization (STOMP, TrajOpt)
Fully featured Cartesian Planner
Like Descartes but better and fully integrated
Force-torque control

Improved Interfaces / State Machines

Deprecate the Pick and Place pipeline
Fully support the Movelt Task Constructor
First class support of state machines
Non-ROS C++ API
Similar to MoveGroup but without middleware

Machine Learning

Neural-network based motion planning - new plugins
General near-optimal heuristics for path planning
e.g. MPNet



Open Source Planning

Cathedral vs Bazaar (*Eric Raymond*)

- Cathedral
 - Software carefully crafted by individuals
 - Isolated, mostly secret development team
- Bazaar
 - Chaotic, babbling open source development
 - Miraculously coherent among the noise of the crowds



Solutions

Write down

- Approaches to fix the categories