MoveIt2

Realtime Motion Planning
ROS Industrial 2020 Annual Meeting

Dave Coleman, PhD
CEO, PickNik Robotics
@davetcoleman
We are your partners in strategically developing custom robotics software, while de-risking open source usage.
About MoveIt

**MoveIt:** A Hardened Motion Planning Platform

![arm_navigation](image1.png) → ![MoveIt](image2.png) → ![MoveIt](image3.png) → ![MoveIt2](image4.png)
MoveIt Capabilities

- **Motion Planning**
  - Generate high-degree of freedom trajectories through cluttered environments and avoid local minimums
- **Manipulation**
  - Analyze and interact with your environment with grasp generation
- **Inverse Kinematics**
  - Solve for joint positions for a given pose, even in over-actuated arms
- **Control**
  - Execute time-parameterized joint trajectories to low level hardware controllers through common interfaces
- **3D Perception**
  - Connect to depth sensors and point clouds with Octomaps
- **Collision Checking**
  - Avoid obstacles using geometric primitives, meshes, or point clouds
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>109,880</td>
<td>Unique users to moveit.ros.org in 2019</td>
</tr>
<tr>
<td>23,662</td>
<td>Downloads per month of moveit_core</td>
</tr>
<tr>
<td>542</td>
<td>Academic citations of MoveIt</td>
</tr>
<tr>
<td>152</td>
<td>Robot types integrated to work with MoveIt</td>
</tr>
<tr>
<td>4200</td>
<td>Members of Discourse, MoveIt’s Discussion Forum</td>
</tr>
<tr>
<td>509</td>
<td>Github users have starred the MoveIt project</td>
</tr>
<tr>
<td>187</td>
<td>Github code contributors to MoveIt</td>
</tr>
<tr>
<td>13</td>
<td>International locations participated in World MoveIt Day 2018</td>
</tr>
<tr>
<td>310</td>
<td>In-person participants of World MoveIt Day 2018</td>
</tr>
</tbody>
</table>
# A Feature-Rich Ecosystem

<table>
<thead>
<tr>
<th>Global Planners</th>
<th>Cartesian Planners</th>
<th>Inverse Kinematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>● OMPL</td>
<td>● RobotState</td>
<td>● KDL</td>
</tr>
<tr>
<td>● SBPL</td>
<td>● Descartes</td>
<td>● IKFast</td>
</tr>
<tr>
<td>● TrajOpt</td>
<td>● JogArm</td>
<td>● TrackIK</td>
</tr>
<tr>
<td>● STOMP</td>
<td>● PilzIndustrial Motion</td>
<td>● LMA</td>
</tr>
<tr>
<td>● CHOMP</td>
<td></td>
<td>● BioIK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grasping Libraries</th>
<th>Collision Checking</th>
<th>Perception / Octomap</th>
</tr>
</thead>
<tbody>
<tr>
<td>● MoveIt Grasps</td>
<td>● Fast Collision Library (FCL)</td>
<td>● Depth Images</td>
</tr>
<tr>
<td>● Grasp Pose Detection (GPD)</td>
<td></td>
<td>● Point Clouds</td>
</tr>
<tr>
<td>● Intel OpenVino GPD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Collision Checking</th>
<th>Perception / Octomap</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>● Fast Collision Library (FCL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Bullet</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What's new in MoveIt?
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

- **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
Movelt Grasps
MoveIt Cpp Interface

- Designed by Industry-requested needs
- Speeds up manipulation product development
- As simple as current MoveGroup
- Disables ROS 1 performance bottleneck
- Direct access to core components provided as needed
- Multi-robot support
Named Frames on Collision Objects
Time Parameterization

- Iterative Cubic Spline Algorithm
  - Smoother trajectory generation
  - Ken Anderson

- Time-Optimal Trajectory Parameterization
  - Follow path within bounds on accelerations & velocities
  - Michael Ferguson, Henning Kaiser
ROS 2 & Realtime
Why ROS 2?

- Realtime support possible
- Multi-platform support: Linux, Windows, OSX
- Production-ready framework based on industry feedback of ROS 1
- DDS: open communication standard
Why care about realtime?

- Vital to many robotics systems, particularly safety and mission critical apps
  - Autonomous vehicles, spacecrafts, and industrial manufacturing.

2 primary types of Realtime:

- **Hard realtime** - missing a deadline is considered a system failure
  - Safety- or mission-critical systems
  - Reactor, aircraft and spacecraft control
- **Soft realtime** - missing a deadline has a cost, but is not catastrophic
  - Reduced quality of service
  - Audio / video streaming and playback
Realtime Computing

Determinism, not performance

- Correct computation guaranteed to be delivered within fixed time allotment
- Failure to respond is as bad as a wrong response

Credits to Jackie Kay's 2015 ROSCon presentation
Applying Realtime: Best Practices

- Realtime Operating System (RTOS)
  - Linux + RT Preempt (soft realtime)
  - Xenomai (hard realtime)
- Zero memory copy message passing:
  - Shared memory between threads or processes
- Lock-free circular buffers
- Prioritize real-time threads
- Avoid system calls (memory allocations, printing to console, mutexes)

Note these techniques have largely been available in ROS 1, e.g. MoveItCpp.
Types of Middleware Communication

- **Inter-process**
  - DDS can deliver soft realtime comms
  - Customizable QoS, can be tuned for real-time use-case

- **Intra-process (several options)**
  - Efficient (zero-copy) shared pointer transport
  - Shared memory with read-only and write-only partition
  - Non-locking circular message queues

- **Same-thread**
  - No need for synchronization primitives. Simple, fast
Realtime Motion Planning

- Enables:
  - Closed loop, reactive control
  - Streaming joint commands (torques, velocities) to robot arms at high rates (e.g. >1000 Hz)

- Improves:
  - Reliability
  - Extended uptime
MoveIt 1.0
Out of Box Approach

- User Application
- MoveGroupInterface
  - ROS 1 Actionlib (non-realtime)
- MoveGroup
  - Core MoveIt functionality
  - Sampling Based Motion Planner
  - ROS 1 Actionlib (non-realtime)
- ROS Control
  - Non-Realtime ROS Interface
  - Realtime Hardware Abstraction Layer

MoveIt Configurations
Movelt 1.0
New Advanced Approach with MoveltCpp

User Application
MoveltCpp
Core Movelt functionality
Sampling Based Motion Planner
ROS 1 Actionlib (non-realtime)

ROS Control
Non-Realtime ROS Interface
Realtime Hardware Abstraction Layer
Movelt 2.0
Current Beta Implementation

User Application
MoveltCpp
Core Movelt functionality
Sampling Based Motion Planner
ROS 2 Joint Pub/Sub
ROS 2 Control ??
Movelt 2.0
Proposed Implementation with ROS 2 Component Nodes

- User Application
- MoveItCpp
  - Core Movelt functionality
- Sampling Based Motion Planner
- ROS 2 Control
  - Component Node (Thread)
Movelt 2.0
Hybrid Motion Planning

- User Application
  - MoveltCpp
    - Core Movelt functionality
  - Sampling Based Global Planner
  - Inverse Kinematics Local Planner
- ROS 2 Control
- Component Node (Thread)
- Component Node (Thread)
- Component Node (Thread)
Global vs Local Planning

Global Planning (assuming sampling)

- Pros:
  - Plan around complex obstacles
  - Avoid getting stuck in local minimum
  - Complete: will find solution if exists

- Cons:
  - Slower computation time
  - Not realtime
  - Not deterministic

Local Planning (assuming jacobian based)

- Pros:
  - Fast / Reactive
  - Deterministic
  - Well suited for visual servoing

- Cons:
  - Gets stuck in local minimum
  - Fewer collision safety guarantees
Hybrid Planning

- Simultaneously plan globally and locally
- Plan at different speeds in separate thread:
  - Global planner (full collision checking): ~30Hz
  - Local Planner (IK-based, field-based): ~300Hz
Deterministic Planning

- Out of box / default planners return reliable paths
  - Improved support for OMPL, TrajOpt
- Further optimize or smooth motions
  - Default use TOTG, TOPP time parameterization
  - Post-processing optimization (STOMP, TrajOpt)
- Fully featured Cartesian Planner
Roadmap
MoveIt 2.0 Releases

- Alpha
  - Released June 2019
  - ROS 2 Dashing Diademata
- Beta
  - Released February 2020
  - ROS 2 Eloquent Elusor
**Straight Port to ROS 2**
- Fully migrate existing MoveIt packages to ROS 2
- Wrap up Acutronic's work porting core MoveIt functionality
- Leverage ROS 2:
  - Build system (ament), middleware, logging, parameters
- Cleanup MoveIt 2 codebase

**Realtime Support**
- Reactive, closed-loop control to sensor input
- 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 pLz_industrial_motion

**Fully Leverage ROS 2**
- Lifecycle management of MoveIt nodes
- Deterministic startup, reset, & shutdown sequences
- Leverage ROS2 component nodes
- Ability to run MoveIt as single or multi-process
- Replace pluginlib with components
- Cleanup API
- More generic and standalone interfaces

**Future Milestones**

**Determinism**
- Out of box / default planners return reliable paths
- Tune or replace OMP, 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 MoveIt 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

**MoveIt 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
- 47% excited about component nodes
Progress on Roadmap

1. Finish migration of MoveIt 1 packages
2. Document how to use ROS1 bridges for legacy support
3. Merge and simplify ecosystem repositories
4. Address realtime support
5. Improve deterministic planning
Getting Involved
Getting Involved

Upcoming Events

Google Summer of Code

MoveltCon
November 17th

World MoveltDay
Contributing to MoveIt

https://moveit.ros.org/documentation/contributing/

Many approaches:

- Adding New Features
- Helping with MoveIt 2 Port
- Financial contributions via code sprints and grants
- Enhancing Documentation
- Reporting & Fixing Bugs
Thanks!

Dave Coleman
dave@picknik.ai
@waffle

PickNik Robotics
(+1) 720 513 2221
1942 Broadway, Suite 314
Boulder, Colorado 80302
@picknikrobotics