The Descartes Planning Library for Semi-Constrained Cartesian Trajectories

Principle Engineer
Shaun Edwards
Southwest Research Institute
Application Need

• Semi-constrained trajectories: traj. DOF < robot DOF
Current Solution

- Arbitrary assignment of 6DOF poses, redundant axes -> IK
- Limited guarantee on trajectory timing
- Limitations
  - Reduced workspace
  - Relies on human intuition
  - Collisions, singularities, joint limits
Descartes

• Planning library for semi-constrained trajectories

• Requirements
  – Generate common sense plans
  – Find easy solutions fast, hard solutions with time
  – Handle hybrid trajectories (joint, Cartesian, specialized points)
  – Fast re-planning/cached planning
Descartes Example

- Robotic Routing
Performance

- 800 5DOF (6DOF robot) waypoints – 30s 1-10s plan time
- Path planners minimize joint motion
- Re-planning near instantaneous
- Support for hybrid, kinematic planning
Robotic Routing

https://www.youtube.com/watch?v=cZxt00uoyBo
• Trajectory Point
  – Robot independent
  – Tolerance (fuzzy)
  – Timing

• Robot Model
  – IK/FK
  – Validity (Collision checking, limits)
  – Similar to MoveIt::RobotState, but with **getAllIK**

• Planner
  – Trajectory solving
  – Plan caching/re-planning
Descartes Implementations

- **Trajectory Points**
  - Cartesian point
  - Joint point
  - AxialSymmetric point (5DOF)

- **Robot Model**
  - MoveIt wrapper (working with MoveIt to make better)
  - FastIK wrappers
  - Custom solution

- **Planners**
  - Dense – graph based search
  - Sparse – hybrid graph based/interpolated search
Descartes Implementations

Legend
- Joint point
- Cart. point
*Size indicates tolerance zone

Hybrid Trajectory

Trajectory sampled (in joint space) based on tolerance, collisions, kinematics

Planning (Graph Based)

Edges made based on dynamic constraints

Planned path (red) lowest cost
Open Source Details

• Public development: https://github.com/ros-industrial-consortium/descartes
• Documentation: http://wiki.ros.org/descartes
• Releases
  – Hydro (stable) – binary
  – Indigo (unstable) – source
• Tutorials: http://wiki.ros.org/descartes/Tutorials
• Acknowledgements:
  – Supported by: NIST (70NANB14H226), ROS-Industrial Consortium FTP
Shaun Edwards

Southwest Research Institute

Contact

Phone: 210-522-3277
Email: sedwards@swri.org