IDEXX-SwRI Path Planning Optimization ROS-I Consortium FTP

Chris Lewis
Southwest Research Institute
IDEXX and SwRI FTP for Path Planning Optimization

- MoveIt! employs OMPL’s planners. Although fast, paths are often non-intuitive and sub-optimal

- FTP Goal:
  - Create a path planning capability which provided paths useful to industrial robot tasks.
  - Generated paths must be asymptotically optimal (longer computation → paths become more optimal)

- FTP Tasks:
  - Fix a planner bug that short-circuited planning optimization
  - Add control to GUI
  - Make tuning parameters accessible and explicit
  - Make parameter tuning dynamic
  - Implement a smoother
  - Add documentation for MoveIt! plug-ins
Bug

• Planner exited at first valid solution, even when set to try multiple attempts

• Detailed bug description:
  – ompl_interface::ModelBasedPlanningContext::solve()
  – Incorrectly called
    • ompl_parallel_plan_.solve(ptc, 1, max_planning_threads_, true)
  – The fix: changed to
    • ompl_parallel_plan_.solve(ptc, 1, count, true)
  – Results: Parallel Planner now computes as many as count solutions, combines best segments from all solutions

• Now, the planner runs until either
  – Planning_attempts_ is reached
  – Planning_time_ runs out
Addition to GUI

• Although the `move_group` interface allows one to set `num_planning_attempts()`, the GUI did not provide an interface.

• Added a `max_planning_attempts` input
OMPL provides an interface to a number of different planners.

Planner parameters are passed from the move_group interface to the move_group node then to the OMPL plugin through a generic parameter mechanism as a map.

Unknown to Most Users:
- What parameters does this planner have?
- What is the syntax to set parameters?
- What do they mean?
Ompl_planner.yaml

• Parameters are already read in from the ompl_planning.yaml file.

• Just edit, and add a desired parameter and its value to respective configuration.

• Example for DEFAULTRRTStarKConfig:
  – range: 0.05
  – goal_bias: 0.08
Setup Assistant

• Generated ompl_planning.yaml now includes definitions, default values, and a short description of each of the configurable parameters of the ompl planners.

• Results:
  – Planning parameter sets are “advertised”
  – Syntax is clear
  – Parameter meaning is clearer
Parameters On the Fly

• All the parameters for each OMPL planner are read from the ROS parameter server on construction. Therefore, one must restart the planning environment to change any of these important values.

• We modified MoveIt! To reload all the parameters every time solve() is called. This allows one to tinker with the parameters using the “rosparam set <parameter_name> <value>” syntax.
Important OMPL Parameters

- **Range**: determines how far, in configuration space, a randomly generated motion can be from an existing part of the path.
  - Large values $\Rightarrow$ fast planning with wild motion
  - Small values $\Rightarrow$ slower planning with intricate motion

- **Goal_bias**: Determines how often the goal is selected instead of some random point in C-Space.
Intricate Paths Aren’t Smooth

- When the *range* parameters is small, paths tend to be very choppy near obstacles
- Smoother trajectories desired
- It is also instructive to learn how to write a plugin for MoveIt!.
- Wrote a planning adapter plugin for MoveIt! which smoothes the path. May cause collisions, especially for very sparsely generated paths.
- A tutorial was written describing this process:
• Planning Request Adapter: add_smoothing_filter
• Smoothes trajectory, but is not collision aware
Improved Paths

Video Link
Getting the Updates

• Install MoveIt! from source
• Erase all three:
  – Moveit_setup_assistant
  – Moveit_core
  – Moveit_ros
• New setup assistant
  – Git clone
    https://github.com/drchrislewis/moveit_setup_assistant.git
  – cd moveit_setup_assistant
  – git branch fix-84
  – catkin_make (at top of workspace)
  – Run setup assistant on your urdf.
  – Look at config/ompl_planning.yaml
Getting the Updates

- fir planning request adapter
- GUI updates
  - git clone https://github.com/drchrislewis/moveit_ros.git
  - git clone https://github.com/drchrislewis/moveit_core.git
  - cd moveit_ros
  - git branch addFirFilter
  - cd moveit_core
  - git branch FIR_filter
  - catkin_make
- Edit launch file to add planning request adapter to planning pipeline if desired
- When running GUI look for new entry field next to planning time called planning attempts.