Winning with MoveIt!
Team Delft – Amazon Picking Challenge

Mukunda Bharatheevesha
08.10.2016
# Self Introduction

<table>
<thead>
<tr>
<th>Stowing Challenge</th>
<th>Picking Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delft - 214</td>
<td>Delft – 105 (00:00:30)</td>
</tr>
<tr>
<td>NimbRo - 186</td>
<td>PFN – 105 (00:01:07)</td>
</tr>
<tr>
<td>MIT - 164</td>
<td>NimbRo - 97</td>
</tr>
</tbody>
</table>

![Team Photo]
Contents

• Getting started with the Amazon Picking Challenge (APC) [1]

• Team Delft APC motion module with MoveIt! [2]

• MoveIt! Specific lessons learnt

• General lessons learnt

• Concluding remarks
Getting started with APC

[Fiad-tud-3me] Fwd: [robocup-worldwide] Amazon Picking Challenge @ RoboCup 2016

fiad-tud-3me-bounces@lists.tudelft.nl on behalf of G.A. vd. Hoorn - 3ME [g.a.vanderhoo...

To: fiad-tud-3me@lists.tudelft.nl
Attachments: (2) Download all attachments
   Attached Message Part (246 B); ATT00001.txt (230 B)

Thursday, December 17, 2015 3:58 PM

so ..... 

;)

Gijs

-------- Original Message --------
Subject: [robocup-worldwide] Amazon Picking Challenge @ RoboCup 2016
Date: Thu, 17 Dec 2015 14:53:06 +0000
From: Durham, Joey <josepdur@amazon.com>
To: robocup-worldwide@cc.gatech.edu <robocup-worldwide@cc.gatech.edu>

Amazon Robotics is pleased to announce the second edition of our manipulation contest to be held at RoboCup at the end of June 2016 in Leipzig, Germany (http://www.robocup2016.org/).
Why MoveIt!? 

- Previous experience with pick and place applications.
- Knowledge of using pick and place pipeline for an industrial “boxpacking” task.
- Curiosity! [3]
Robot setup

Workspace analysis of different configurations

- Fixed Configuration
  - Upper and lower bins at the centre.
  - Deep corner bins.

- Vertical configuration
  - Side and corner bins.
  - Practicalities with the rail.
Robot Setup

Selected configuration

```bash
roslaunch moveit_workspace_analysis workspace_analysis.launch
roslaunch moveit_workspace_analysis reader.launch
```
Basic Software Architecture

High level task manager

**Sense**
- Object Recognition
- Object Pose Estimation
- Bin Pose Estimation

**Plan**
- Grasp Synthesis
- Manipulation Planning

**Act**
- Coarse Motion
- Fine Motion
- I/O
Motion Module - Manipulation

- Grasp pose scoring
  - Perpendicular distance to CoM.
  - Approach direction of the robot.
  - Preferred grasping direction.

- Grasp strategy based MoveGroups
  - Top Suction.
  - Front Suction.
  - Pinch.

- Final grasp pose selection
  - Highest scoring grasp pose
    *(the_chosen_one)*.
  - Collision check based pruning
    *(moveit_msgs::GetPositionIK)*.
Motion Module – Robot motion

Robot Motions

Offline

Coarse Motions
Free (joint) space motion planning with RRT-Connect.
(Trajectory Cache)

Fine Motions
Cartesian path planning to achieve final grasp pose.

Online

• `robot_state::RobotState` APIs for setting various “MasterPoses”.

• `planning_interface::MoveGroup` APIs for planning.
Motion Module – I/O (1)

• “Event” based I/O
  • Key waypoints - “approach”, “contact”, “lift”, “retreat”
  • I/O actions - “vacuum on”, “suction on”, “front suction activate”, …

• Reliable trajectory tracking
  • Track trajectory waypoints (distance to waypoints).
  • Stable tracking regardless of other tasks.
  • Dedicated callback queue and background spinner.
Motion Module – I/O (2)

- Requires additional “AsyncSpinner”
  - `MoveGroup::execute()` already running an AsyncSpinner.

- New `SimpleAsyncSpinner`
  - No dependencies besides ROS and standard library.
  - Lock-free using atomic variables.

- Multiple AsyncSpinners now supported in ROS-Kinetic [4]
Lessons Learnt
1. Collision checking redundancy

“Gap” in octomap due to reflections
Lessons Learnt

1. Collision checking redundancy

- Bin collision models

- Collision checking with “new” meshes in the planning scene
  - Non-reproducible but random failures in collision checking.
- Add sanity checks for “impossible” motions.
Lessons Learnt
2. Octomap – Planning Scene

- “Static” point cloud
  - One point cloud snapshot instead of a point cloud “live feed”.
  - Simulated point cloud stream using a topic.

- `MoveGroup: :getCurrentState()` takes long time to return while using octomaps
  - Use `robot_state: :JointModelGroup` to avoid delays when real robot state is different from “test” robot state during online planning.
  - Cause: Planning scene monitor prioritises servicing octomap updates(*)
Lessons Learnt

3. Octomap clearance

- “Static” point cloud
  - One point cloud snapshot instead of a point cloud “live feed”.
  - Simulated point cloud stream using a topic.
- `/clearOctomap()` - Inconsistency with clearing the current octomap.
  - Publish “out of range” valued point cloud.
  - Octomap always cleared on service call.
Lessons Learnt
4. Trajectory stitching

- Combine fine motions with coarse motions
  - IK solution need not necessarily match with coarse motion start.
  - RRT-Connect from last IK point to “MasterPose”

- Trac-IK with “distance” setting
  - ROS parameter namespace issues.

- Possibly stitch entire trajectory to destination.
  - Combine multiple motion segments in joint space.
  - Re-(time) parameterize the “stitched” motions in time
    \[\text{iptp.computeTimeStamps}()\].
Lessons Learnt
5. I/O as joints?

• Common practice
  • Define various I/O as joints with a very small range.
  • “Plan” solutions to command I/O and use actionlib interface.

• Our approach via external trajectory monitoring
  • Dependency on /joint_states topic.
  • Dedicated I/O support possibly the way to go forward?
Lessons Learnt
Behavior design

- Interspersed Plan-Act
  - Robot stuck in the bin.
  - Design adequate and appropriate recovery behaviour.

- Fake components
  - Provide lots of room for testing.
  - Clarity in interface definitions.
  - Easy integration when real components are available.

- Writing “json” files correctly
Lessons Learnt

Assumptions

• Optimal planners
  • Euclidean distance NOT default.
  • Effort-based, time-based, etc.

• Trac-IK cost functions
  • Distance
  • Speed
  • Manipulability

• Robot Driver*
  • Sanity checks
  • Joint naming

“Assumptions bear the roots of all disasters!”

“Gravity is a heartless entity!”
Concluding Remarks

How MoveIt! is designed to be used.

How we used it!

Image Sources:
www.dx.com
www.northerntools.com
Concluding Remarks

20 days before the challenge!
(No stowing yet!)
Concluding Remarks

- Stowing completed in almost half of allotted time.
- More than 16 picks performed in the given time (including “move jobs”)
- Not moving like Icebergs anymore!
Acknowledgements

Made possible by:
Thank you for your attention!
References


[2] moveit.ros.org
