

Collaborative Manipulation Pipeline

Outline

- Overview
- Implementation details
- Current demonstrations
- Pros and Cons
- Next steps

SBPL Demo Framework

Motivation and Design Goals

Current State:

- Demos are time consuming and have to be made from scratch
- Past demos largely cannot be recreated or reused

Design Goals:

1. To enable multi-robot manipulation tasks (quadcopter projects out of scope)
2. Allow researchers to quickly script demos that feature their area of interest
3. Maintain demonstration code that can be reused
4. Allow independent development on robots on differing platforms

ROS Architecture

Tatooine
Indigo



Groovy ↔ Indigo
Moveit Interface

Alan1
Groovy



PR2 Local Master

Actions:

- Move_base
- Move_arm
- Torso_controller
- Point_head
- TF_lookup
- Gripper Cmds

Topics:

- Raw sensor data

roslaunch sbpl_multimaster sbpl_hub.launch

Rocon Hub (Redis Server)
@Dagobah:6830

SBPL
LAN

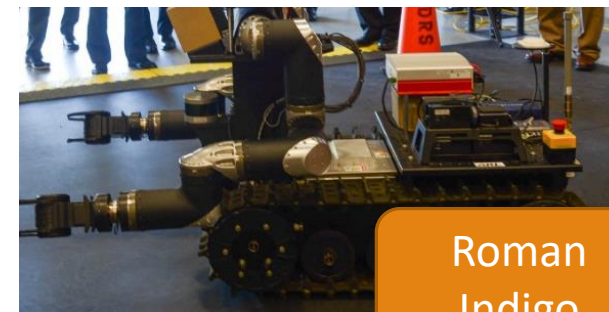
Rocon
Gateway

roslaunch sbpl_multimaster alan1_gateway.launch

Rocon
Gateway

roslaunch sbpl_multimaster dagobah_gateway.launch

Roman
Indigo

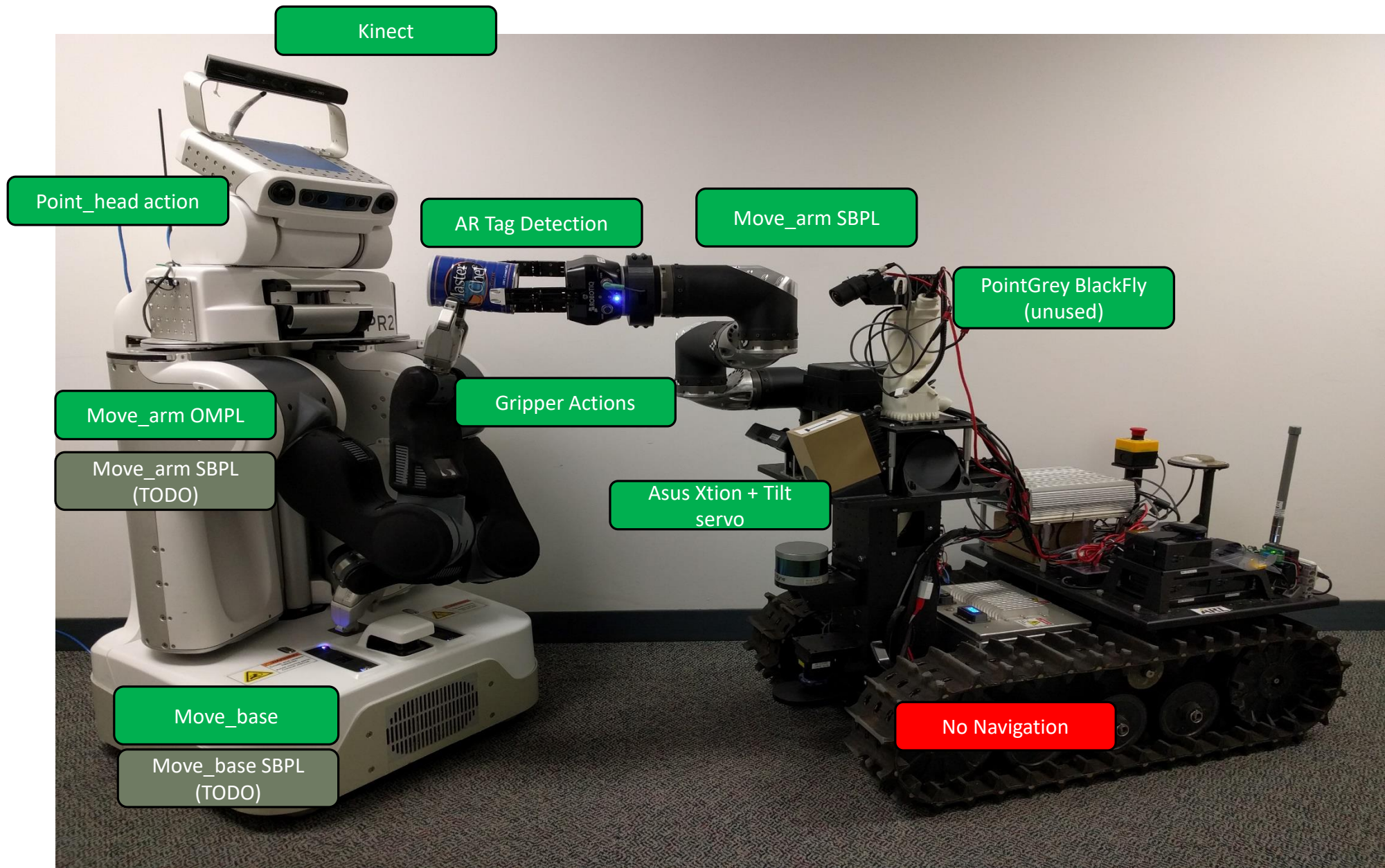


Dagobah
Indigo



Commander Local Master

- Action clients
- Service calls
- Read topics
- Runs State Machine
- TF bookkeeping between robots



Implementation details

SBPL_DEMOS

ROS package to contain python-based scripting utilities

- Robot-specific action clients and services
- State machine (python based)
- AR Tag functions

Status:

- **Functional, continued development**

https://github.com/sbpl/sbpl_demos

SBPL_MULTIMASTER

ROS package to manage the Rocon setup within lab

- Rocon configuration generator
- Gateway launch files
- Rocon hub launch

Status:

- **Completed**

https://github.com/sbpl/sbpl_multimaster

GROOVY_INDIGO_MOVEIT_WRAPPER

ROS package to manage PR2 from Indigo Moveit

- Custom message packages that match Groovy MD5sums
- I2g_follow_joint_trajectory action server

Status:

- **Completed**

https://github.com/abrinkmacmu/groovy_indigo_moveit_wrapper

Sbpl_demos



folder	action
folder	config/pr2
folder	data
folder	launch
folder	screen_configs
folder	scripts
folder	src
folder	ws_setup
file	CMakeLists.txt
file	package.xml
file	readme.md
file	setup.py

- Lab maps

- Installation instructions

- Pr2 and Roman details

folder examples

folder multi_robot_demos

folder nodes

folder sbpl_demos

folder single_robot_demos

-Example usage code, not clean but may help users

-

-

-

Convention for finding python scripts “from sbpl_demos.pr2_helpers import GripperCommander”

-

Screen Configurations

One Screen config per machine

Aids startup time and guides users to correct launches and scripts

Contained in
spbl_demos/screen_configs

Tatooine
Indigo

```
0$ bash 1$ teleop 2$ moveit 3-$ AR 4$* PR_demo
```

Alan1
Groovy

```
0$* bash 1-$ dash 2$ kinect 3$ 2dnav 4$ 2dmap
```

Dagobah
Indigo

```
0$ rHub 1$ rocon 2$ bash 3$ rom_view 4$ ctrl 5$ move 6$ map 7$ am4 8$ AR 9-$ asus_tilt 10$* demo
```

Roman
Indigo

```
[tl1-1-am1][ 0 bash 1- Limb 1 (Right) (2*Limb 2 (Left)) 3 Torso 4 Hand Right 5 Hand Left 6 Client 7 RCRU 8 dynmx mngr 9 dynmx meta ][2017-02-24 10:03 ]  
[Batt Voltage: -/bin/bash] [Load Avg(1m,5m,15m):2.09 1.56 0.77]
```


Sbpl_multimaster

The user should only have to modify one file

- Modify config/gateway_topics.yaml to suit your needs
- Launch the generate_multimaster_configs.yaml to create <machine>_gateway.launch files for all machines in the network
- Push the changes to the other machines
- Rerun the <machine>_gateway.launch files

Does require manual configuration of each machine after change

- could be addressed by installing workspaces

Config/gateway_topics.yaml

```
generator_commander_action_clients:
  pr2:
    - torso_controller/position_joint_action
    - tf2_buffer_server
    - r_gripper_controller/gripper_action
    - l_gripper_controller/gripper_action
    - head_traj_controller/point_head_action
    - move_base
    - pr2_move_base
    - pr2_move_base_local
    - move_base_local

#list out the servers the commander needs
# This will add the server to both the robot and commander configs
generator_commander_servers:
  pr2:
    - pr2/tf_lookup_server
  roman:
    - roman/tf_lookup_server
```


Single Robot: Roman Demo

`sbpl_demos/scripts/single_robot_demos`

Use AR pose detection to command a simple sequence of Arm motions

Required:

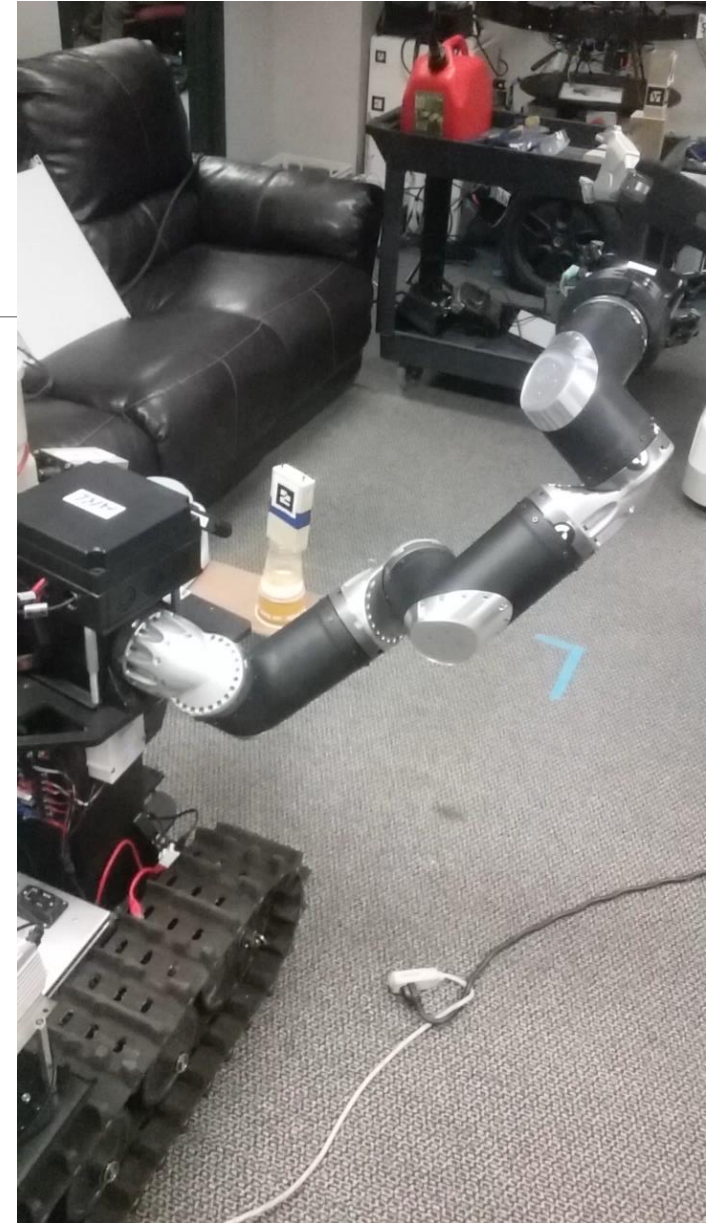
- Roman and Commander (Dagobah)
- AR Tagged item

Instructions:

- Place AR tag in view of the Asus, confirm AR pose estimation in Rviz
- The pre-grasp and post grasp are computed on the right side of the tag, so ensure the Roman can reasonably reach the tag
- Once supporting servers are running, execute `roman_grasp_test.py`

Improvement Ideas:

- Instead of just one grasp approach to the AR item, consider many (top down, 90 deg intervals, or RPY randomization)
- Include collision object for picking surface and attached collision object



Multi Robot: Hardcoded handoff

`sbpl_demos/scripts/multi_robot_demos`

Use AR pose detection to command a simple sequence of coordinated arm motions across 2 robots

Required:

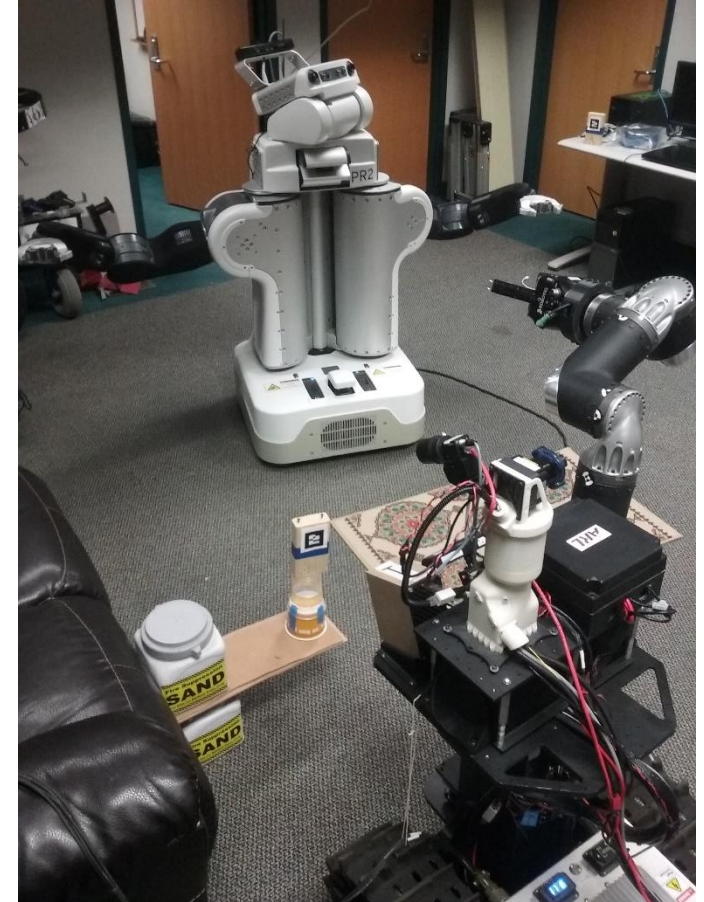
- Roman and Commander (Dagobah)
- PR2 and Commander (Tatooine)
- AR Tagged item

Instructions:

- Place AR tag in view of the Asus, confirm AR pose estimation in Rviz
- The pre-grasp and post grasp are computed on the right side of the tag, so ensure the Roman can reasonably reach the tag
- Once supporting servers are running, execute `roman_pr2_hardcoded_handoff.py`

Improvement Ideas:

- Coordinate collision objects between robots



Pain Points

- Roman elmo/ethercat driver initialization (ongoing)
 - Unhelpful error message:
 - *** ERROR (12) : Erroroneous response when reading SO
 - System restarts have been the only way I know to clear this issue, sometimes many times
- Rocon networking (intermittent)
 - Most of the time it functions properly but occasionally subscribed topics do not appear
 - Number of subscribers do not increase over the network, so beware of code that requires ≥ 1 subscriber to begin
- System Complexity (solved)
 - Shear number of computers and electronics to manage adds significant non-value overhead
 - Computers total: 9 (5 – roman, 4 – PR2), LAN configuration, Roman electronics

Strengths

- Rocon Solves Groovy/Indigo problem
 - Barrier to entry should be low for PR2 development
- Pr2 Navigation, actions, and Arm Planning
- Roman Arm Planning
- Lab network is robust across Pr2 and Roman
- AR tag pose estimation is robust

Next Steps

- Demos
 - Exploit more of the PR2's features in multi-robot demo (Alex)
 - Pass objects back and forth between robots indefinitely (Alex)
- Features
 - Explore moveit sensor manager for real-time collision obstacle generation (Tae-hyung)
 - Improve arbitrarily located AR tag grasping (Alex)
 - Collision object sharing between robots (Alex)
 - Other-robot collision modelling (Alex)
 - Get SBPL arm planner packages in sbpl_demos setup configurations (Sung)
- MRSD Handoff Development
 - An MRSD team of 5 people will be taking this system to create a more robust handoff system