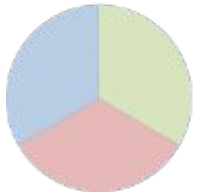


SECURE

Safety Enables Cooperation in Uncertain Robotic Environments

ROS basics with NAO

SoftBank Robotics

Natalia Lyubova

31.01.2017

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
    - connect NAO to ROS
    - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
    - Object recognition
    - MoveIt!
    - Teleop
    - Gazebo

Outline

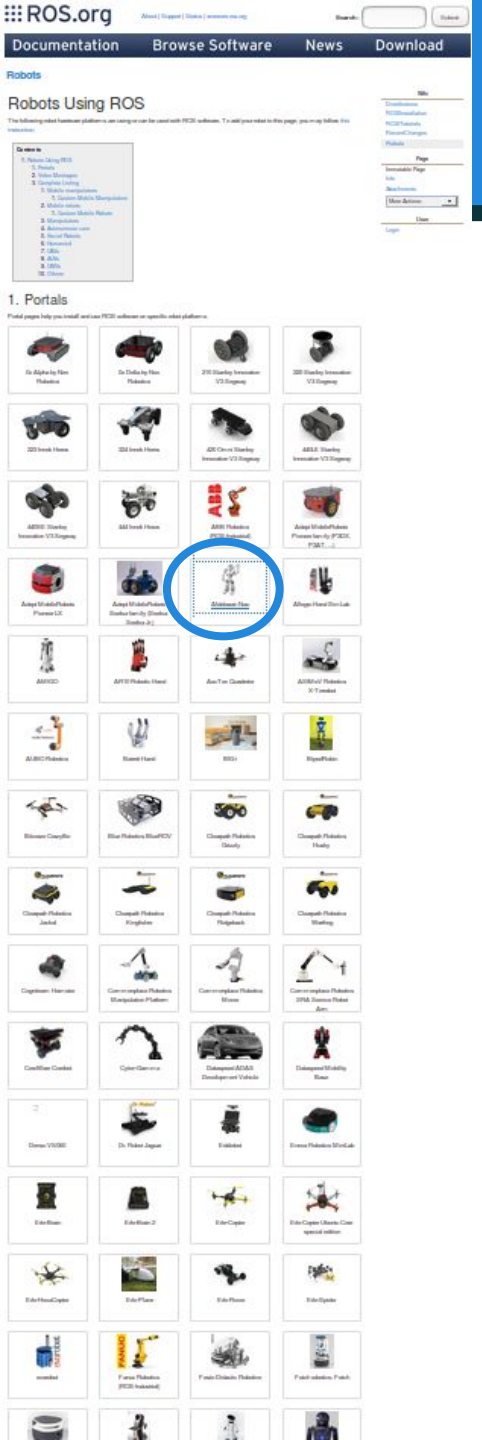
1. wiki
2. robot description in ROS
3. ROS Bridge
    - connect NAO to ROS
    - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
    - Object recognition
    - MoveIt!
    - Teleop
    - Gazebo

1. ROS wiki

http://wiki.ros.org

How many robots use ROS?

- http://wiki.ros.org/Robots
- about 100+ including
  - NAO
  - Romeo
  - Pepper



1. ROS wiki

Where is wiki?

- http://wiki.ros.org/nao
- http://wiki.ros.org/pepper
- http://wiki.ros.org/romeo

For issues and questions

- special interest group

  ROS SIG Aldebaran

Task 1: open NAO wiki page

SoftBank Robotics

1. ROS wiki

Where is wiki?

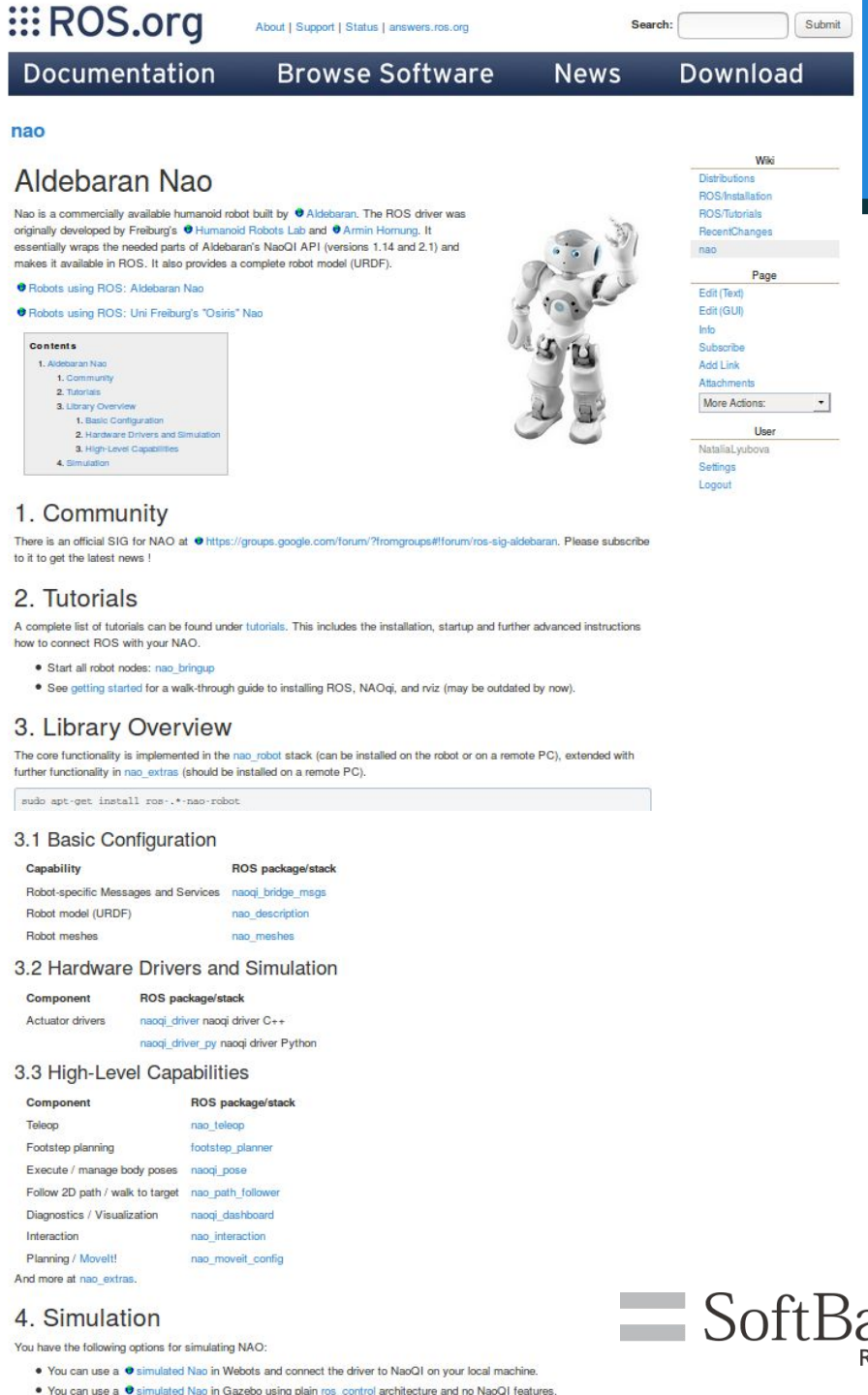
- http://wiki.ros.org/nao
- http://wiki.ros.org/pepper
- http://wiki.ros.org/romeo

For issues and questions

- special interest group

  ROS SIG Aldebaran

Task 1: open NAO wiki page



ROS.org About | Support | Status | answers.ros.org Search: Submit

Documentation Browse Software News Download

nao

Aldebaran Nao

Nao is a commercially available humanoid robot built by Aldebaran. The ROS driver was originally developed by Freiburg's Humanoid Robots Lab and Armin Hornung. It essentially wraps the needed parts of Aldebaran's NaoQI API (versions 1.14 and 2.1) and makes it available in ROS. It also provides a complete robot model (URDF).

Robots using ROS: Aldebaran Nao

Robots using ROS: Uni Freiburg's "Osiris" Nao

Contents

1. Aldebaran Nao
   1. Community
   2. Tutorials
   3. Library Overview
      1. Basic Configuration
      2. Hardware Drivers and Simulation
      3. High-Level Capabilities
   4. Simulation

Wiki
Distributions
ROS/Installation
ROS/Tutorials
RecentChanges
nao

Page
Edit (Text)
Edit (GUI)
Info
Subscribe
Add Link
Attachments
More Actions:

User
NataliaLyubova
Settings
Logout

1. Community

There is an official SIG for NAO at https://groups.google.com/forum/?fromgroups#!forum/ros-sig-aldebaran. Please subscribe to it to get the latest news !

2. Tutorials

A complete list of tutorials can be found under tutorials. This includes the installation, startup and further advanced instructions how to connect ROS with your NAO.

- Start all robot nodes: nao_bringup
- See getting started for a walk-through guide to installing ROS, NAOqi, and rviz (may be outdated by now).

3. Library Overview

The core functionality is implemented in the nao_robot stack (can be installed on the robot or on a remote PC), extended with further functionality in nao_extras (should be installed on a remote PC).

```
sudo apt-get install ros-.*-nao-robot
```

3.1 Basic Configuration

Capability	ROS package/stack
Robot-specific Messages and Services	naoqi_bridge_msgs
Robot model (URDF)	nao_description
Robot meshes	nao_meshes

3.2 Hardware Drivers and Simulation

Component	ROS package/stack
Actuator drivers	naoqi_driver naoqi driver C++
	naoqi_driver_py naoqi driver Python

3.3 High-Level Capabilities

Component	ROS package/stack
Teleop	nao_teleop
Footstep planning	footstep_planner
Execute / manage body poses	naoqi_pose
Follow 2D path / walk to target	nao_path_follower
Diagnostics / Visualization	naoqi_dashboard
Interaction	nao_interaction
Planning / MoveIt!	nao_moveit_config

And more at nao_extras.

4. Simulation

You have the following options for simulating NAO:

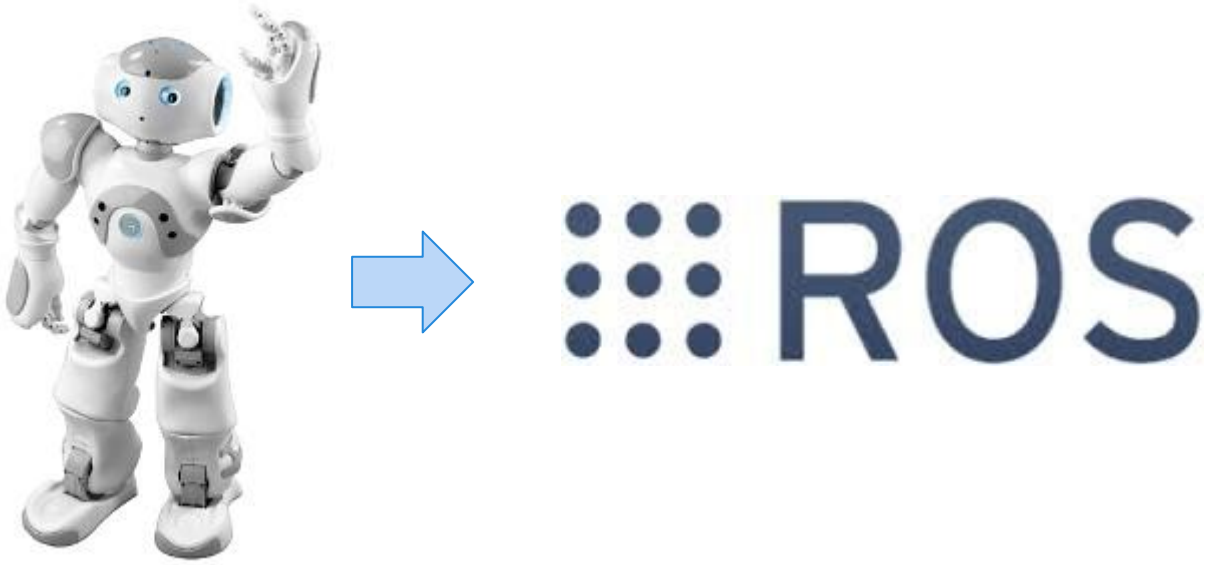
- You can use a simulated Nao in Webots and connect the driver to NaoQI on your local machine.
- You can use a simulated Nao in Gazebo using plain ros_control architecture and no NaoQI features.

SoftBank Robotics

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

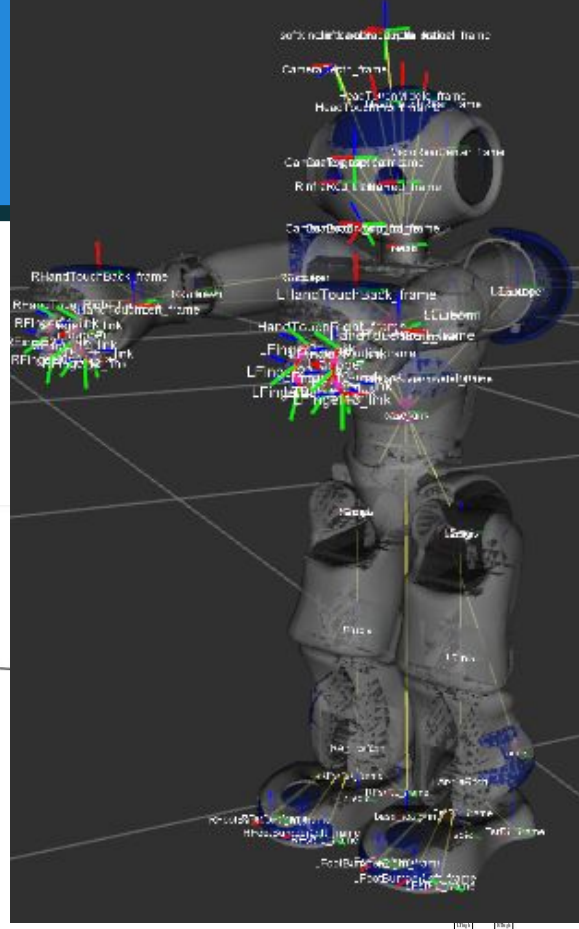
2. NAO description in ROS

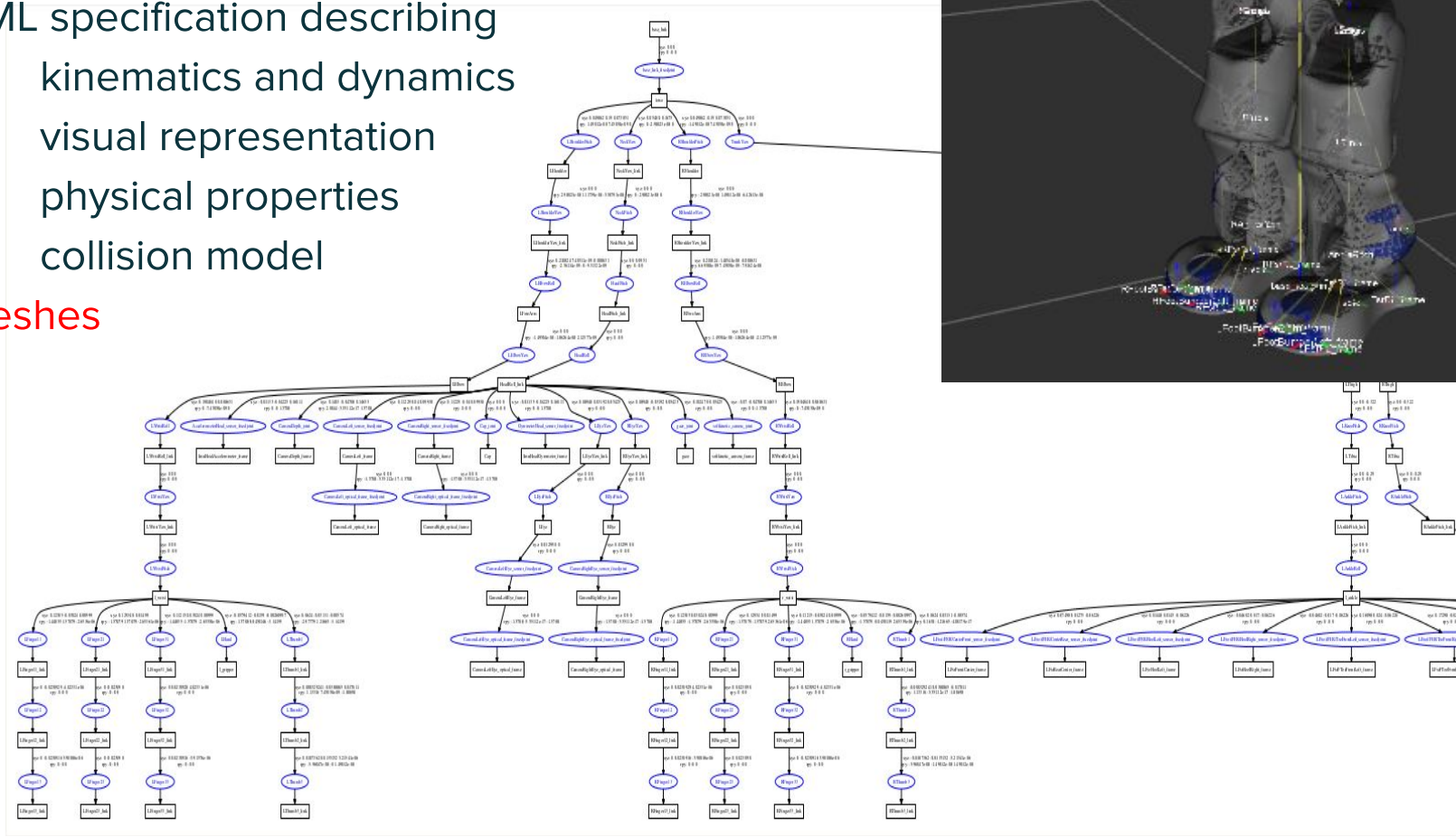


SoftBank Robotics

2. NAO description in ROS

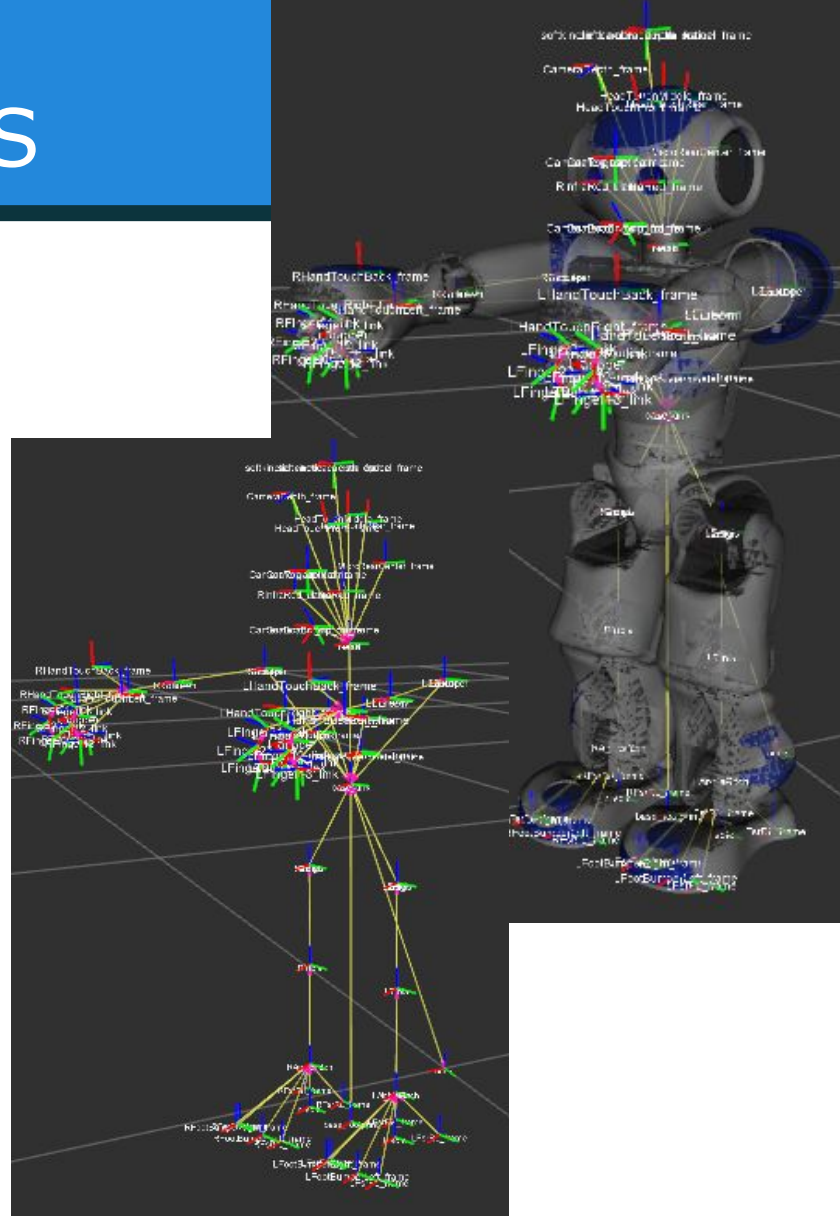
- what are the essential components for integrating a robot in ROS?
  - URDF: unified robot description format: XML specification describing
    - kinematics and dynamics
    - visual representation
    - physical properties
    - collision model
  - meshes





2. NAO description in ROS

- why do we need URDF robot description?
  - low level:
    - kinematics, actuators/sensors
  - high level:
    - planning, navigation, grasping
  - GUI



SoftBank Robotics

2. NAO description in R

- doc:
  - http://wiki.ros.org/nao_description
- install:
  - $ sudo apt-get install ros-kinetic-nao-description
- To build your robot description follow this tutorial

ROS.org

About | Support | Status | answers.ros.org

Search

Documentation | Browse Software | News

nao_description

electric | fuerte | groovy | hydro | indigo | jade | kinetic | Documentation Status

nao_robot: nao_apps | nao_bringup | nao_description

Package Summary

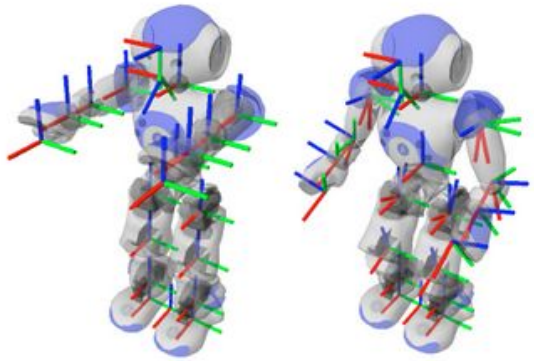
Released | Continuous integration | Documented

Description of the Nao robot model that can be used with robot_state_publisher to display the robot's state of joint angles.

- Maintainer status: maintained
- Maintainer: Séverin Lemaignan <severin.lemaignan AT epfl DOT ch>, Vincent Rabaud <vincent.rabaud AT gmail DOT com>
- Author: Armin Hornung, Stefan Osswald
- License: BSD
- Bug / feature tracker: https://github.com/ros-nao/nao_robot/issues
- Source: git https://github.com/ros-naoqi/nao_robot.git (branch: master)

Package Links
Code API
FAQ
Changelog
Change List
Reviews
Dependencies (8)
Used by (2)
Jenkins jobs (13)

Contents
1. Documentation



1. Documentation

Nao's URDF description contains all joints and links according to the documentation by Aldebaran Robotics for V3 and V4 Naos. In accordance with REP-105 (Coordinate Frames for Mobile Platforms) and REP-120 (Coordinate Frames for Humanoid Robots) the root link is **base_link**, directly connected to **torso**. The camera frames are **CameraBottom_frame** and **CameraTop_frame**. nao_remote publishes the **odom -> base_link** transform and a **base_footprint** frame, projected between the feet on the ground. The defined links for the end effectors are **[l|r]_gripper** for the arms and **[l|r]_sole** for the feet.

SoftBank Robotics

2. NAO description in ROS

Task 2: URDF tutorial

- Install URDF tutorial

  $ sudo apt-get install ros-kinetic-urdf-tutorial
- Launch URDF tutorial with NAO model

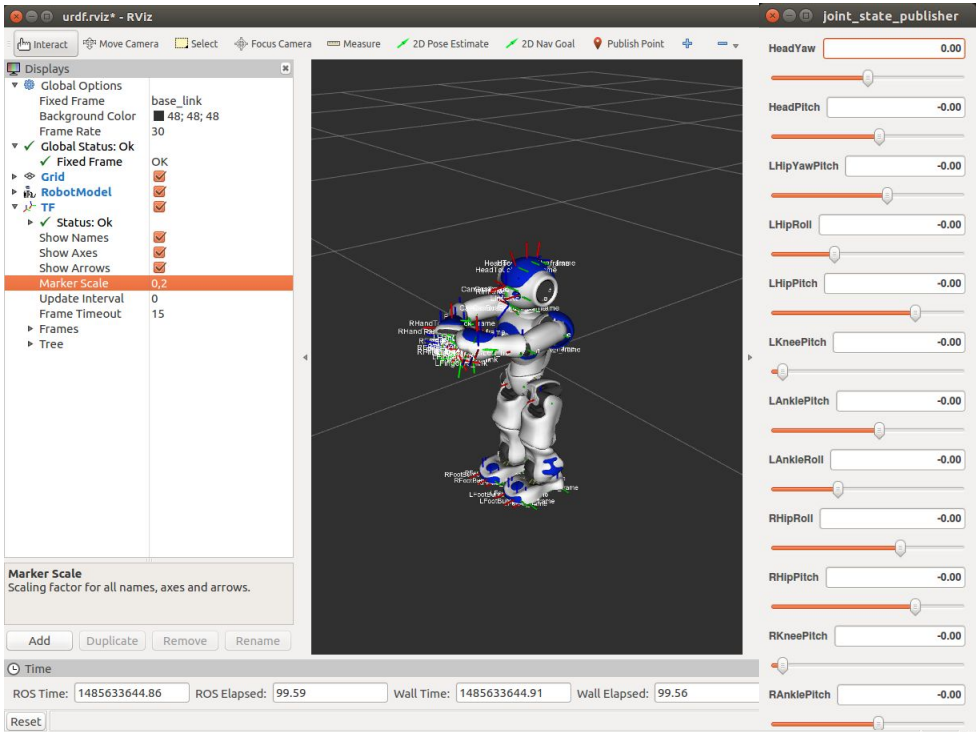
  $ roslaunch urdf_tutorial display.launch model:='$(find nao_description)/urdf/naoV50_generated_urdf/nao.urdf' gui:=true

In case of problems,

- source ROS

  $ source /opt/ros/kinetic/setup.bash
- check if roscore starts

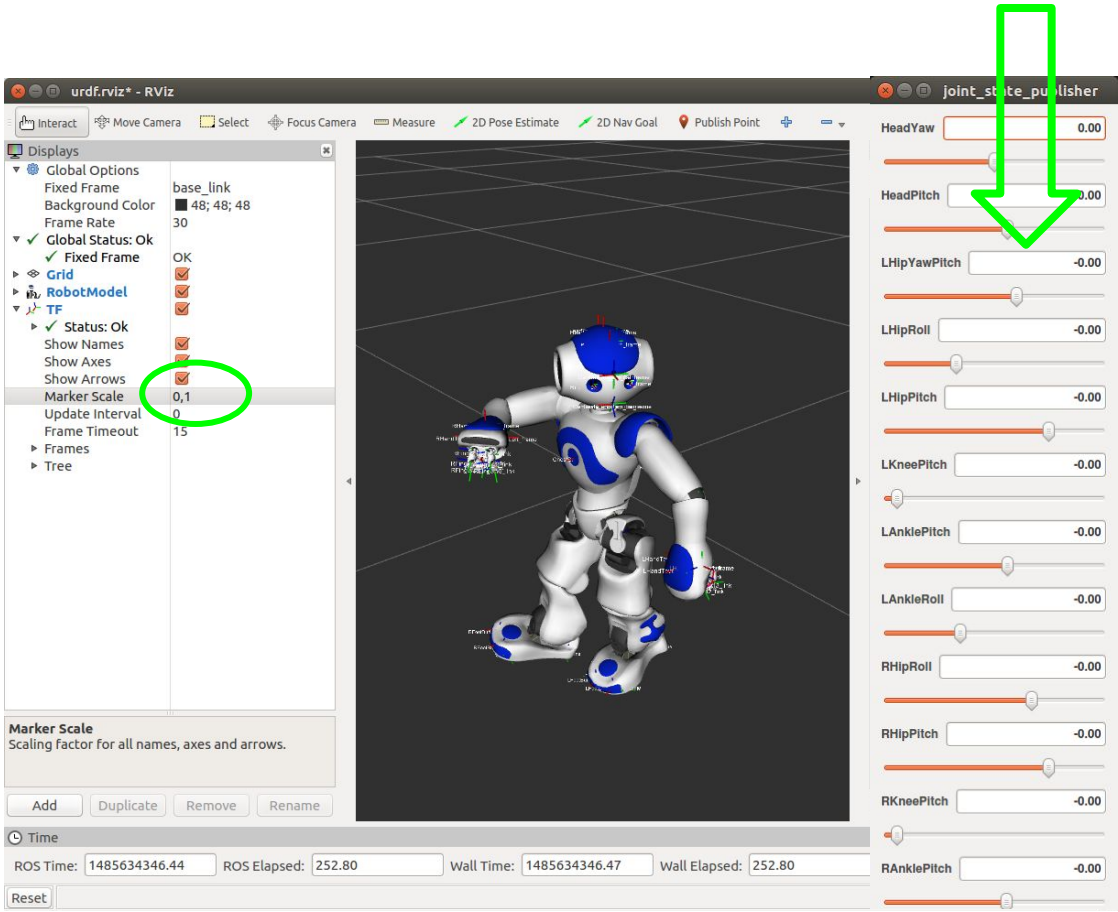
  $ roscore



2. NAO description in ROS

Task 2

- in RViZ
  - adapt TF params
    - Marker scale
  - interact with the robot
  - zoom in/out
  - change the view angle
- in GUI
  - move joints
- what is it useful for?
  - know joints limits and how do they move



SoftBank Robotics

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

3. ROS bridge

NAOqi

ROS

SoftBank Robotics

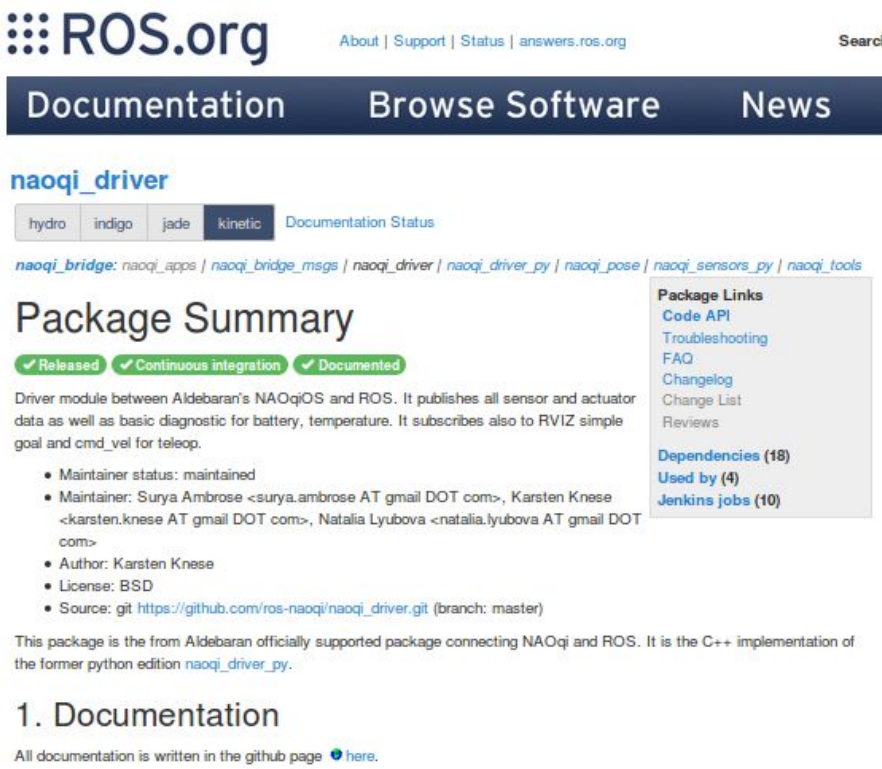
3. ROS bridge

- How to connect a real robot to ROS?
- How to access the robot's status and data in ROS?
- The useful packages
  - Naoqi_driver: common for Nao, Pepper, Romeo robots
  - Nao_bringup: Nao-specific
- which data can we access through ROS bridge (e.g. Naoqi Driver)?
  - actuator data
  - sensor data
  - basic diagnostic: battery, temperature of motors
  - some outcome data from NAOqi modules, e.g. detected people, faces, ...
- which data can we control?
  - robot velocity: /cmd_vel
  - move to target: /move_base_simple/goal
  - joints: /joint_angles

3. ROS bridge: Naoqi Driver

More about Naoqi Driver?

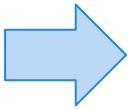
- doc: http://ros-naoqi.github.io/naoqi_driver
- can be executed
  - remotely on PC
  - locally on a robot
- install:
  - $ sudo apt-get install ros-kinetic-naoqi-driver



SoftBank Robotics

3. ROS bridge:

NAOqi



ROS

3. ROS bridge: Naoqi Driver

Task 3.1: connecting NAO to ROS

- PLAN A: you have Nao, then start Naoqi Driver
  $ roslaunch naoqi_driver naoqi_driver.launch nao_ip:=<robot_IP> network_interface:=<eth0 | wlan0 | tethering | vpn>
  - nao_ip: IP address of the robot
  - network_interface: network connection on your PC

- PLAN B: if you do not have a robot
  - download ROSmaterials/task3.1 and move it to /tmp
    - *.bag
    - *.launch
  - $ cd /tmp/ROSmaterials/task3
  - play the rosbag by launching run_rosbag.launch with the full path as an argument
    $ roslaunch run_rosbag.launch rosbag:=`pwd`/basic_sit.bag
- in both cases, start RViZ
  $ rosrun rviz rviz

SoftBank Robotics

3. ROS bridge: Naoqi Driver

Useful environment variables, when connecting to a real robot:

- the robot's IP

  export NAO_IP=<robot_IP>

- rosmaster IP

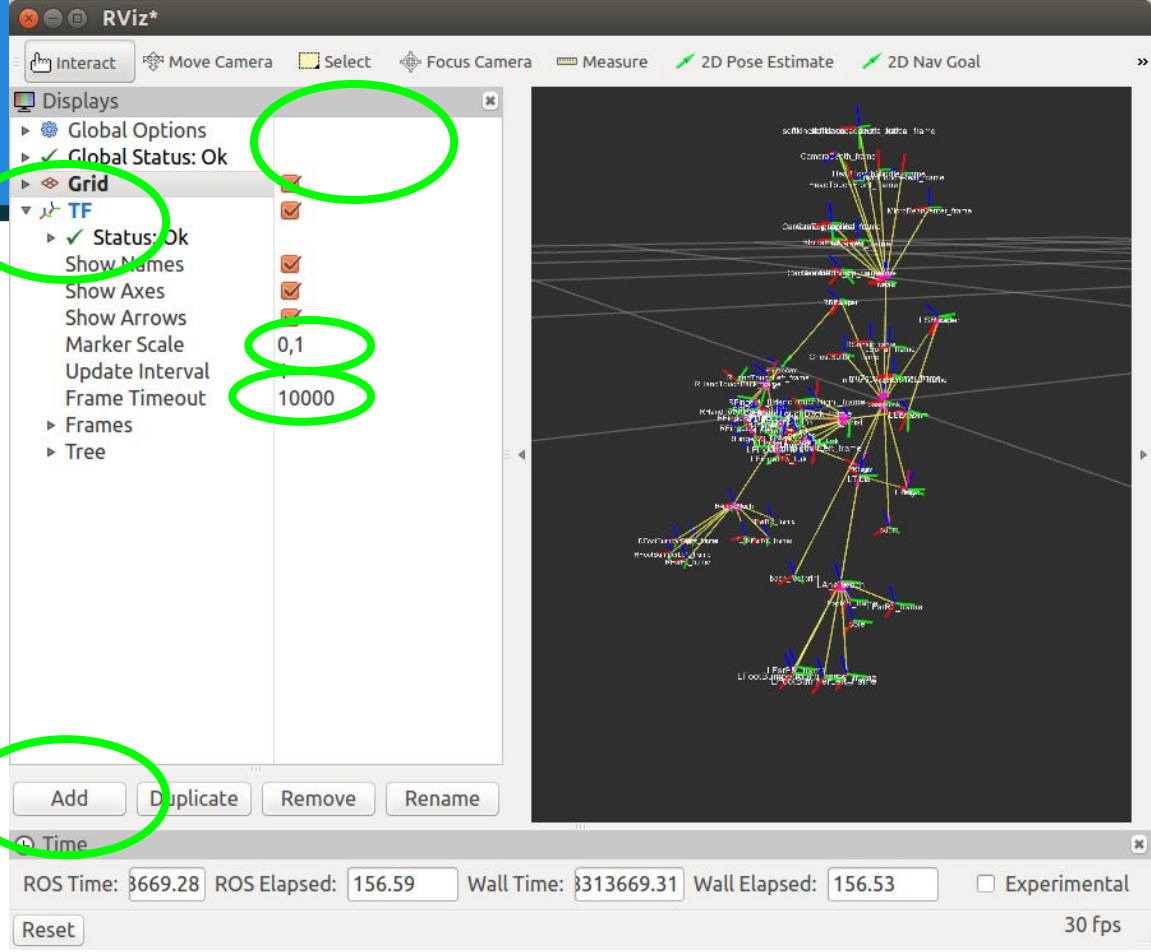
  export ROS_MASTER_URI=http://<roscore_IP>:11311

3. ROS bridge: Naoqi Driver

Task 3.1

Setup RViZ

- set GlobalOptions -> FixedFrame
  - choose odom
- add visualization plugins:
  - TF: transform tree
    - set Update Interval
    - set FrameTimeout



- if you do not see TF, then check if you install NAO description and relaunch

  $ sudo apt-get install ros-kinetic-nao-description

  $ source /opt/ros/kinetic/setup.bash

SoftBank Robotics

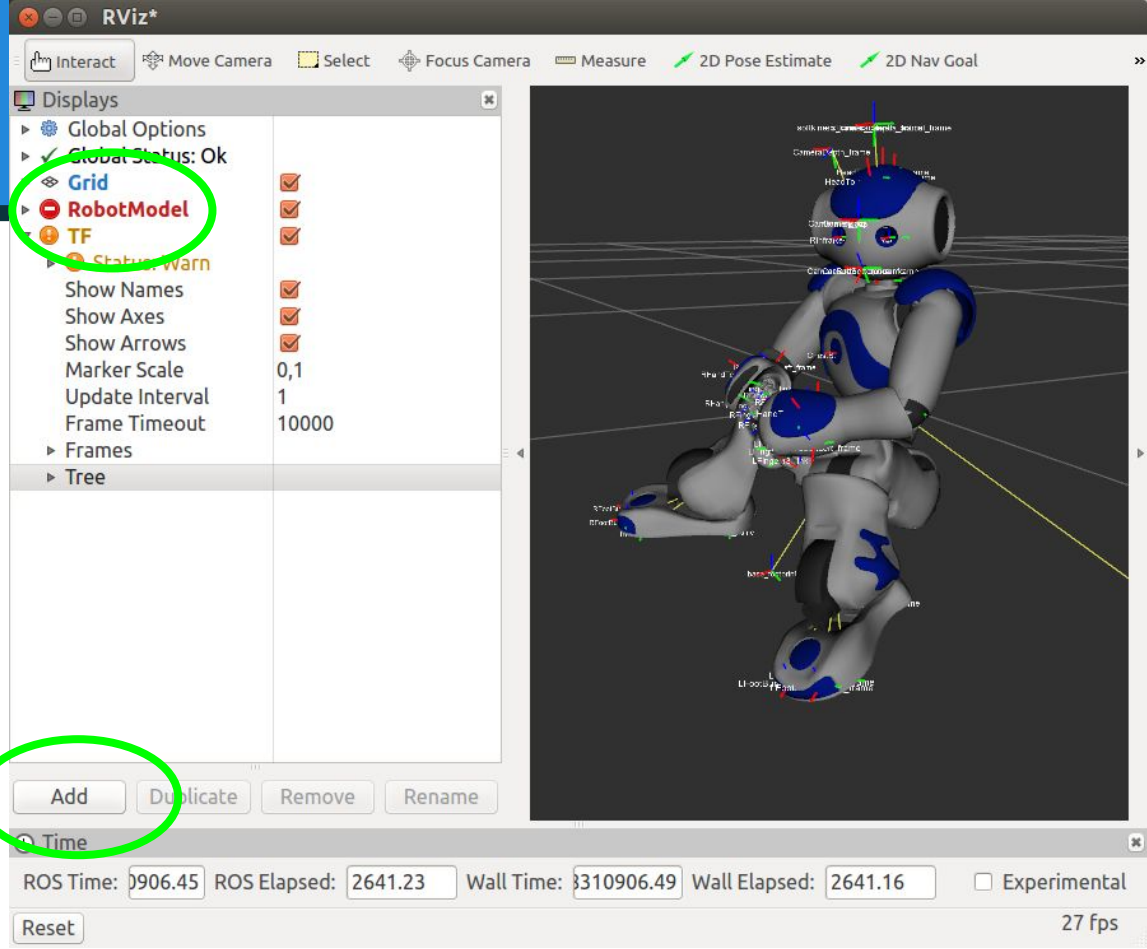
3. ROS bridge: Naoqi Driver

Task 3.1

Setup RViZ

- add visualization plugins:
  - RobotModel

- if you do not see the robot, then check if you installed NAO meshes and relaunch




```
$ sudo apt-get install ros-kinetic-nao-meshes
$ source /opt/ros/kinetic/setup.bash
```

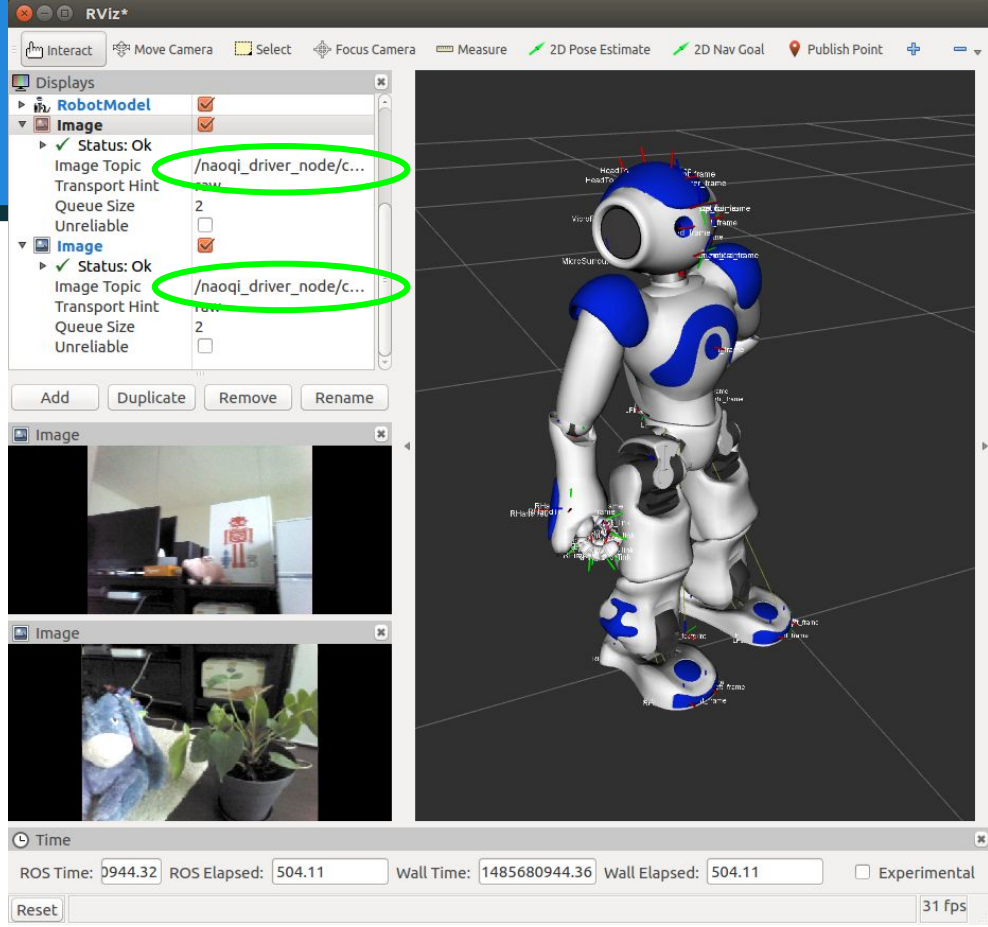
SoftBank Robotics

3. ROS bridge: Naoqi Driver

Task 3.1

In RViZ,
visualize data from robot's sensors

- add visualization plugins:
  - Image
    - set a front camera topic
  - Image
    - set a bottom camera topic



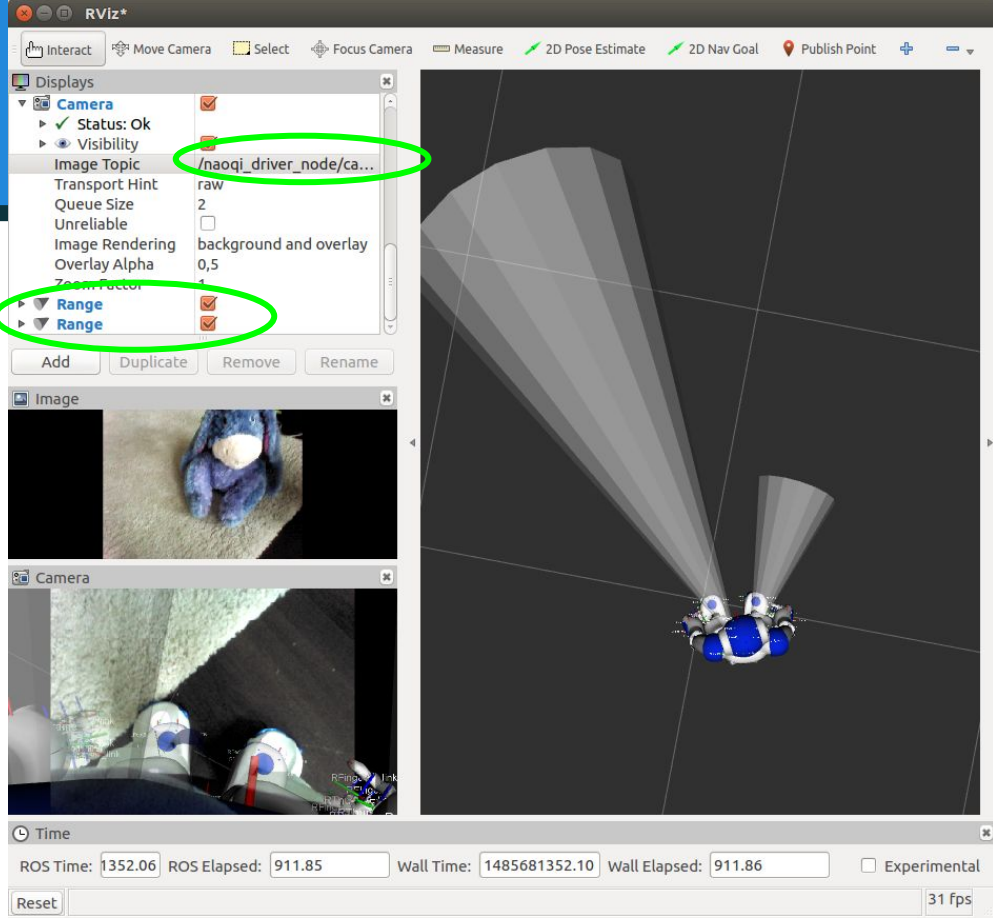
SoftBank Robotics

3. ROS bridge: Naoqi Driver

Task 3.1

In RViZ,
visualize data from robot's sensors

- add visualization plugins:
  - Camera
    - set a camera topic
  - Range
    - set sonar topics



SoftBank Robotics

3. ROS bridge: Naoqi Driver

Task 3.1

Check useful command-line tools:

- check topics
  - $ rostopic list
  - $ rostopic echo /tf
  - $ rostopic echo /joint_states
  - $ rostopic echo /camera/front/camera_info
  - $ rostopic info /camera/front/camera_info
  - $ rosnode list
- check tf
  - $ rosrun tf tf_monitor -> all frames
  - $ rosrun tf tf_echo <frame1> <frame2> -> position of one frame to another
- record data and play them
  - $ rosbag record -a -> record all topics into a rosbag
  - $ rosbag play yourbagname.bag -> play teh rosbag

3. ROS bridge: Naoqi Driver

Task 3.1

- Check available topics:
  $ rostopic list
  - topics of Naoqi Driver are:

clicked_point
/cmd_vel
/diagnostics
/initialpose
/joint_angles
/joint_states
/naoqi_driver_node/camera/bottom/camera_info
/naoqi_driver_node/camera/bottom/image_raw
/naoqi_driver_node/camera/bottom/image_raw/compressed
/naoqi_driver_node/camera/bottom/image_raw/compressed/parameter_descriptions
/naoqi_driver_node/camera/bottom/image_raw/compressed/parameter_updates
/naoqi_driver_node/camera/bottom/image_raw/theora
/naoqi_driver_node/camera/bottom/image_raw/theora/parameter_descriptions
/naoqi_driver_node/camera/bottom/image_raw/theora/parameter_updates
/naoqi_driver_node/camera/front/camera_info
/naoqi_driver_node/camera/front/image_raw
/naoqi_driver_node/camera/front/image_raw/compressed
/naoqi_driver_node/camera/front/image_raw/compressed/parameter_descriptions
/naoqi_driver_node/camera/front/image_raw/compressed/parameter_updates
/naoqi_driver_node/camera/front/image_raw/theora
/naoqi_driver_node/camera/front/image_raw/theora/parameter_descriptions
/naoqi_driver_node/camera/front/image_raw/theora/parameter_updates
/move_base_simple/goal
/naoqi_driver_node/audio
/naoqi_driver_node/bumper
/naoqi_driver_node/hand_touch
/naoqi_driver_node/head_touch
/naoqi_driver_node/imu/torso
/naoqi_driver_node/info
/naoqi_driver_node/odom
/naoqi_driver_node/sonar/left
/naoqi_driver_node/sonar/right
/rosout
/rosout_agg
/speech
/tf
/tf_static

3. ROS bridge: Naoqi Driver

Task 3.1

- Check data published on topics, for example:
  $ rostopic echo /joint_states


```
header:
  seq: 11054
  stamp:
        secs: 1485635198
        nsecs: 564073085
  frame_id: ''
name: ['HeadYaw', 'HeadPitch', 'LHipYawPitch', 'LHipRoll', 'LHipPitch', 'LKneePitch', 'LAnklePitch', 'LAnkleRoll', 'RHipYawPitch', 'RHipRoll',
'RHipPitch', 'RKneePitch', 'RAnklePitch', 'RAnkleRoll', 'LShoulderPitch', 'LShoulderRoll', 'LElbowYaw', 'LElbowRoll', 'LWristYaw', 'LHand',
'RShoulderPitch', 'RShoulderRoll', 'RElbowYaw', 'RElbowRoll', 'RWristYaw', 'RHand', 'RFinger13', 'RFinger12', 'LFinger21', 'LFinger13',
'LFinger11', 'RFinger22', 'LFinger22', 'RFinger21', 'LFinger12', 'RFinger23', 'RFinger11', 'LFinger23', 'LThumb1', 'RThumb1', 'RThumb2',
'LThumb2']
position: [1.1366901499999997, -9.093359999989836e-05, -0.41238729119999995, -0.0001550409999999669, 0.4839800000000001,
-0.00016400378000000493, -0.00016097489999999937, -1.490230000000814e-05, -0.41238729119999995, -0.0001959274999999705,
-0.00038482599999989375, -0.00016400378000000493, -4.8639999999711137e-05, -0.34324383030000005, 1.492505452, 0.9143290192,
0.0, -0.7897633000000001, 0.0, 0.0, 0.0, -1.7623500000008008e-05, 0.0, 0.7897633000000001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
velocity: []
effort: []
```


SoftBank Robotics

3. ROS bridge:

SoftBank Robotics

3. ROS bridge: Nao Bringup

Task 3.2: connecting a simulated NAO to ROS

1. download ROSmaterials/Task3.2
   - move to /tmp
   - extract C++ and Python SDK
2. start Naoqi standalone
   - $ cd /tmp/ROSmaterials
   - $ naoqi-sdk-2.1.4.13-linux64/naoqi --verbose --broker-ip 127.0.0.1
3. Start ROS bridge
   - $ export PYTHONPATH=/opt/ros/kinetic/lib/python2.7/dist-packages:/tmp/ROSmaterials/pynaoqi-python2.7-2.4.3.28-linux64
   - $ roslaunch nao_bringup nao_full_py.launch
4. Start RViZ
   - $ cd /tmp/ROSmaterials/Task3.2
   - $ roslaunch run_rviz.launch

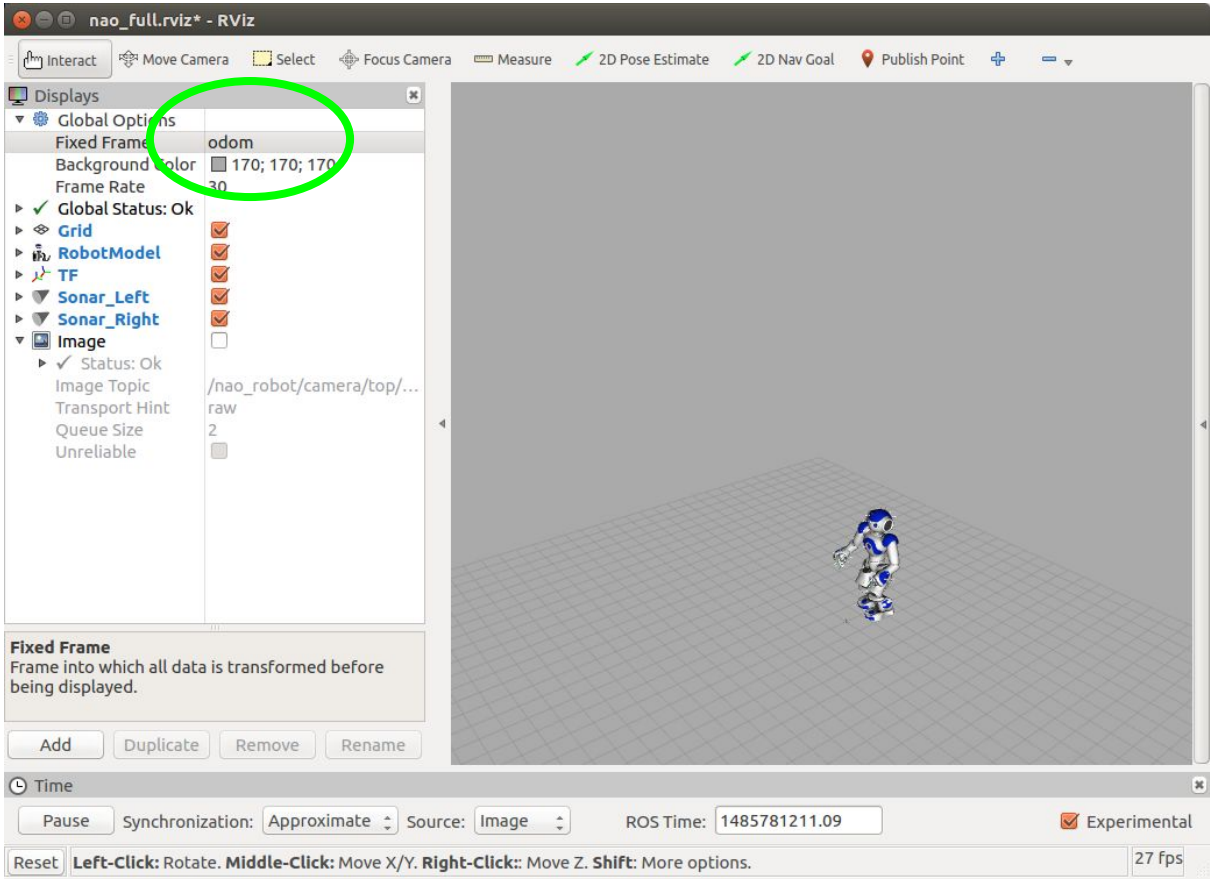
SoftBank Robotics

3. ROS bridge: Nao Bringup

Task 3.2

in RViZ

- set Global options > Fixed Frame
  - odom



3. ROS bridge: Nao Bringup

Task 3.2

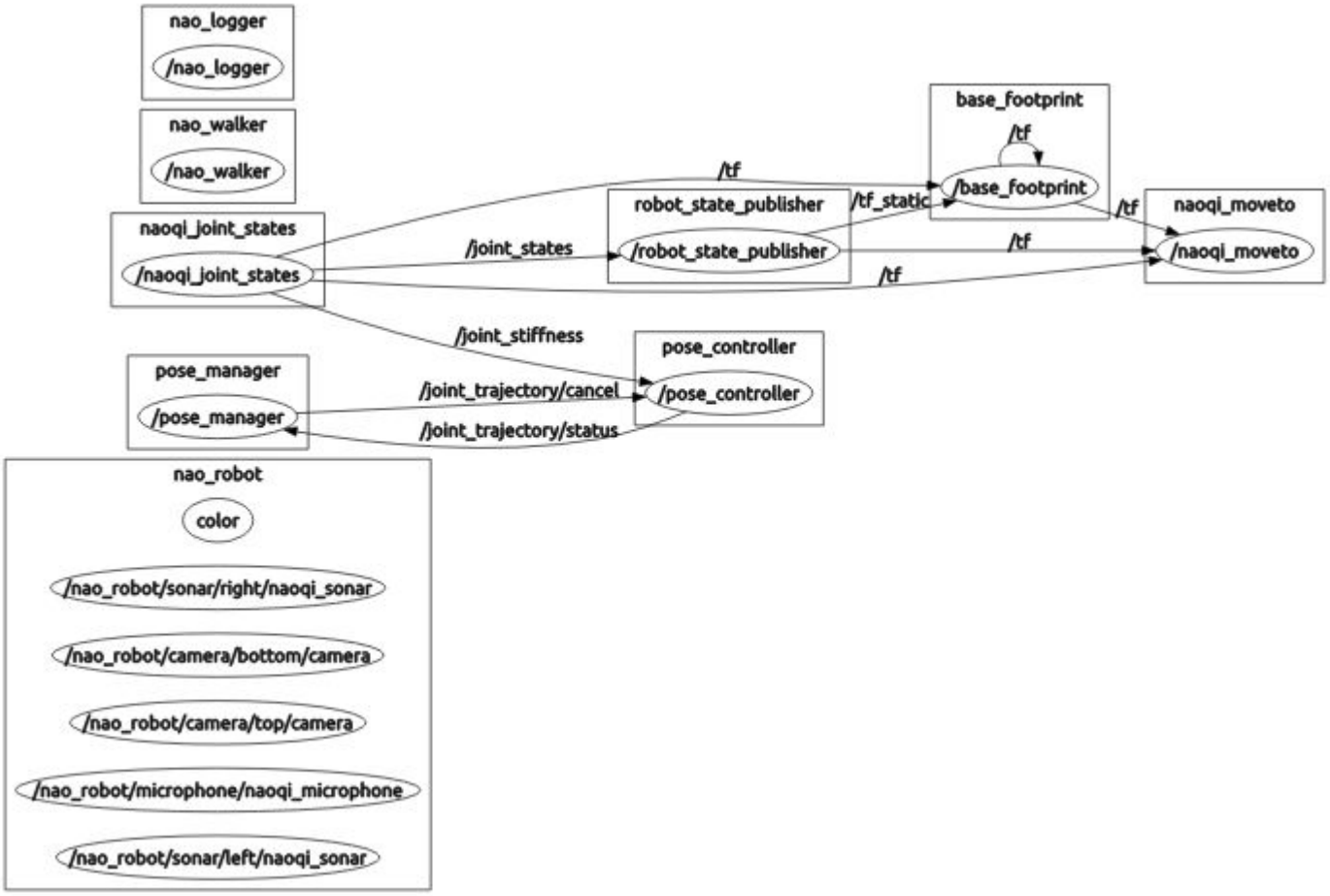
Check useful command-line tools:

- check topics
  - $ rostopic list
  - $ rostopic echo /tf
  - $ rostopic echo /joint_states
  - $ rostopic echo /camera/front/camera_info
  - $ rostopic info /camera/front/camera_info
  - $ rosnode list
- check tf
  - $ rosrun tf tf_monitor -> all frames
  - $ rosrun tf tf_echo <frame1> <frame2> -> position of one frame to another
- record data and play them
  - $ rosbag record -a -> record all topics into a rosbag
  - $ rosbag play yourbagname.bag -> play the rosbag

3. ROS bridge: Nao Bringup

Task 3.2

- Check available nodes and topics:



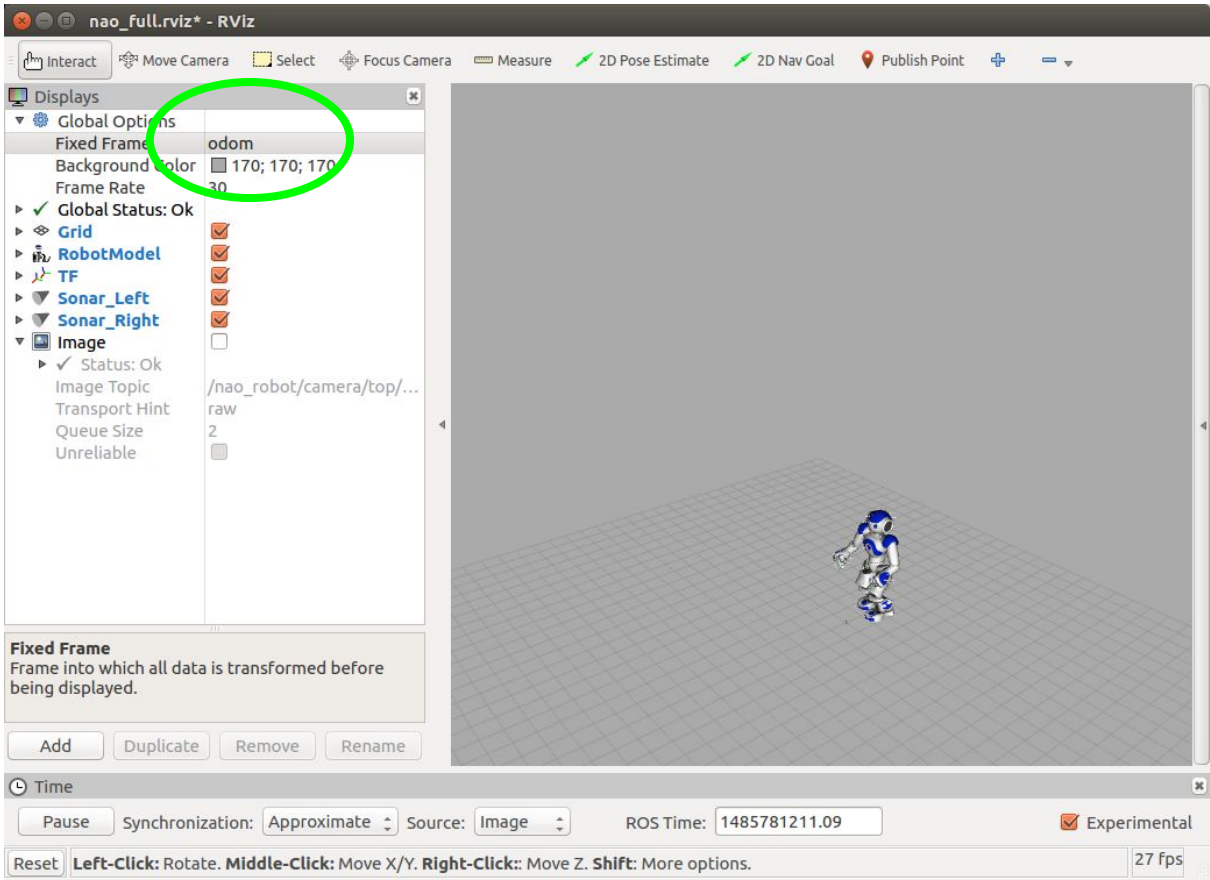
SoftBank Robotics

3. ROS bridge: Nao Bringup

Task 3.2

Test walker node

- $ rostopic pub -1 /cmd_vel geometry_msgs/Twist '{linear: {x: 1.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}'
- check if Nao is walking

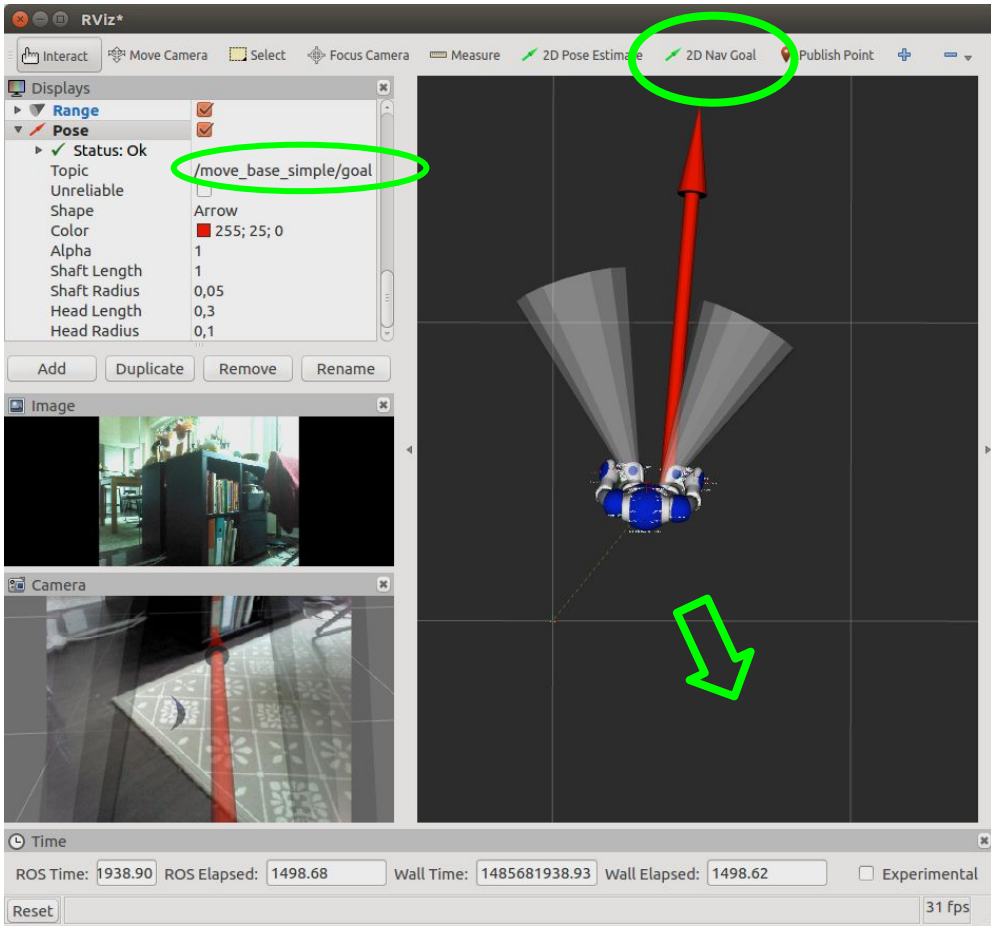


3. ROS bridge: Nao Bringup

Task 3.2

In RViZ, ask the robot to walk

- In RViZ add plugins:
  - Pose
    - move_base_simple/goal
- set a goal to move
  - click on 2D NavGoal
  - click on a point in space and turn to choose orientation



SoftBank Robotics

3. ROS bridge: Nao Bringup

Task 3.2

- Record a rosbag

  $ rostopic record -a
- Play the rosbag

  $ rostopic play <name>.bag

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
    - connect NAO to ROS
    - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
    - Object recognition
    - MoveIt!
    - Teleop
    - Gazebo

4. Visual sensors

List of sensors supported in ROS: http://wiki.ros.org/Sensors

Examples of most recent 3D sensors:

- ASUS Xtion or Kinect sensor
  - check: http://dlcdnet.asus.com
- Creative senz3d
  - check: https://github.com/ipa320/softkinetic
- Intel SR200 and F200
  - check: https://github.com/intel-ros/realsense
- Intel R200
  - check: https://github.com/intel-ros/realsense

4. Visual sensors

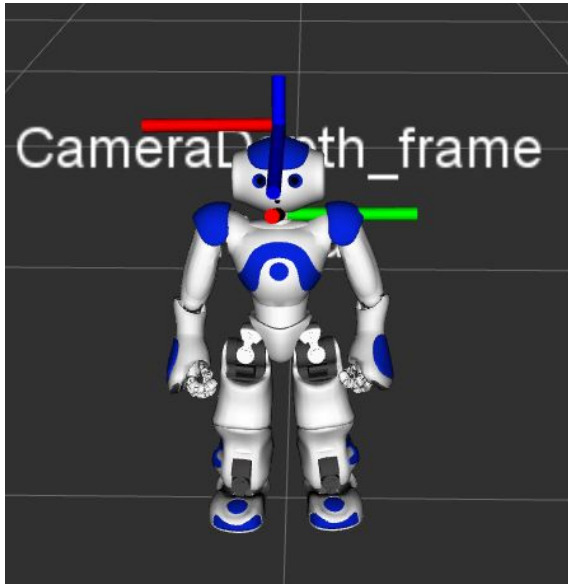
How to integrate a sensor into a robot model?

- add to URDF of the robot
  - position/orientation of the sensor
  - child/parent links

4. Visual sensors

How to integrate a sensor into a robot model?

- add to URDF of the robot
  - position/orientation of the sensor
  - child/parent links



Example:

```
<link name="CameraDepth_frame">
    <inertial>
        <mass value="0.05" />
        <origin xyz="0 0 0" />
        <inertia ixx="1e-9" ixy="0" ixz="0" iyy="1e-9" iyz="0" izz="1e-9" />
    </inertial>
</link>
<joint name="CameraDepth_joint" type="fixed">
    <parent link="Head"/>
    <child link="CameraDepth_frame"/>
    <origin rpy="1.05 3.14159265 1.57079632" xyz="-0.01 0.0 0.13"/>
</joint>
```

4. Visual sensors

How to get visual data?

- install the driver

  $ sudo apt-get install ros-kinetic-openni2-launch

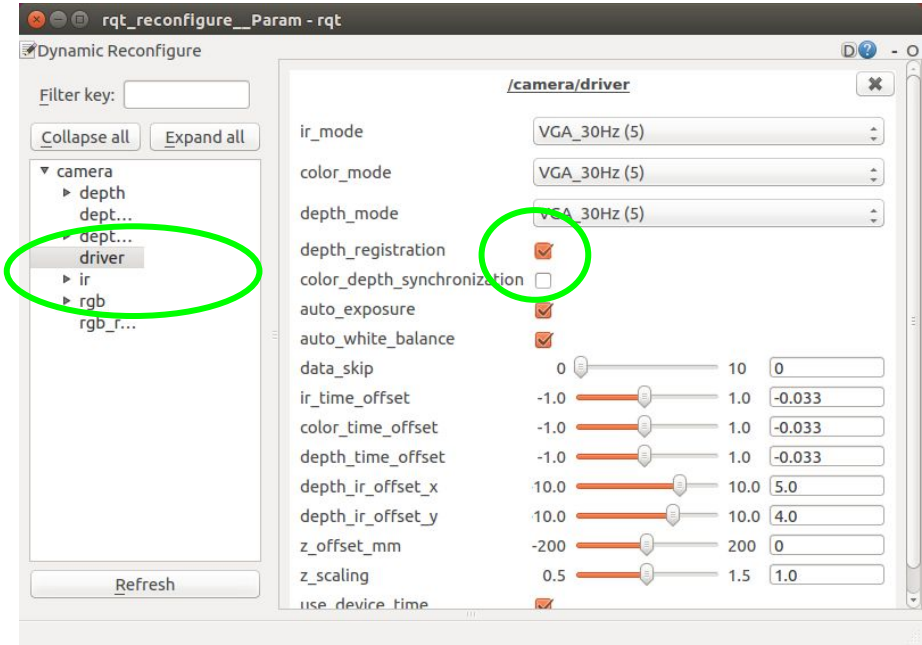
- start the driver, e.g. ASUS Xtion:

  $ roslaunch openni2_launch openni2.launch

  depth_frame_id:=/CameraDepth_frame rgb_frame_id:=/CameraDepth_frame

- start image registration:

  $ rosrun rqt_reconfigure rqt_reconfigure



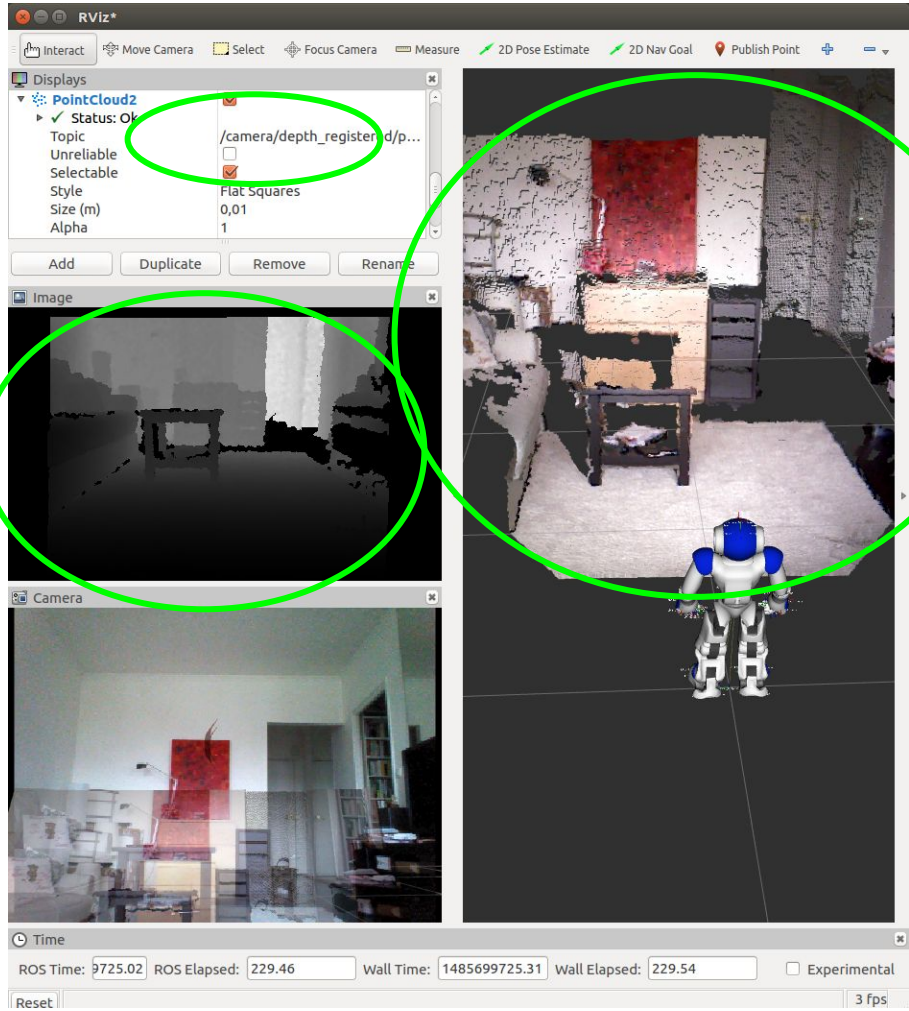
4. Visual sensors

How to get data relative to the robot?

- start ROS bridge

  $ roslaunch naoqi_driver naoqi_driver nao_ip:=<robot_IP>
- start RViZ

  $ rosrun rviz rviz
- In RViZ, add plugins to visualize depth data
  - Images:
    - set Xtion depth topic
  - DepthCloud
    - set Xtion depth topic
  - PointCloud2
    - set Xtion pointcloud topic



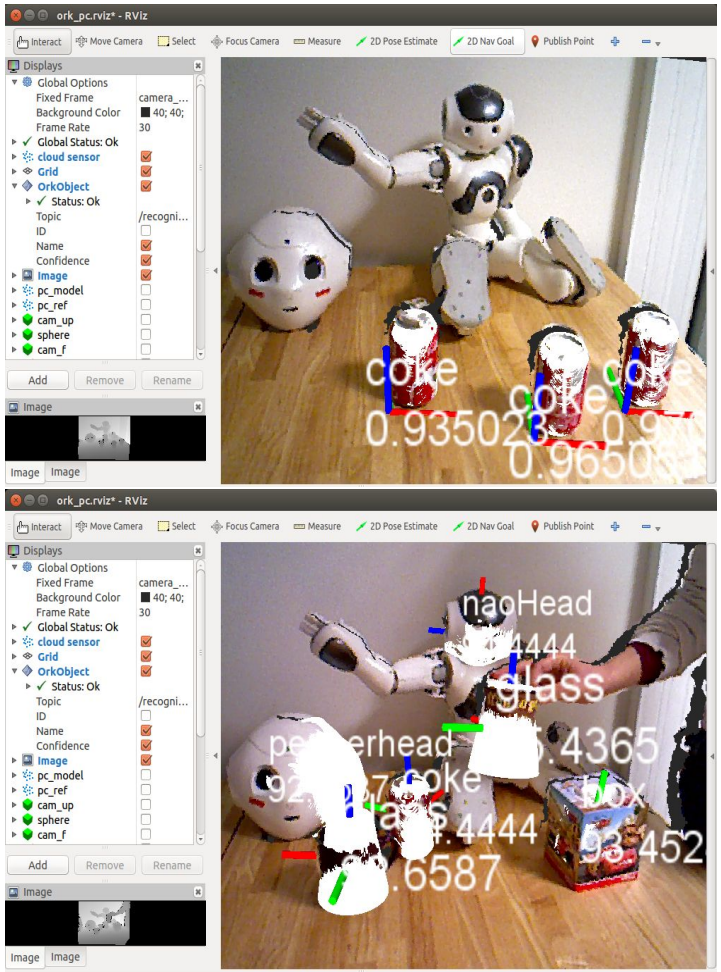
SoftBank Robotics

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

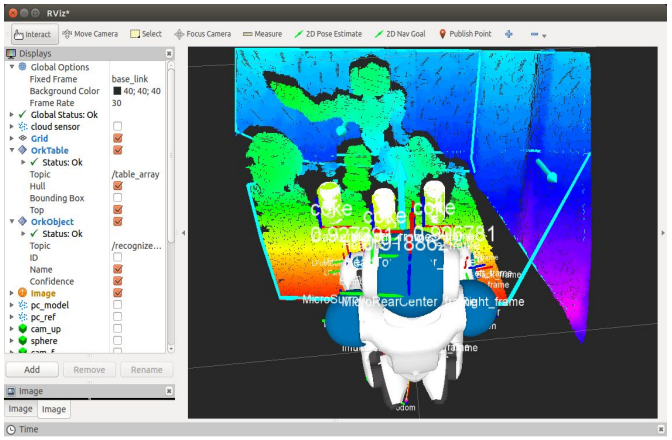
5.3 Object recognition

- Object recognition kitchen (ORK) pipelines:
  - objects on a planar surface
    object_recognition_tabletop
  - objects out of plane
    object_recognition_linemod
  - transparent objects
    object_recognition_transparent
  - textured objects
    object_recognition_tod
- ecto-based computations
- fast computation based on acyclic graphs
- object DB management
- i/o handling
- Doc:
  http://wg-perception.github.io/object_recognition_core



5.3 Object recognition

- install:
  $ sudo apt-get install ros-kinetic-object-recognition-*
- launch providing config file:
  - $ roslaunch naoqi_driver naoqi_driver.launch
  - $ rosrun object_recognition_core detection -c detection.ros.ork
  - $ rosrun rviz rviz
- add visualization plugins:
  - OrkObject
  - OrkTable
- check object recognition

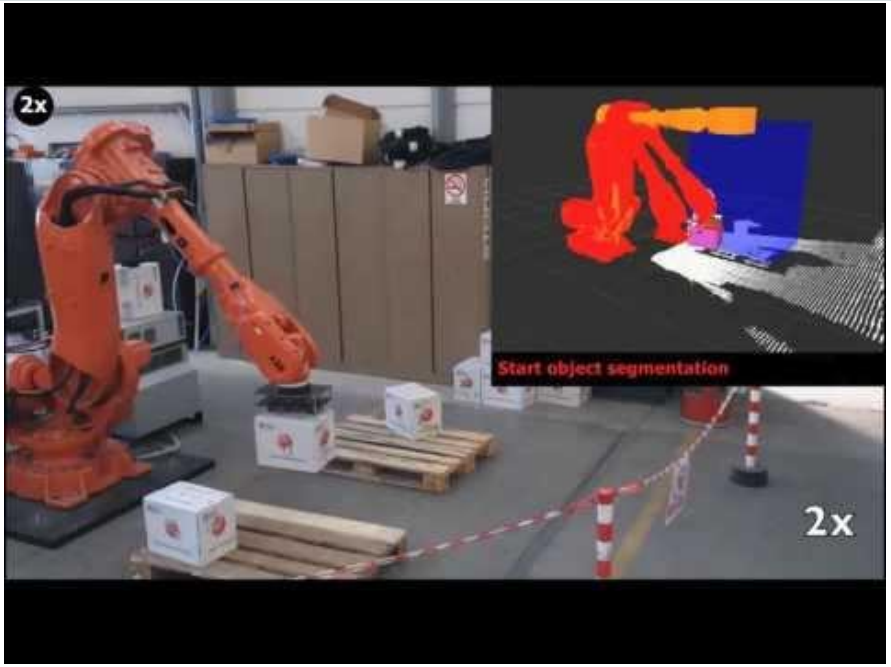


SoftBank Robotics

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

5.2 MoveIt!

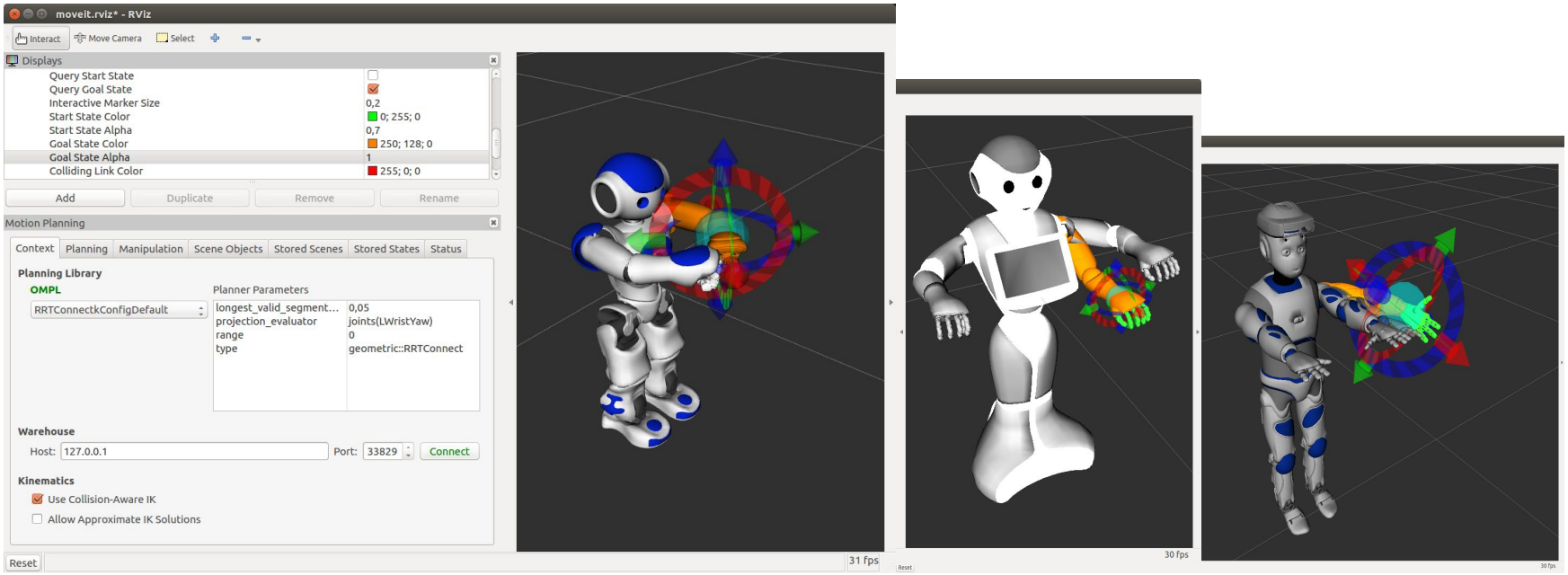


- set of tools for
  - mobile manipulation
  - 3D perception,
  - motion planning,
  - kinematics,
  - trajectory proceeding/execution
- > 65 robots
- the most widely used open-source for manipulation
- #3 ROS package people care about (after Rviz and Navigation)
- state of art
- Doc: http://moveit.ros.org
- Install: $ sudo apt-get install ros-kinetic-moveit

5.2 MoveIt!

How to use MoveIt with NAO?

- nao_moveit_config package
- Doc: http://wiki.ros.org/nao_moveit_config
- Install: $ sudo apt-get install ros-kinetic-nao-moveit



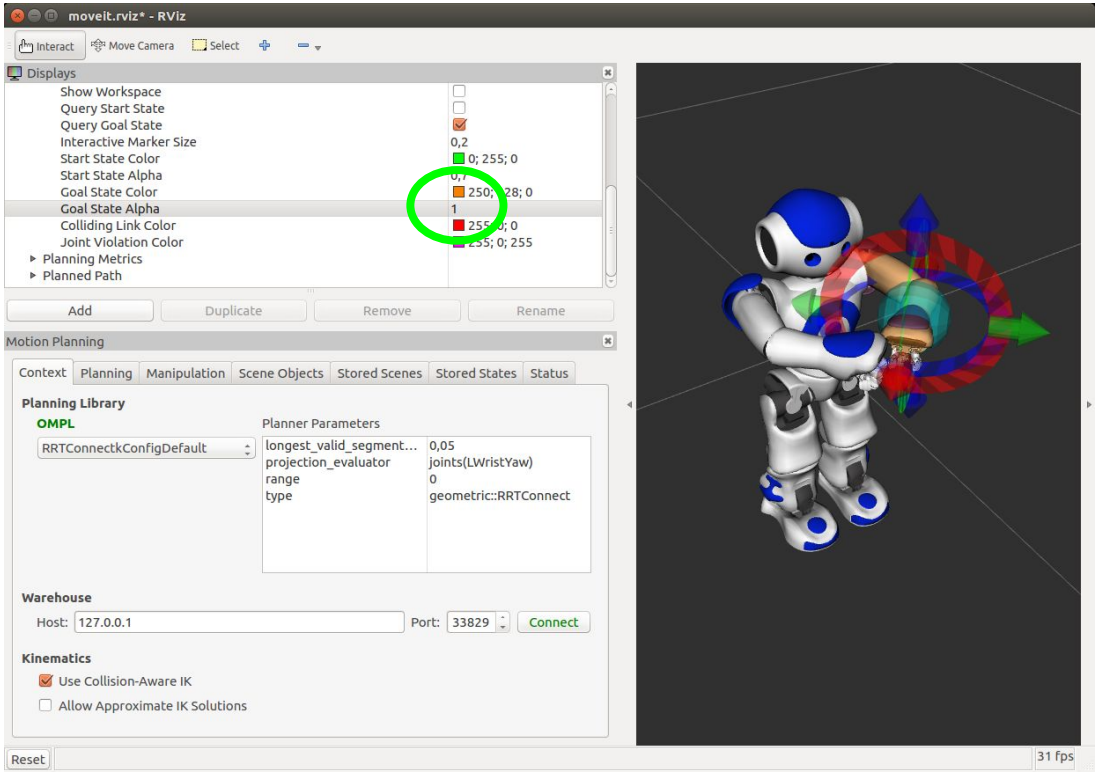
SoftBank Robotics

5.2 MoveIt!

Task 5.2: action planning with MoveIt!

- Start MoveIt with simulated NAO

  $ roslaunch nao_moveit_config demo.launch
- Visualize the robot Goal state
  - set Goal State Alpha = 1

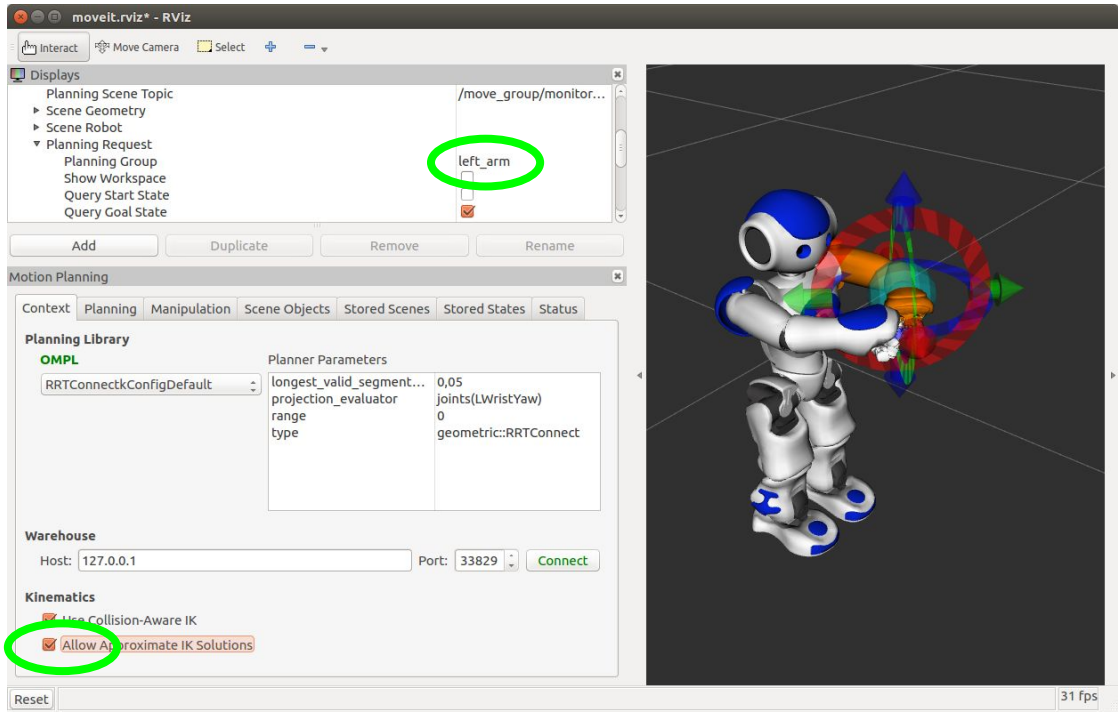


5.2 MoveIt!

Task 5.2

Try Action planning:

- in MotionPlanning plugin
- in Planning Request:
  - set a planning group (left arm or right arm)
  - choose Query Goal State or Start State
- in Context tab
  - activate “Allow approximate IK …”
  - optionally, choose a Planning library

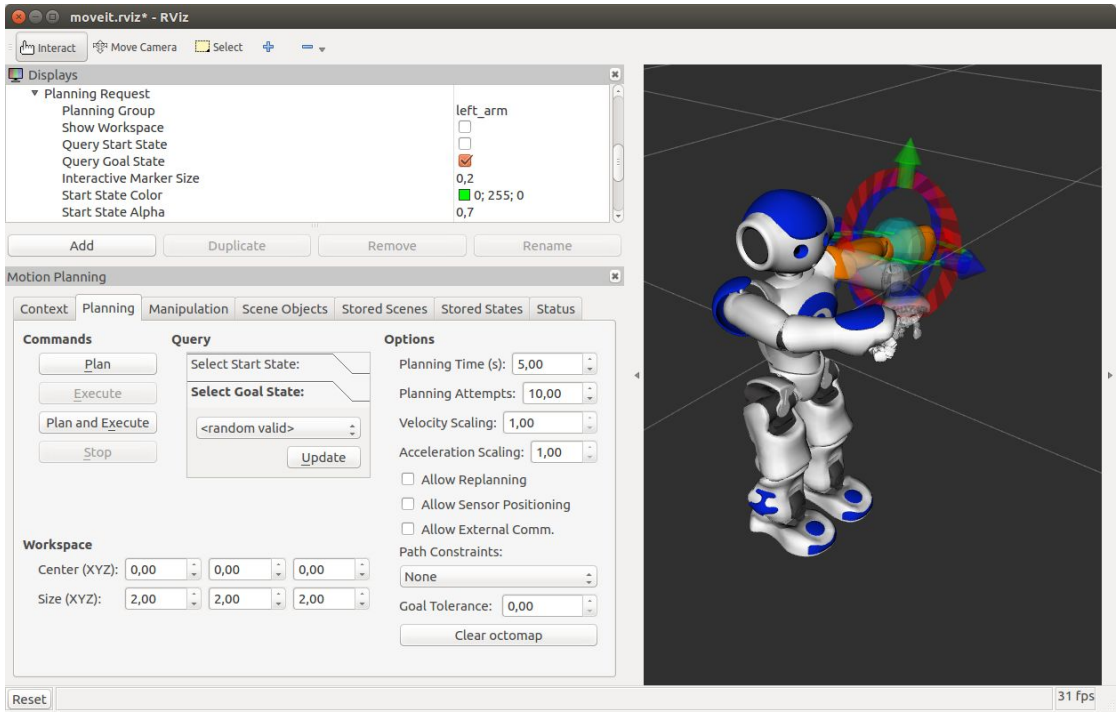


5.2 MoveIt!

Task 5.2

Try Action planning:

- in MotionPlanning plugin
- in Planning Request:
  - set a planning group (left arm or right arm)
  - choose Query Goal State or Start State
- in Context tab
  - activate “Allow approximate IK …”
  - optionally, choose a Planning library
- use the interactive marker
  - move/rotate or drag/drop

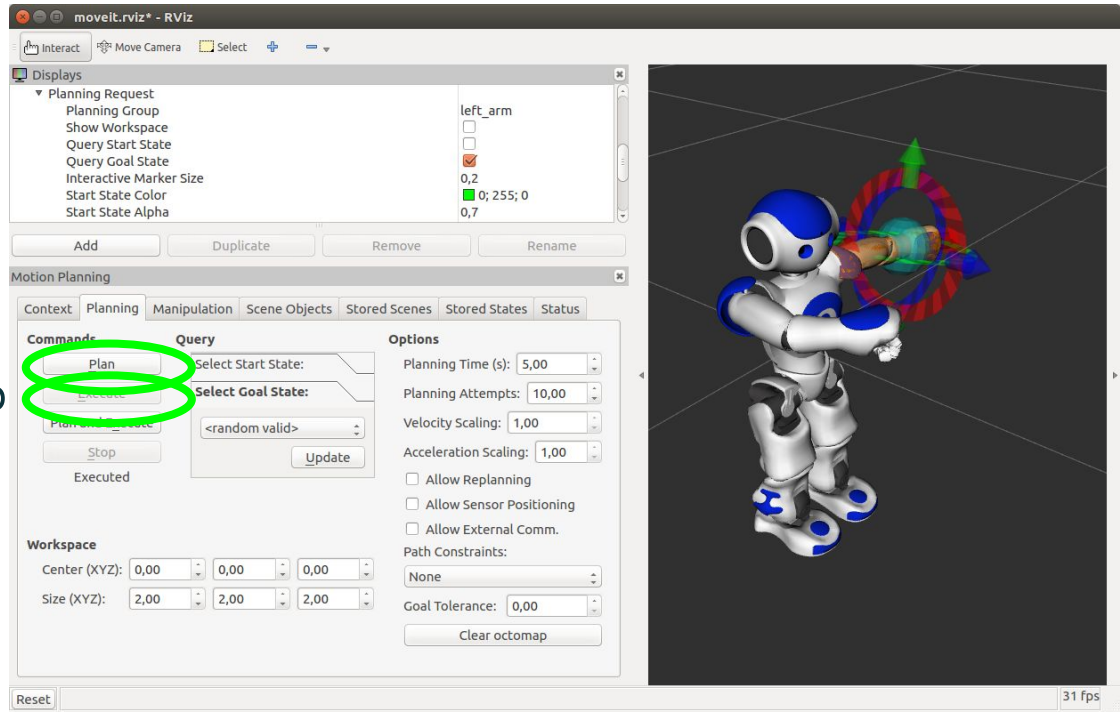


5.2 MoveIt!

Task 5.2

Try Action planning:

- in MotionPlanning plugin
- in Planning Request:
  - set a planning group (left arm or right arm)
  - choose Query Goal State or Start State
- in Context tab
  - activate “Allow approximate IK …”
  - optionally, choose a Planning library
- use the interactive marker
  - move/rotate or drug/drop
- in Planning tab
  - click “Plan” button and watch planning
  - click “Execute” button

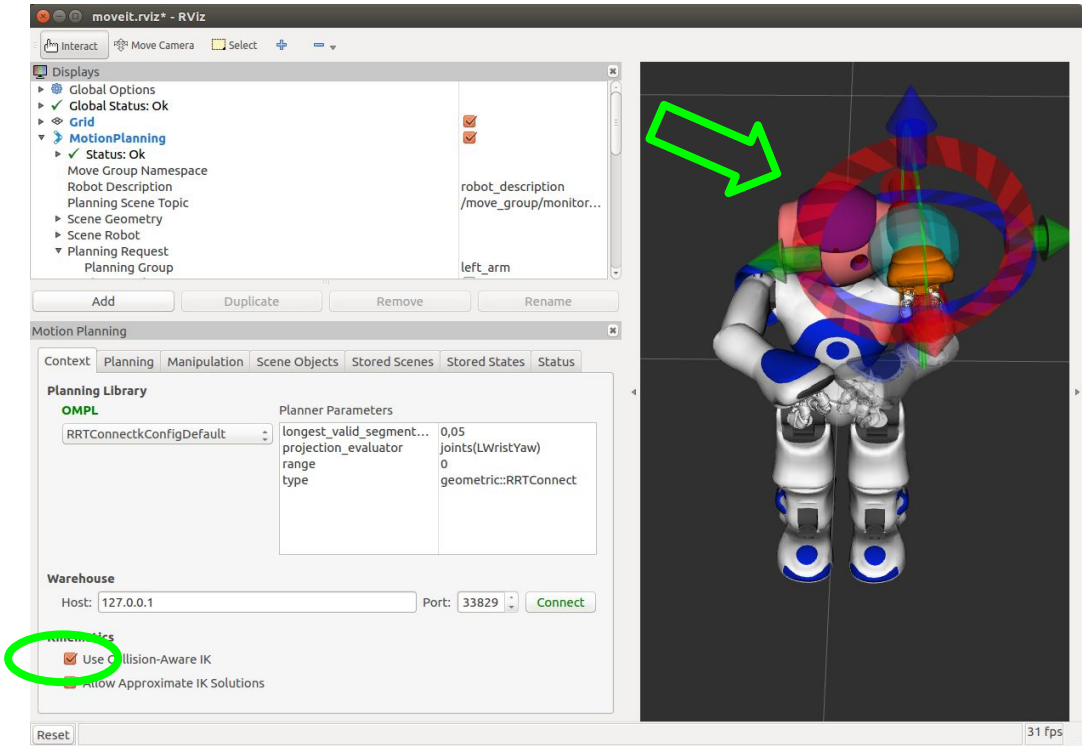


5.2 MoveIt!

Task 5.2

Try collision avoidance

- move the arm into collision
  - the two links that are in collision will turn red
  - the checkbox “Use Collision-Aware IK” forces solver to keep finding a collision-free solution

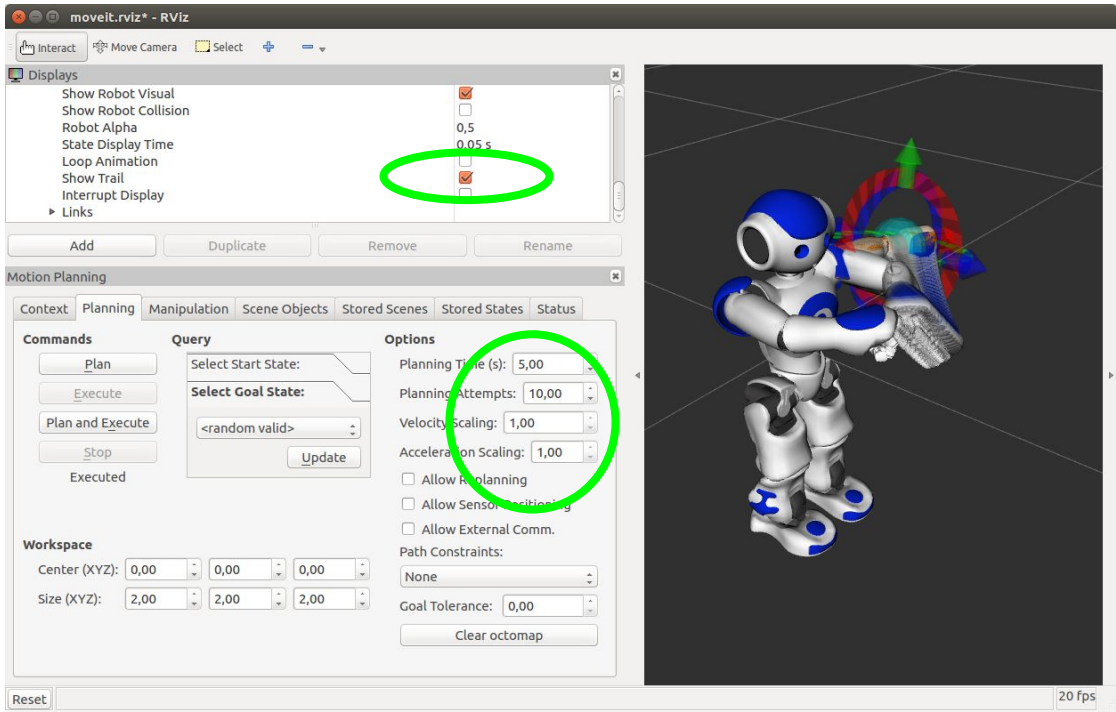


5.2 MoveIt!

Task 5.2

Try Action planning:

- in MotionPlanning plugin
  - Planned path
    - Show trail
    - Loop animation
- in Planning tab
  - Planning time
  - Velocity
  - Workspace



5.2 MoveIt!

How to control real NAO with MoveIt?

- Naoqi dcm driver
  - allow to control each robot joint
  - $ sudo apt-get install ros-kinetic-naoqi-dcm-driver
- Nao dcm bringup
  - allow to use ROS controllers with Nao
  - compile from source

    $ mkdir -p ~/catkin_ws/src && cd ~/catkin_ws/src

    $ git clone https://github.com/ros-naoqi/nao_dcm_robot.git

    $ git clone https://github.com/ros-naoqi/nao_virtual.git

    $ cd ../..

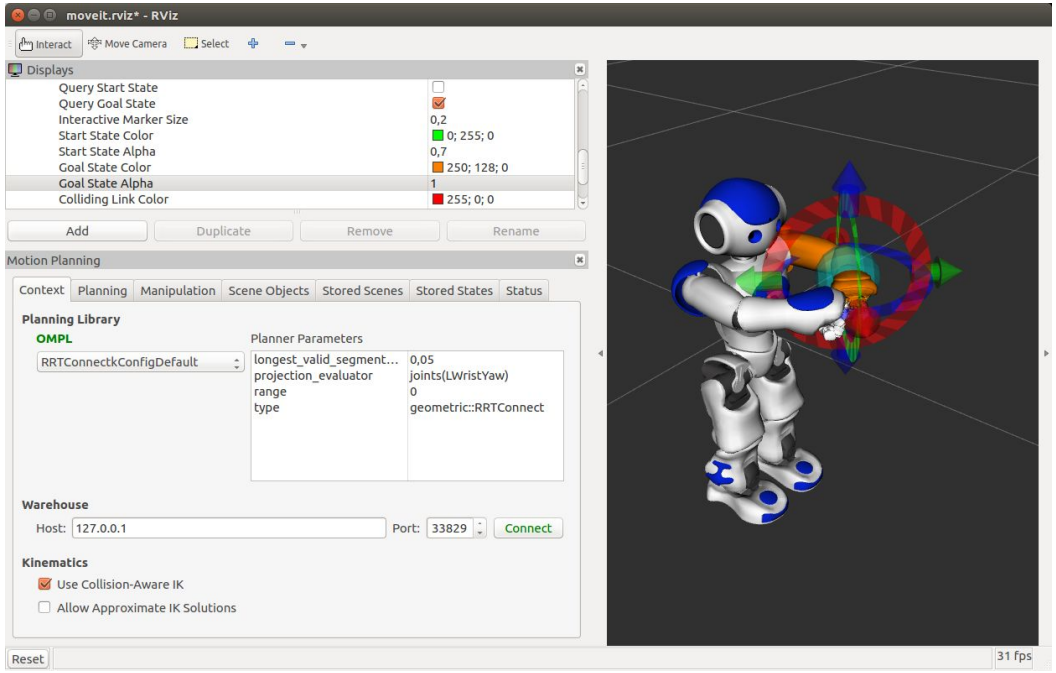
    $ catkin_make

    $ source devel/setup.bash

5.2 MoveIt!

How to control real NAO with MoveIt?

- Start DCM bringup
  $ roslaunch nao_dcm_bringup nao_dcm_H25_bringup_remote.launch
- Wait until all controllers are loaded and then, Start NAO with MoveIt
  $ roslaunch nao_moveit_config moveit_planner.launch
- enjoy controlling a real NAO

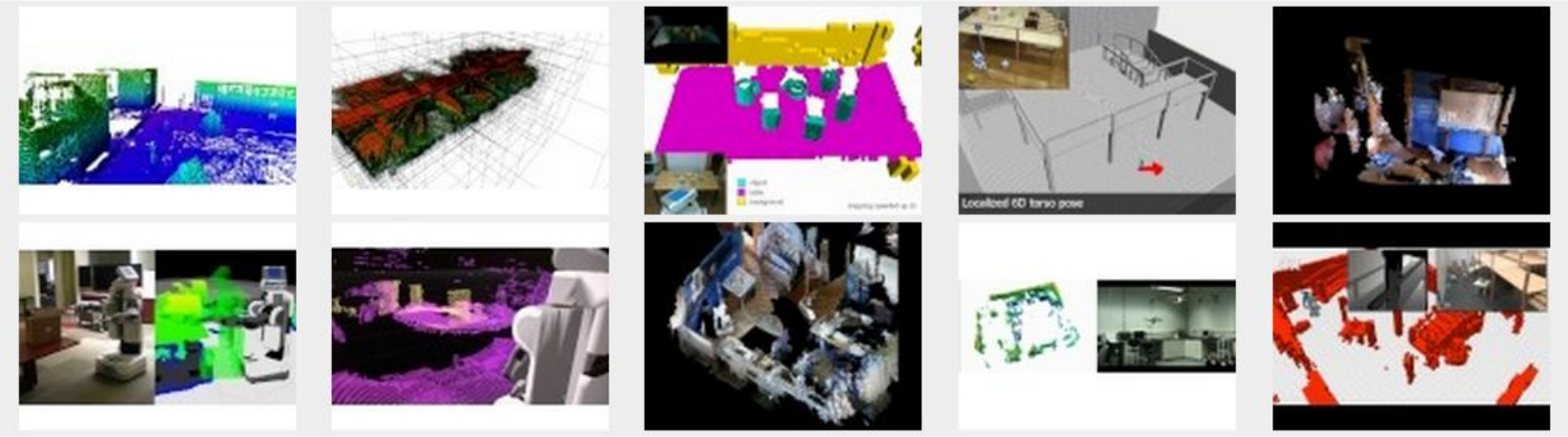


SoftBank Robotics

5.2 MoveIt! + Octomap

How to integrate MoveIt! and visual data?

- Octomap library
  - 3D occupancy grid
  - encodes probabilistic information about cells being occupied
  - fast computation based on octrees
  - doc: http://octomap.github.io
  - no need to install, it is already in MoveIt!

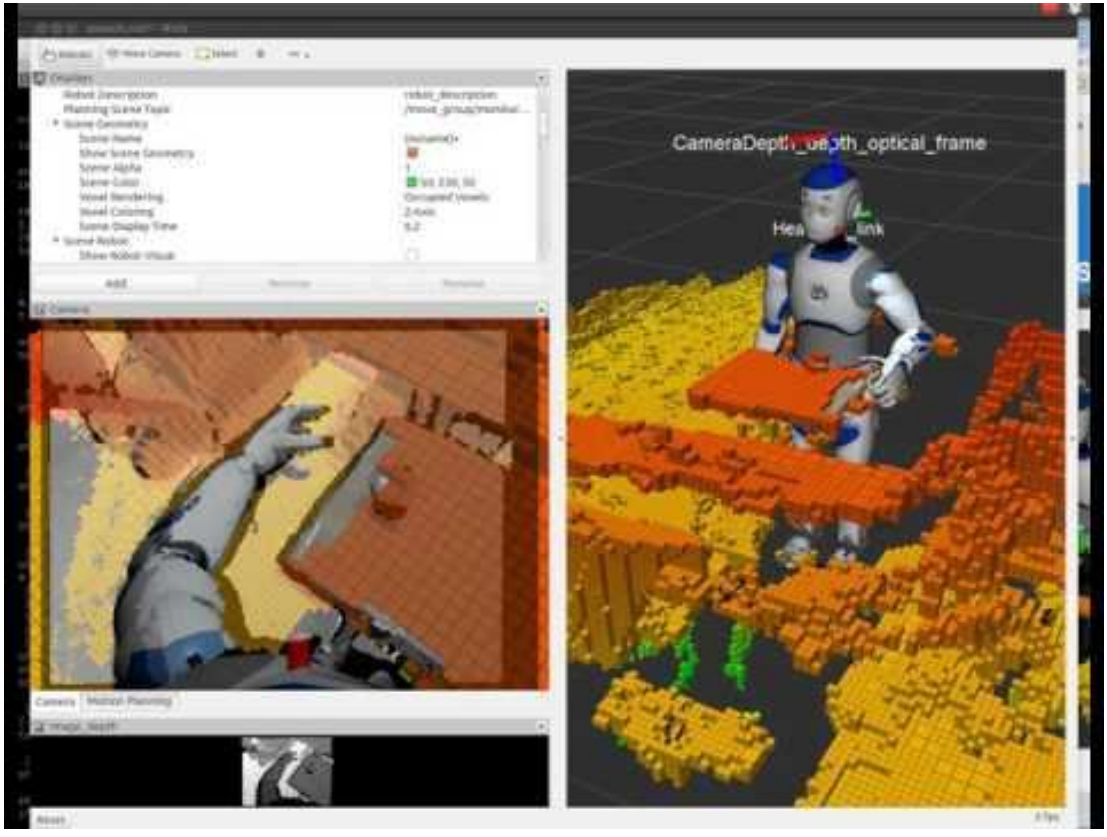


SoftBank Robotics

5.2 MoveIt! + Octomap

How to use Octomap with NAO?

- update the Octomap config file
  - $ rosed nao_moveit_config/config/sensors_xtion.yaml
  - set topics: depth image or pointcloud topic
- launch MoveIt! and check the occupancy grid



Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

5.1 Teleoperation

nao_teleop package

- allows to move the robot
  - with a gamepad
    - move/turn the base
    - move/turn the head
  - with Android Phone
    - move/turn the base
    - see a camera image
- works for Nao and Pepper
- doc: http://wiki.ros.org/nao_teleop
- How to launch on PC?

  $ roslaunch naoqi_driver naoqi_driver.launch ano_ip:=<robot_IP>

  $ roslaunch nao_teleop teleop_joy.launch

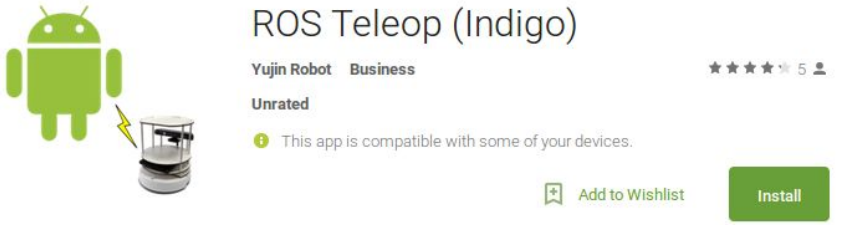
5.1 Teleoperation

nao_teleop package

- allows to move the robot
  - with a gamepad
    - move/turn the base
    - move/turn the head
  - with Android Phone
    - move/turn the base
    - see a camera image
- works for Nao and Pepper
- doc: http://wiki.ros.org/nao_teleop
- How to launch on a phone?
  - launch on PC

    $ roslaunch naoqi_driver naoqi_driver.launch ano_ip:=<robot_IP>
  - launch on Android phone

    $ roslaunch nao_teleop teleop_joy.launch



SoftBank Robotics

Outline

1. wiki
2. robot description in ROS
3. ROS Bridge
   - connect NAO to ROS
   - connect simulated NAO to ROS
4. Visual sensors
5. High-level capabilities
   - Object recognition
   - MoveIt!
   - Teleop
   - Gazebo

SoftBank Robotics

5.4 Gazebo

What is Gazebo simulator?

- allows simulation of robots
- robust physics engine, high-quality graphics, convenient GUI
- suits for indoor and outdoor environments
- useful for
  - rapidly test algorithms,
  - design robots,
  - perform regression testing using realistic scenarios
- Doc: http://gazebosim.org

5.4 Gazebo

How to use Nao with Gazebo?

- doc: http://wiki.ros.org/nao_gazebo_plugin
- compile from source

  $ git clone https://github.com/ros-naoqi/nao_virtual
- start Gazebo with NAO

  $ roslaunch nao_gazebo_plugin nao_gazebo_plugin_H25.launch
- start MoveIt to control NAO

  $ roslaunch nao_moveit_config moveit_planner.launch

GAZEBO

SoftBank Robotics

Conclusion

In these classes we have covered

- general ROS understanding, useful tools and libraries
- connecting your robot to ROS
  - nao description
  - ROS bridge: naoqi_driver, nao_bringup
  - listening recorded data
  - connecting standalone Naoqi
- high-level capabilities: ORK, MoveIt!, Octomap, teleop, Gazebo

Thank you!

Developing own package

- Download SDK: C++ or Python
  - create an account and login in https://community.ald.softbankrobotics.com
  - go to resources, download NaoqiSDK, extract, and source it
    - export PYTHONPATH=$PYTHONPATH:~/prog/pynaoqi-python2.7-2.5.5.5-linux64/lib/python2.7/site-packages:~/prog/pynaoqi-python2.7-2.5.5.5-linux64/lib/python2.7/dist-package:~/catkin_ws/devel/lib/python2.7/dist-packages
- Compiling from source:
  - mkdir -p ~/catkin_ws/src && cd ~/catkin_ws/src
  - git clone https://github.com/ros-naoqi/nao_moveit_config
  - rosdep install -i -y --from-paths ./nao_moveit_config
  - source /opt/ros/kinetic/setup.sh
  - cd ../ && catkin_make