

Solution home / Intera SDK / Tutorials

Gazebo Tutorial

Print

Modified on: Thu, 10 Feb, 2022 at 8:37 AM



This tutorial describes how to use Sawyer with Gazebo the standard Physics Simulator for ROS.

Contents

- Contents
- Installation/Prerequisites
- Sawyer Simulator Installation
- Simulation
- Smoke Test
- Run SDK Examples

Installation/Prerequisites

- Make sure you have followed the **Workstation Setup** tutorial before beginning this section.
- Ensure the following software packages are installed:

ROS Noetic	<ol style="list-style-type: none"> 1 <code>\$ sudo apt-get install gazebo11 ros-noetic-gazebo-ros ros-noetic-gazebo-ros-noetic-gazebo-ros-pkgs ros-noetic-ros-control ros-noetic-control-toolbox ros-noetic-ros-controllers ros-noetic-xacro python3-wstool ros-noetic-tf-co</code>
ROS Melodic	<ol style="list-style-type: none"> 1 <code>\$ sudo apt-get install gazebo9 ros-melodic-qt-build ros-melodic-gazebo-ros ros-melodic-gazebo-ros-pkgs ros-melodic-ros-control ros-melodic-control-too ros-melodic-ros-controllers ros-melodic-xacro python-wstool ros-melodic-tf-</code>
ROS Kinetic	<ol style="list-style-type: none"> 1 <code>\$ sudo apt-get install gazebo7 ros-kinetic-qt-build ros-kinetic-gazebo-ros ros-kinetic-ros-control ros-kinetic-control-toolbox ros-kinetic-realtime-to python-wstool ros-kinetic-tf-conversions ros-kinetic-kdl-parser ros-kinetic</code>

Sawyer Simulator Installation

From your catkin workspace where the SDK resides, use wstool to install and update:

Install sawyer_simulator

ROS Noetic	<ol style="list-style-type: none"> 1 <code>\$ mkdir -p ~/ros_ws/src</code> 2 <code>\$ cd ~/ros_ws/src</code> 3 <code>\$ git clone https://github.com/RethinkRobotics/sawyer_simulator.git -b noet</code> 4 <code>\$ git clone https://github.com/RethinkRobotics-opensource/sns_ik.git -b mel</code> 5 <code>\$ cd ~/ros_ws/src</code> 6 <code>\$ wstool init .</code> 7 <code>\$ wstool merge sawyer_simulator/sawyer_simulator.rosinstall</code> 8 <code>\$ wstool update</code>
ROS Melodic	<ol style="list-style-type: none"> 1 <code>\$ mkdir -p ~/ros_ws/src</code> 2 <code>\$ cd ~/ros_ws/src</code> 3 <code>\$ git clone https://github.com/RethinkRobotics/sawyer_simulator.git</code> 4 <code>\$ git clone https://github.com/RethinkRobotics-opensource/sns_ik.git -b mel</code> 5 <code>\$ cd ~/ros_ws/src</code> 6 <code>\$ wstool init .</code> 7 <code>\$ wstool merge sawyer_simulator/sawyer_simulator.rosinstall</code> 8 <code>\$ wstool update</code>
ROS Kinetic	<ol style="list-style-type: none"> 1 <code>\$ mkdir -p ~/ros_ws/src</code> 2 <code>\$ cd ~/ros_ws/src</code> 3 <code>\$ git clone https://github.com/RethinkRobotics/sawyer_simulator.git</code> 4 <code>\$ cd ~/ros_ws/src</code> 5 <code>\$ wstool init .</code> 6 <code>\$ wstool merge sawyer_simulator/sawyer_simulator.rosinstall</code> 7 <code>\$ wstool update</code>

IMPORTANT: Make sure all simulator repositories update to their proper branch.

Build Source

ROS Noetic	<ol style="list-style-type: none"> 1 <code>\$ source /opt/ros/noetic/setup.bash</code> 2 <code>\$ cd ~/ros_ws</code> 3 <code>\$ catkin_make</code>
------------	---

Related Articles

- Tutorial Overview
- Workstation Setup
- How to export coordinates and perform an ...
- Motion interface Tutorial
- MoveIt Tutorial
- Interaction Control Tutorial
- Using Constrained Zero-G Mode
- Send Email on Error
- Using Measured Force
- Overview

ROS Melodic	1	\$ source /opt/ros/melodic/setup.bash
	2	\$ cd ~/ros_ws
	3	\$ catkin_make
ROS Kinetic	1	\$ source /opt/ros/kinetic/setup.bash
	2	\$ cd ~/ros_ws
	3	\$ catkin_make

Simulation

The intera.sh shell has a special hook of *sim* for Simulation. Run the Intera shell script with sim specified:

```
1 $ ./intera.sh sim
```

Start simulation with controllers:

```
1 $ roslaunch sawyer_gazebo sawyer_world.launch
```

Smoke Test

Check if the simulator was installed and launched successfully by typing the following commands:

```
1 $ rosnodetool list
```

This should list the nodes.

```
1 $ rostopic list
```

This should list the topics.

```
1 $ rostopic echo /robot/state
```

By default, the following messages should be displayed at 100 HZ.

```
1 enabled: False
2 stopped: False
3 error: False
4 estop_button: 0
5 estop_source: 0
6 ---
```

Run SDK Examples

Start Joint Torque Springs example:

```
1 $ ./intera.sh sim
2 $ roslaunch sawyer_gazebo sawyer_world.launch
3 *in a new terminal*
4 $ ./intera.sh sim
5 $ rosrun intera_examples joint_torque_springs.py
```

Or, Start Simulated Pick and Place example:

```
1 $ ./intera.sh sim
2 $ roslaunch sawyer_sim_examples sawyer_pick_and_place_demo.launch
```

For additional information on the interfaces that are implemented with this release, visit the [Simulator API](#) page.

< | >

Did you find it helpful? Yes No