groovy	hydro	indigo	jade	kinetic	Documentation Status

industrial_core (/industrial_core?distro=kinetic)*: industrial_deprecated (/industrial_deprecated?distro=kinetic)* | *industrial_msgs (/industrial_msgs?distro=kinetic)* | *industrial_robot_client (/industrial_robot_client?distro=kinetic)* | *industrial_robot_simulator* | *industrial_trajectory_filters (/industrial_trajectory_filters?distro=kinetic)* | *industrial_utils (/industrial_utils?distro=kinetic)* | *simple_message (/simple_message?distro=kinetic)*

Package Links

- **Code API (http://docs.ros.org/kinetic/api/industrial_robot_simulator/html)**
- FAQ (http://answers.ros.org/questions/scope:all/sort:activity-desc/tags:industrial_robot_simulator/page:1/)
- Changelog (http://docs.ros.org/kinetic/changelogs/industrial_robot_simulator/changelog.html)
- Change List (/industrial_core/ChangeList)
- Reviews (/industrial_robot_simulator/Reviews)

Dependencies (8)
Used by (1)
Jenkins jobs (13)

Package Summary

✔ **Released** ✔ **Continuous integration** ✔ **Documented**

The industrial robot simulator is a stand in for industrial robot driver node(s). It adheres to the driver specification for industrial robot controllers.

- Maintainer status: maintained
- Maintainer: Shaun Edwards <sedwards AT swri DOT org>
- Author: Shaun Edwards
- License: BSD
- Source: git https://github.com/ros-industrial/industrial_core.git (https://github.com/ros-industrial/industrial_core) (branch: kinetic)

Contents

1. Overview

This package simulates an industrial robot controller that adheres to the ROS-Industrial (/Industrial) driver specification. Currently the simulator only supports the minimum requirements (/Industrial/Industrial_Robot_Driver_Spec#Essential_Capabilities). The purpose of this node is to provide a simulated robot controller for development. This simulator publishes standard topics that can be fed into Rviz (/rviz) to create a realistic visualization of an actual robot cell. Note that the simulation is at the ROS API level, the node does not accept Simple Message TCP/UDP connections.

2. Usage

The industrial robot simulator package contains a convenience launch file for bringing up typical low level nodes (sometimes called bringup scripts). These nodes include the simulator and a controlling action server. In typical applications the action server receives trajectories from a higher level planner (not included in this launch file).

Standalone execution is through roslaunch:

```
roslaunch industrial_robot_simulator robot_interface_simulator.launch
```

3. Node API

3.1 industrial_robot_simulator

Simulates an industrial robot controller as defined in ROS-Industrial.

3.1.1 Subscribed Topics

`joint_path_command` (trajectory_msgs/JointTrajectory (http://docs.ros.org/api/trajectory_msgs/html/msg/JointTrajectory.html))
    Commanded joint trajectories.

3.1.2 Published Topics

`joint_states` (sensor_msgs/JointState (http://docs.ros.org/api/sensor_msgs/html/msg/JointState.html))
    Joint State for each non-fixed joint in the robot.

`feedback_states` (control_msgs/FollowJointTrajectoryFeedback (http://docs.ros.org/api/control_msgs/html/msg/FollowJointTrajectoryFeedback.html))
    Feedback information (errors) for each non-fixed joint state.

3.1.3 Parameters

`controller_joint_names` (`str[]`, default: ['joint_1', ..., 'joint_6'])
    Joint controller names (for more info see here (/industrial_robot_client/generic_implementation#Parameters)).

`initial_joint_state` (`double[]`, default: [0, ..., 0])

Initial state of all (revolute) joints. Units: rad.

`motion_update_rate` (`double`, default: 100.0)
: Internal update rate of the motion interpolator. If 0, node will not interpolate between trajectory points. Units: Hz.

`pub_rate` (`double`, default: 10.0)
: Publish rate for state publisher(s). Units: Hz.

4. Contact us/Technical support

For questions related to industrial robot simulator or ROS Industrial in general, please contact the developers using the ROS-Industrial (https://groups.google.com/forum/?fromgroups#!forum/swri-ros-pkg-dev) Google group (direct mail: ROS-Industrial (mailto:swri-ros-pkg-dev@googlegroups.com)).

5. Reporting bugs

Use GitHub to report bugs or submit feature requests (https://github.com/ros-industrial/industrial_core/issues/new). [View active issues (https://github.com/ros-industrial/industrial_core/issues?page=1&state=open)]

Wiki: industrial_robot_simulator (last edited 2016-03-13 14:32:36 by IanMcMahon (/IanMcMahon))