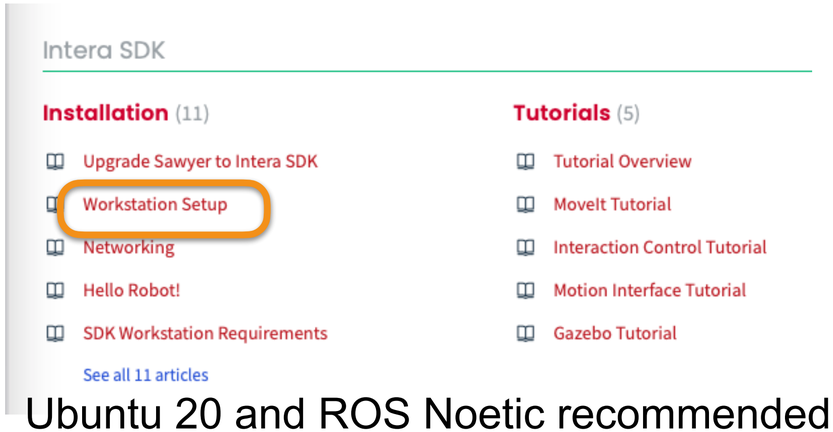
Sawyer with ROS1 / ROS1 with Sawyer

Ostfalia
University of
Applied Sciences

- Content
  - Setting up Workstation for Sawyer (on your computer*)
  - ROS1
    - Turtlesim
    - ‘Hello World’ (with Python and Sawyer)
    - MoveIt!
- Literature/Sources/Inspiration
  - https://support.rethinkrobotics.com/support/solutions
  - https://ubuntu.com
  - https://www.ros.org
- To Do
  - Set up workstation (own computer)



Ubuntu 20 and ROS Noetic recommended

For ROS boot Sawyer into SDK mode:

- Keep hitting CTRL+F on the keyboard during startup
- Select ‘Intera-SDK’
- Reboot

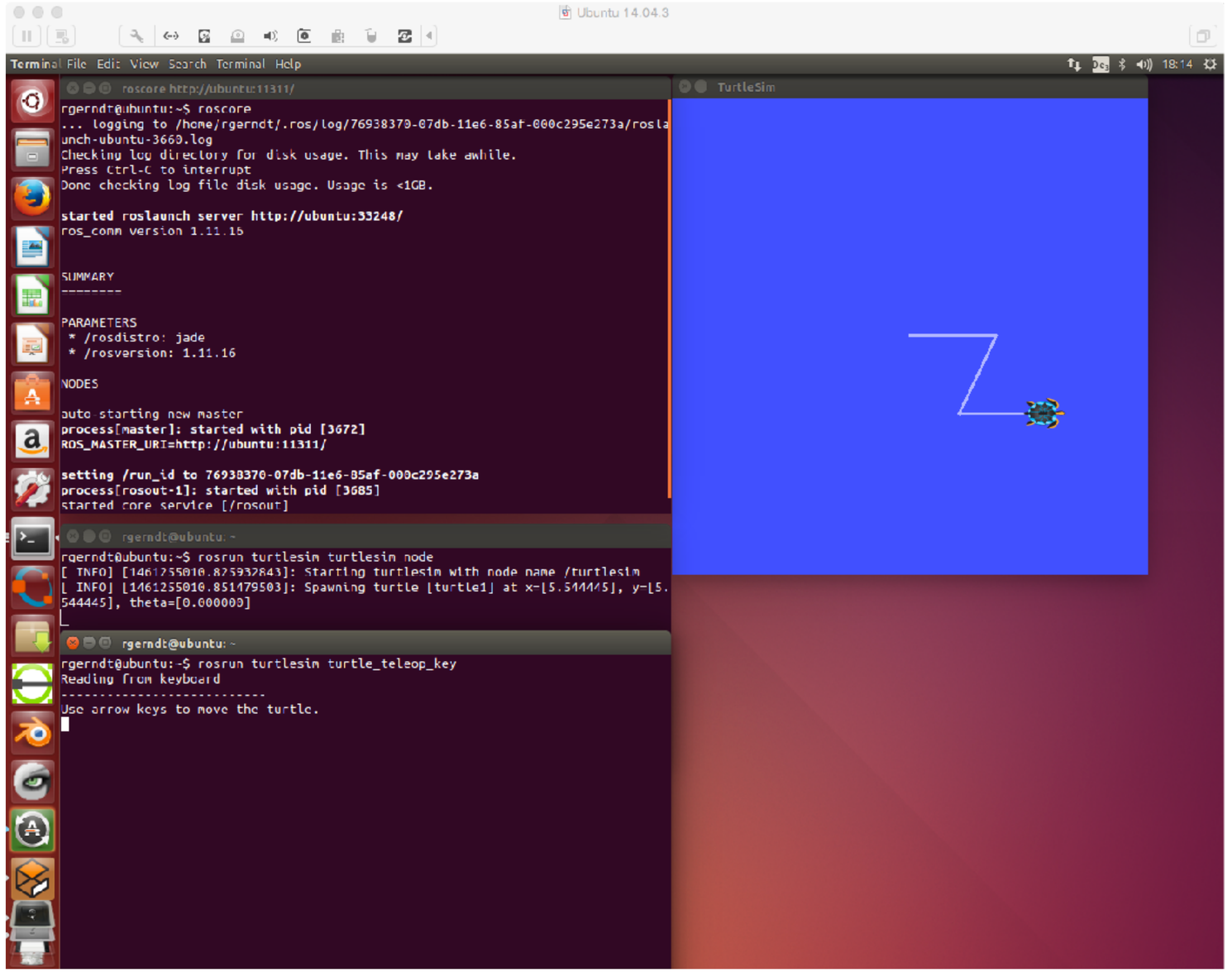
(For Intera5 select ‘Intera’)

*consider installing a VirtualBox (www.virtualbox.org)

ROS Run&Smoke Test - Turtlesim and Teleop

Ostfalia
University of
Applied Sciences

- 3 terminals
  - **roscore**
  - **rosrun turtlesim turtlesim_node**
  - **rosrun turtlesim turtle_teleop_key**
- Control turtle with keys in teleop terminal



Packages

Ostfalia
University of
Applied Sciences

- All ROS software in packages
  - e.g. ‚turtlesim‘
- package is collection of files
  - e.g. ‚turtlesim_node‘,…
- Publisher - Subscriber
  - nodes publish messages on a topic
  - nodes subscribe to a topic
- Instructions
  - rospack list
  - rospack find *package-name* (e.g. turtlesim)
  - rosls *package-name*
  - roscd *package-name*

ROS Core and Nodes

Ostfalia
University of
Applied Sciences

- Central ROS Core
  - handling communication
- Distributed Nodes
  - independently running programs
  - Instructions
    - rosrun *package-name executable-name*
      - e.g. rosrun turtlesim turtlesim_node
    - rosnode list
    - rosnode info *node-name* (see list)
    - rosnode kill *node-name*

Topics and Messages

Ostfalia
University of
Applied Sciences

- Related instructions
  - rqt_graph
  - rostopic list
  - rostopic echo *topic-name*
    - e.g. rostopic echo /turtle1/cmd_vel
  - rostopic info *topic-name*
  - rostopic {hz, bw} *topic-name* ({speed, bandwidth})
  - rostopic info *topic-name* (outputs message type)
  - rosmsg show *message-type-name*

Multiple Instances

Ostfalia
University of
Applied Sciences

- Four separate terminals, roscore running
  - rosrun turtlesim turtlesim_node __name:=A
  - rosrun turtlesim turtlesim_node __name:=B
  - rosrun turtlesim turtle_teleop_key __name:=C
  - rosrun turtlesim turtle_teleop_key __name:=D
    - ‚name‘ parameter overwrites the default name

Debugging

Ostfalia
University of
Applied Sciences

- Sanity checks
  - roswtf
- Publishing messages from the command line for debugging
  - rostopic pub -r *rate-in-hz topic-name message-type message-content*
    - e.g. rostopic pub -r 1 /turtle1/cmd_vel geometry_msgs/Twist ‚[0, 0, 0]‘ ‚[0, 0, 1]‘

catkin - ROS Build System

Ostfalia
University of
Applied Sciences

- Setting up a workspace
  - mkdir -p ~/catkin_ws/src
  - cd ~/catkin_ws/src
  - catkin_init_workspace
- Build package
  - cd ~/catkin_ws
  - catkin_make

Creating an ROS package

Ostfalia
University of
Applied Sciences

- change to source directory
  - cd ~/catkin_ws/src
- create hello_world package
  - catkin_create_pkg hello_world std_msgs rospy
- change to hello_world package/folder
- create folder ‚scripts‘ and change working directory
  - mkdir scripts
  - cd scripts

Hello_world_publisher.py

Ostfalia
University of
Applied Sciences

```
#!/usr/bin/env python
# license removed for brevity
import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('hello_pub', String, queue_size=10)
    rospy.init_node('hello_world_publisher', anonymous=True)
    r = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        str = "hello world %s"%rospy.get_time()
        rospy.loginfo(str)
        pub.publish(str)
        r.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException: pass
```

import rospy

publisher object ‚hello_pub‘, type string, queue 10

name assignment

publishing rate of 10 Hz

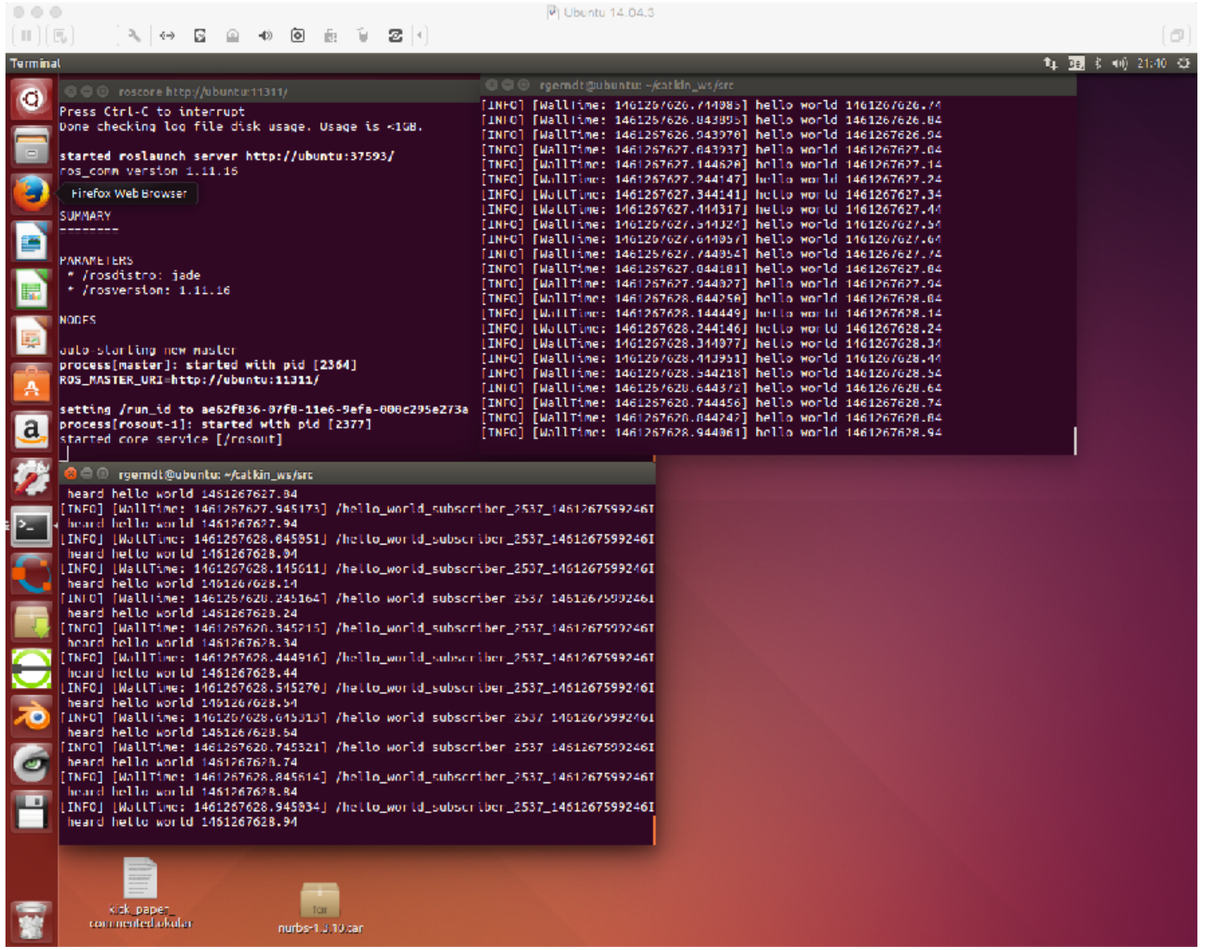
Hello_world_subscriber.py

Ostfalia
University of
Applied Sciences

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id()+"I heard %s",data.data)

def listener():

    # in ROS, nodes are unique named. If two nodes with the same
    # node are launched, the previous one is kicked off. The
    # anonymous=True flag means that rospy will choose a unique
    # name for our 'listener' node so that multiple listeners can
    # run simultaenously.
    rospy.init_node('hello_world_subscriber', anonymous=True)

    rospy.Subscriber("hello_pub", String, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == '__main__':
    listener()
```

Run

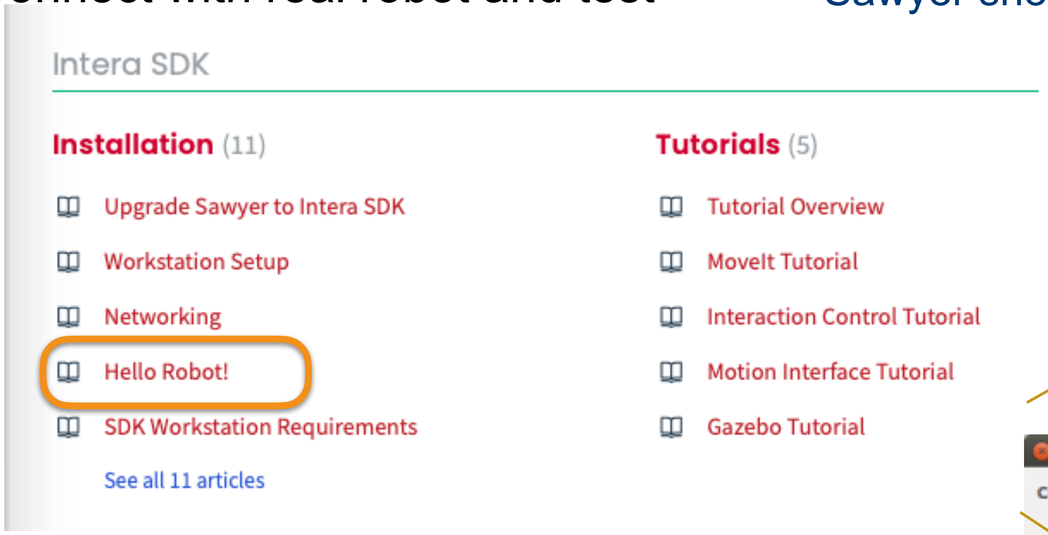
- make files executable
  - chmod +x hello_world_publisher.py
  - chmod +x hello_world_subscriber.py
- build
  - cd ~/catkin_ws
  - catkin_make
- possibly add path
  - echo „source ~/catkin_ws/devel/setup.bash“ >> ~/.bashrc
  - source ~/.bashrc (depending on shell, e.g. .profile on mac)
- run (in individual terminals)
  - roscore
  - rosrun hello_world hello_world_publisher.py
  - rosrun hello_world hello_world_subscriber.py
- try
  - rqt_graph

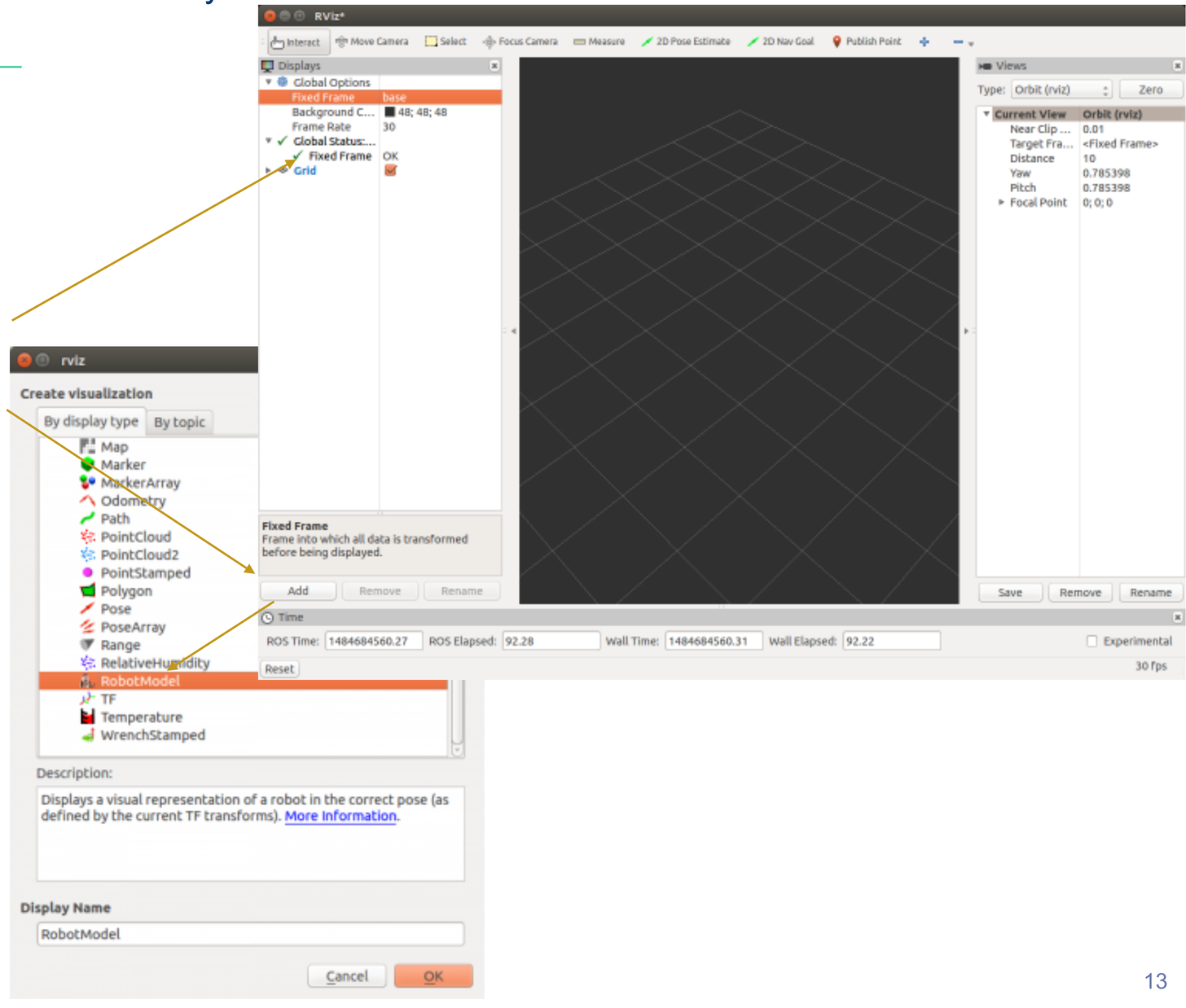


Continue with Sawyer

- Connect with real robot and test

Sawyer should be ‘sawyer.local’





- Check out RVIZ
  - Start RVIZ
    - $ rosrun rviz rviz
  - Configure RVIZ
    - Set Fixed Frame as ‚/base‘
    - Add ‚Robot Model‘ to RVIZ

Continue with Sawyer

Ostfalia
University of
Applied Sciences

- Check out ‘MoveIt! (https://moveit.ros.org)

Intera SDK

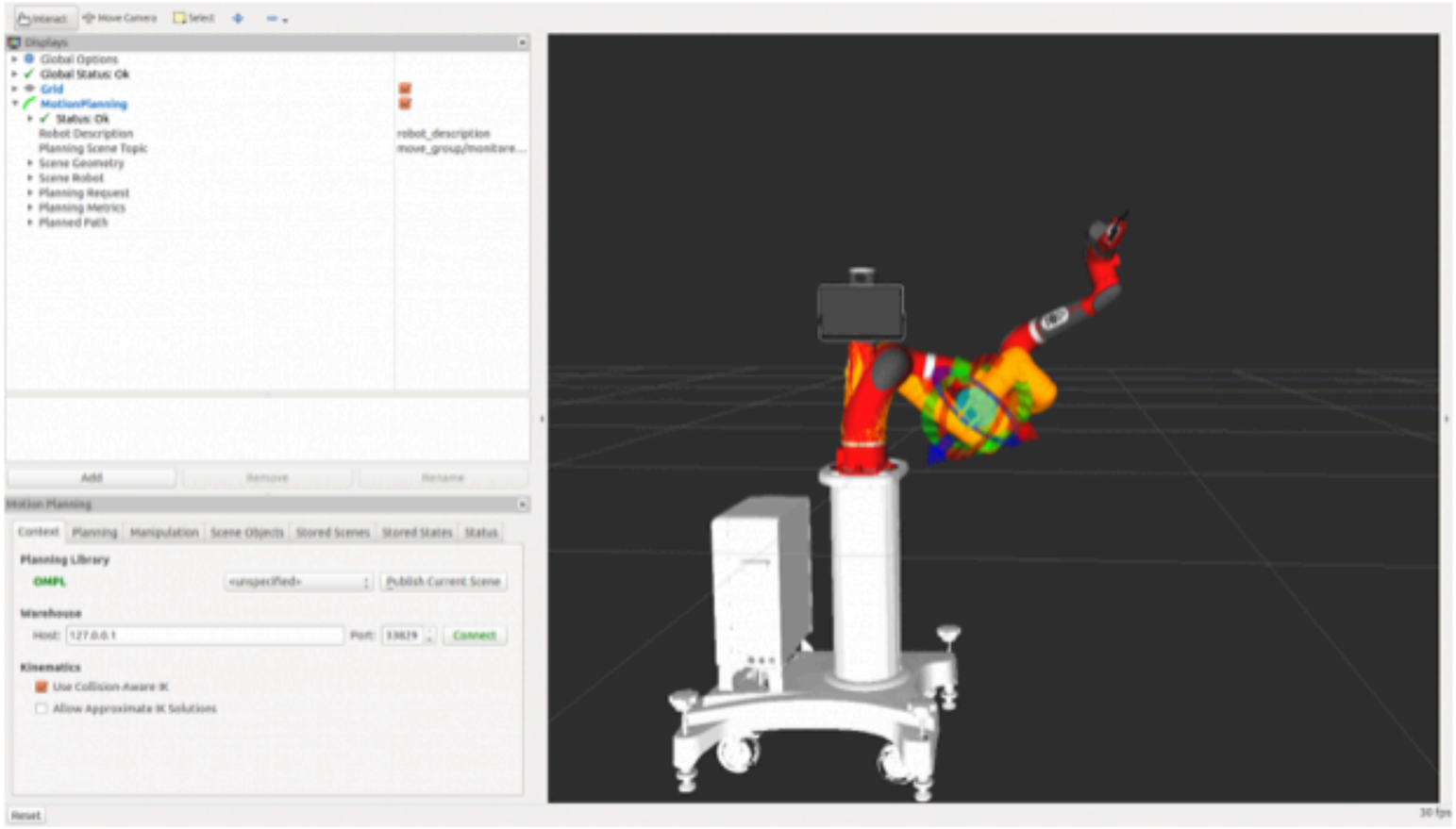
Installation (11)

- Upgrade Sawyer to Intera SDK
- Workstation Setup
- Networking
- Hello Robot!
- SDK Workstation Requirements

See all 11 articles

Tutorials (5)

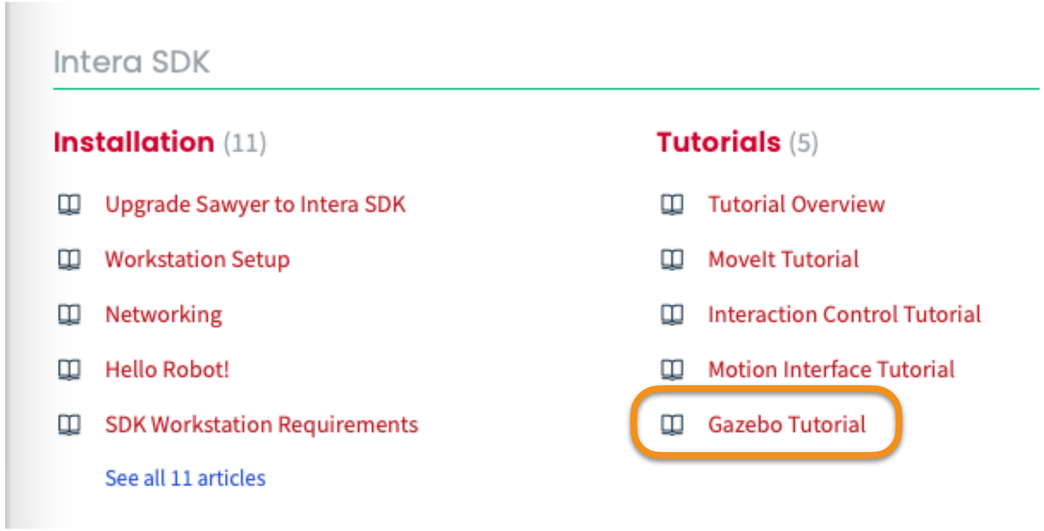
- Tutorial Overview
- MoveIt Tutorial
- Interaction Control Tutorial
- Motion Interface T
- Gazebo Tutorial



Gazebo Simulation for Sawyer (no sawyer required)

Ostfalia
University of
Applied Sciences

- Visit https://support.rethinkrobotics.com/support/solutions/articles/80000980381-gazebo-tutorial
-



Sawyer Task

Ostfalia
University of
Applied Sciences

- Implement ‘some (kinematical) function’
- Run on
  - Gazebo
  - Sawyer
- Compare
- Possibly try with a simple ‘teleop’