Managing Coordinate Frames with ROS

João Bimbo
King's College London

Part II
Hands-on

ROS.org
tf command line operations

• Create a new frame with the name “cam_base”, parent to “/openni_camera” with the following parameters:
  • Rotation: roll = 0; pitch = 0.51; yaw = 3.14159265
  • Translation: [ 0 0 0.1 ]
  • Frequency: 100 Hz
  • Usage: rosrnn tf static_transform_publisher x y z yaw pitch roll frame_id child_frame_id period(milliseconds)
Publishing a static transform from the command line

rosrun tf static_transform_publisher 0 0 0.1 3.14159265 0.51 0 /cam_base /openni_camera 100

• Now, in a new window, run:

rosrun tf tf_echo /cam_base /openni_camera
Publishing a static transform from the command line

```bash
rosrun tf static_transform_publisher 0 0 0.1 3.14159265 0.51 0 /cam_base /openni_camera 100
```

• Now, in a new window, run:

```bash
rosrun tf tf_echo /cam_base /openni_camera
```
- Open `/tf_workshop/launch/tf_frames.launch`

- Uncomment the first two static transform publishers

- Run `rviz`: `rosrun rviz rviz`
Create a Transform Broadcaster (Python)

- Open tf_workshop/nodes/track_kin.py
- Line 17: Broadcast the end_effector pose, according to the values in x,y,z and the rotation quaternion, acquired from the topic “/arm_pose”, which relate to the frame “robot_base”

```python
br.sendTransform(x,y,z,rot,Time,"frame1", "frame2")
```
Create a Transform Broadcaster (Python)

```python
br.sendTransform((x, y, z, quat, rospy.Time.now()),
"end_effector", "robot_base")

• Run: rosrun tf_workshop track_kin.py
```
Create a Transform Broadcaster (Python)

```
br.sendTransform( (x,y,z, quat, rospy.Time.now(),
"end_effector", "robot_base")

• Run: rosr
```
Create a Transform Listener (C++)

- Open tf_workshop/src/object_tracker.cpp
- Notice there's a new frame called "object_frame" being broadcast
- Go to line 66: uncomment the three lines. Compile using:
  
  rosmake tf_workshop

  rosrn tf_workshop obj_tracker

- Is everything working well?
Create a Transform Listener (C++)

```cpp
listener.waitForTransform("/object_frame", "/end_effector", 
msg.header.stamp, ros::Duration(0.2));

listener.lookupTransform("/object_frame", "/end_effector", 
msg.header.stamp, transform);

listener.transformPointCloud("/end_effector", objectpc, objectpc);
```