Managing Coordinate Frames with ROS

João Bimbo
King's College London

Part I

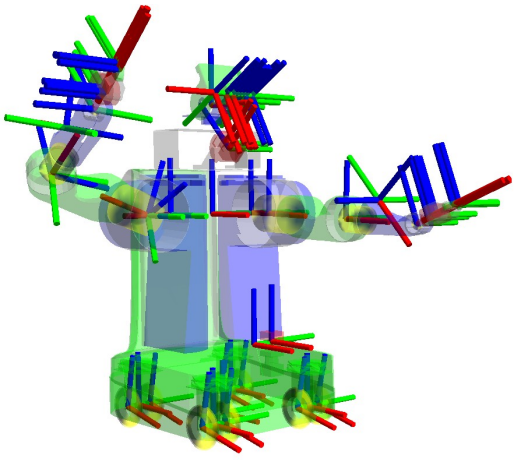
Tutorial

KING'S College LONDON

ROS.org

HanDLe

Introduction – The *tf* package



- The *tf* package allows the tracking over time of coordinate systems tree(s)
- Allows the easily creation of new frames (static or dynamic)
- Eases the process of transforming points, vectors, etc.
- Distributed system – no centralised storage
- Caches the past information on the transforms

KING'S College LONDON

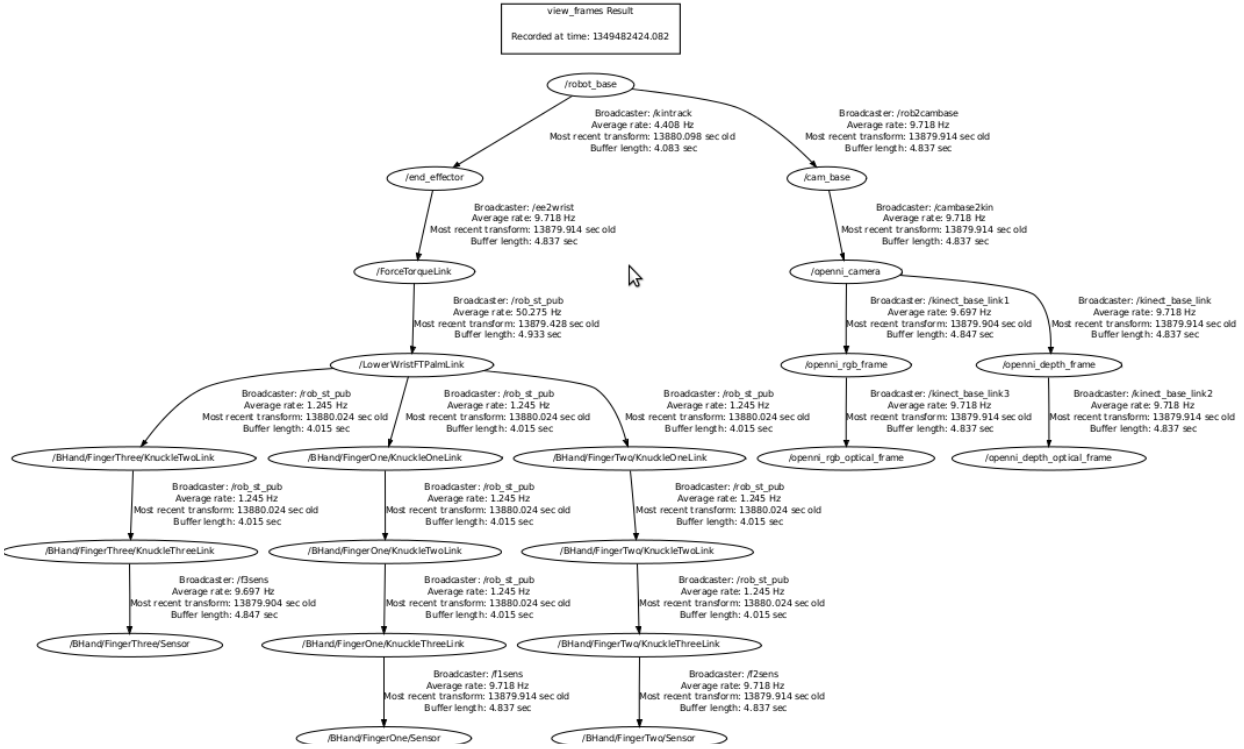
ROS.org

HanDLe

The *tf* coordinate frame tree

- A tree of the current coordinate frame can be generated using the command: `rosrun tf view_frames`

- The output is a pdf file containing the different frames, their publishing frequency, the relevant node. etc.



KING'S College LONDON

ROS.org

HANDLE

Other *tf* commands

- `rosrun tf tf_echo`
- `rosrun tf tf_monitor`
- `rosrun tf static_transform_publisher:`
  - `Usage: rosrun tf static_transform_publisher x y z yaw pitch roll frame_id child_frame_id  period(milliseconds)`

    `OR`
  - `Usage: rosrun tf static_transform_publisher x y z qx qy qz qw frame_id child_frame_id  period(milliseconds)`

tf Python/C++ library

- Broadcasters create new frames and defines the transformations:
  - Python:

```
br = tf.TransformBroadcaster()
br.sendTransform(x,y,z,rot,Time,"frame1", "frame2")
```

  - C++:

```
  static tf::TransformBroadcaster br;
  tf::Transform transform;
  transform.setOrigin(tf::Vector3);
  transform.setRotation( tf::Quaternion(r, p, y) );
  br.sendTransform(tf::StampedTransform(transform,Time, "frame1",
"frame2"));
```

KING'S College LONDON

ROS.org

HanDLe

tf Python/C++ library

- Listeners read the current tree of transforms and compute the transform between two frames in the same tree
  - Python:

```
listener = tf.TransformListener()
(trans,rot)=listener.lookupTransform('/frame1','/frame2',
rospy.Time(0))
```

  - C++:

```
tf::TransformListener listener;
tf::StampedTransform transform;

listener.lookupTransform("/turtle2", "/turtle1",ros::Time(0),
transform);
```

KING'S College LONDON

ROS.org

HanDLe

tf and time

• Because not every transform is available at all times, it is advisable to use the following code block (wrap around a try/catch):

```
Try {
ros::Time now = ros::Time::now();
listener.waitForTransform("/frame2","/frame1",now,ros::Duration(3.0));
listener.lookupTransform("/turtle2", "/turtle1",now, transform); }
```

• ROS can also deal with transformations in the past:

```
listener.lookupTransform("/frame2",now,"/frame1", past,"/world", transform);
```

KING'S College LONDON

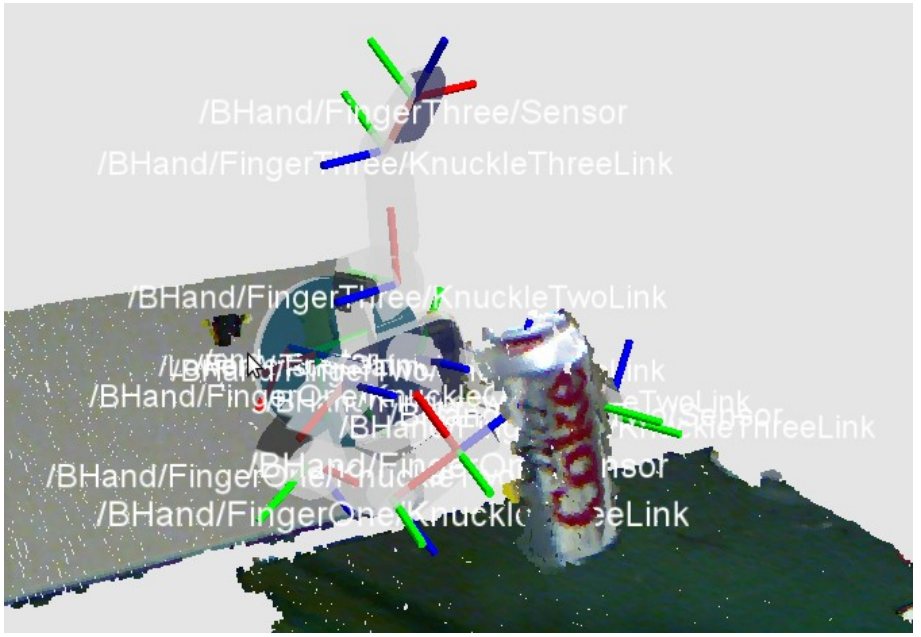
ROS.org

HANDLE

URDF robot model

- A Unified Robot Description Format model can be used to calculate the kinematics, add new coordinate frames and moving them according to encoder values from the robot.

- URDF model can also include other physical properties such as inertia, collisions, joint dynamics, etc.



KING'S College LONDON

ROS.org

HANDLE