`tf`: The Transform Library

Autonomous Robotics Labs

Tomáš Petříček

March 16, 2020

Overview

- Maintains the relationship between **named coordinate frames organized in a directed tree** structure
- Lets the user transform data between any two coordinate frames at any desired point in time
  - Interpolate by default, extrapolate on request
- Distributed
  - Broadcast everything and assemble at consumers
  - Limited buffer size (usually 10 s)
- Currently second generation: `tf2`
- See http://wiki.ros.org/tf2

ROS Topics

- `tf2_msgs/TFMessage` to interchange transforms
- `/tf` for dynamic transforms
- `/tf_static` for static transforms
  - Publisher should latch at the last message

Python API

- `tf = tf2_ros.Buffer()`

```
tf.lookup_transform(target_frame, source_frame,
                    time, timout)
tf.lookup_transform_full(target_frame, target_time,
                         source_frame, source_time,
                         fixed_frame, timeout)
tf.transform(object_stamped, target_frame, timeout)
```

- `tf_sub = tf2_ros.TransformListener(tf)`

- `tf_pub = tf2_ros.TransformBroadcaster()`
  `tf_pub = tf2_ros.StaticTransformBroadcaster()`

```
tf_pub.sendTransform(...)
```

Tools

Graphical & Textual

```
$ rosrun rqt_tf_tree rqt_tf_tree
$ rosrun rviz rviz

$ rosrun tf tf_echo <from> <to>
$ rosrun tf tf_monitor
$ rosrun tf view_frames
$ roswtf
```

Python Libraries

- ros_numpy

```python
from geometry_msgs.msg import Transform
import numpy as np
from ros_numpy import msgify, numpify
from sensor_msgs.msg import PointCloud2

# Get 4-by-4 matrix from transform message.
T = numpify(Transform())
R = T[:3, :3]
t = T[:3, 3:]

# Get flat structured array from point cloud message.
x = numpify(PointCloud2()).ravel()
x = np.stack([x[f] for f in ('x', 'y', 'z')])
x = np.matmul(R, x) + t

# Convert transform matrix to message.
msg = msgify(Transform, T)
```