BLG456E
Robotics
ROS TF Library

Lecture Contents:

- Reference frames in ROS.
- /tf topic.
- tf library.
- Visualising/debugging TF.

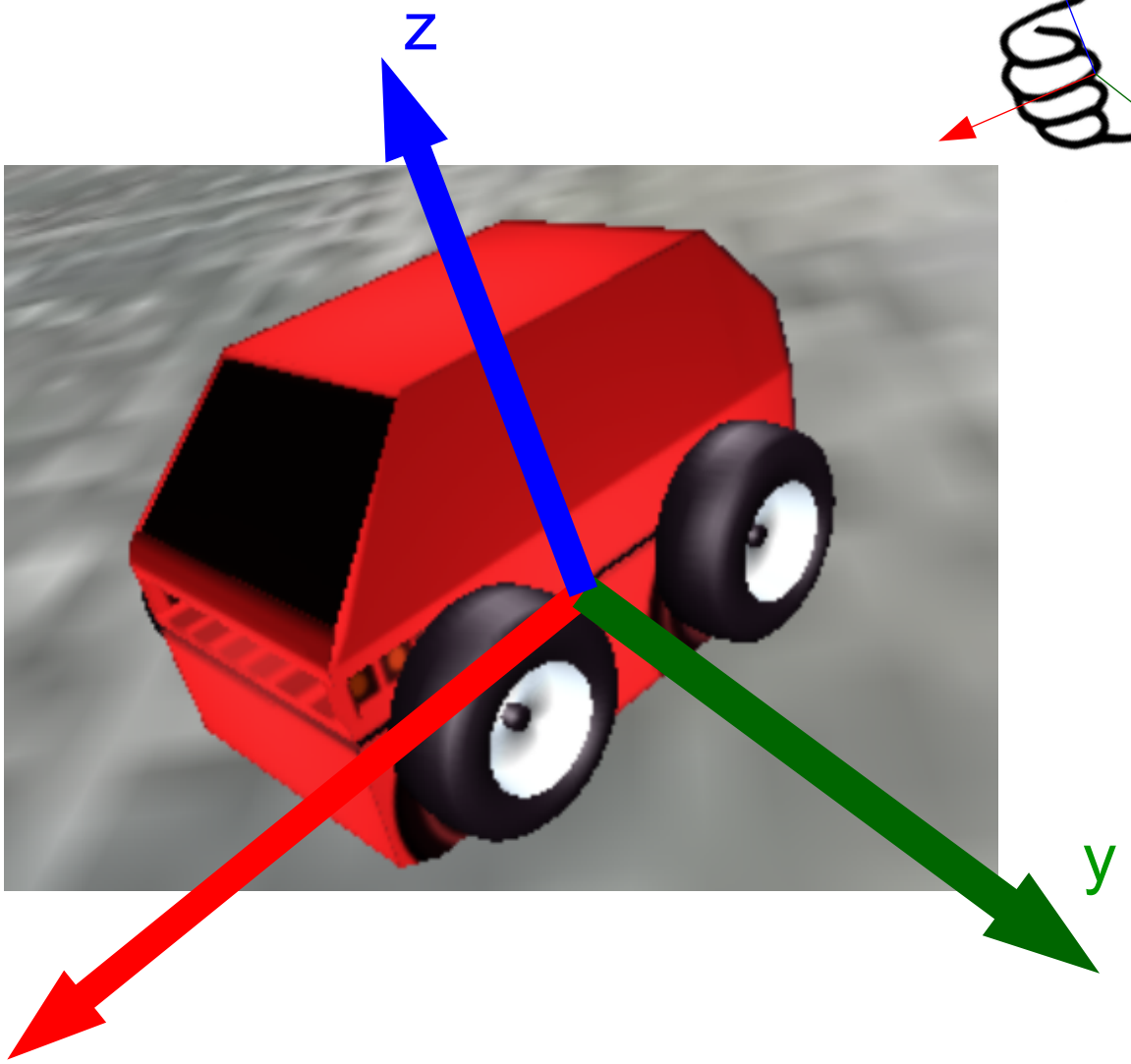
Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Course web:	Ninova
Slides:	Damien Duff

ROS robot coordinate frames convention

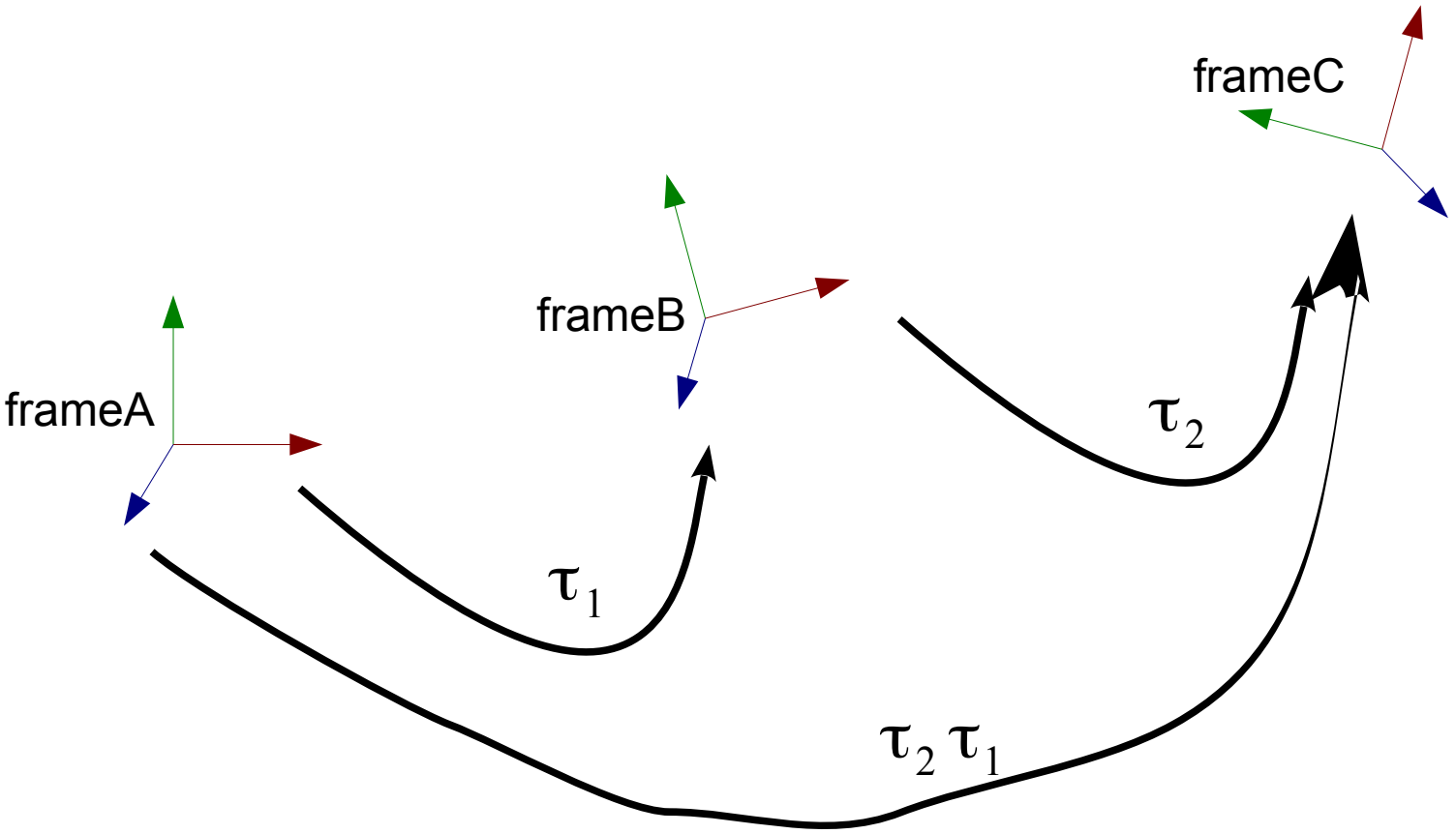
Mobile robots:

- **x** is forward.
- **y** is left.
- **z** is up.

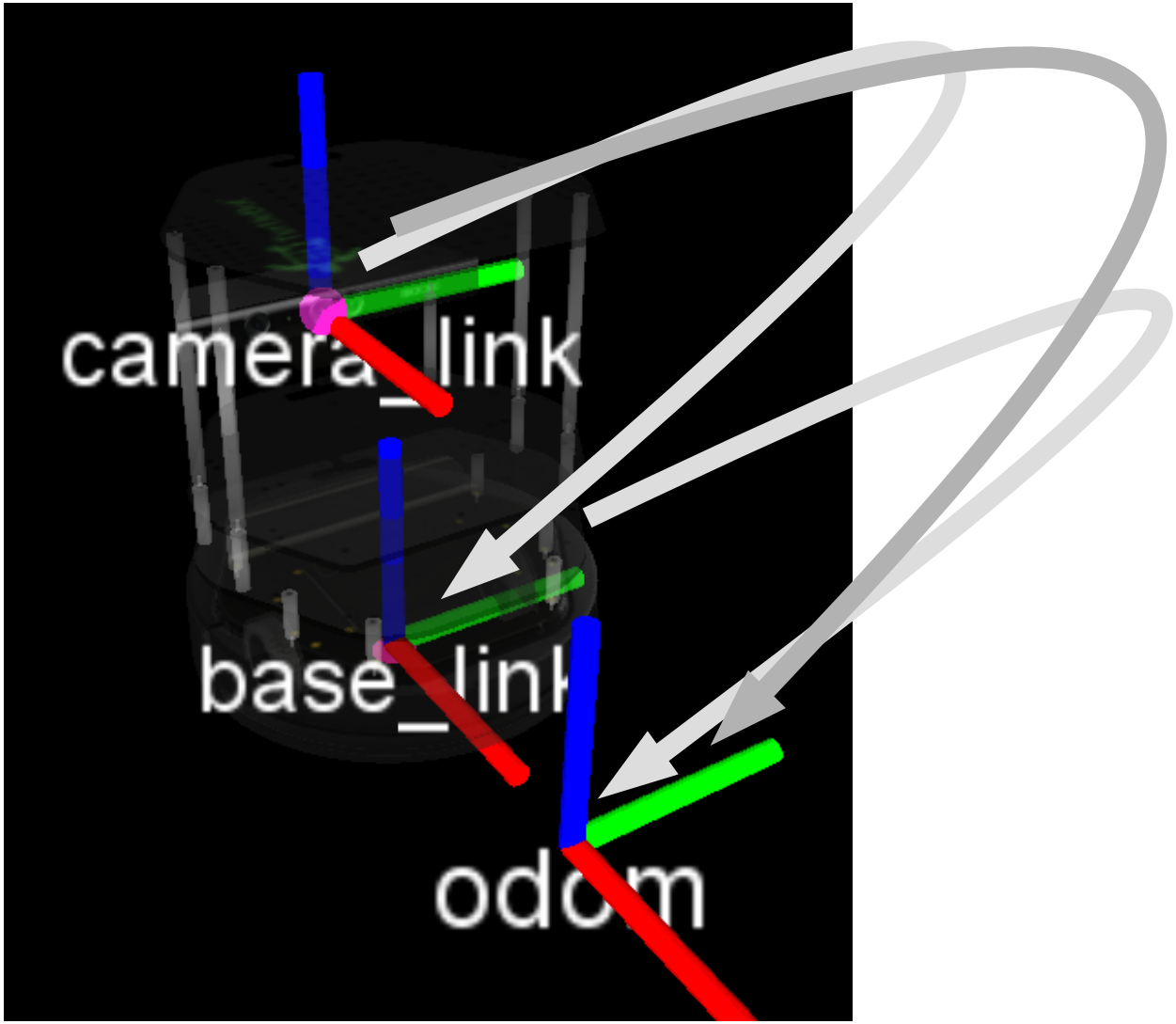
("right-handed coordinate system")



Recall: transformations between multiple frames



Reference frames / transforms in ROS (TF)



BLG456E
Robotics
ROS TF Library

Lecture Contents:

- Reference frames in ROS.
- /tf topic.
- tf library.
- Visualising/debugging TF.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Course web:	Ninova
Slides:	Damien Duff

ROS /tf

`/tf`

- A topic.
- Any node can publish transformation messages to it using the `tf` library.
- Any node can subscribe to it using the `tf` library.

See http://wiki.ros.org/tf/Tutorials

Example TF message

```
header:
        seq: 0
        stamp:
                secs: 11
                nsecs: 440000000
        frame_id: odom
child_frame_id: base_footprint
transform:
        translation:
                x: -0.135
                y: 0.0
                z: 0.009
        rotation:
                x: -0.707106781185
                y: 0.0
                z: 0.0
                w: 0.707106781188
```

When the transform was computed

A transform from **base_link** to **odom**.

odom is this displacement from **base_footprint**

odom is rotated like this with respect to **base_footprint** (represented as a quaternion)

BLG456E
Robotics
ROS TF Library

Lecture Contents:

- Reference frames in ROS.
- /tf topic.
- tf library.
- Visualising/debugging TF.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Course web:	Ninova
Slides:	Damien Duff

C++

ROS tf

`tf` is a library that

- Listens to the `/tf` topic for a node.
- Automatically composes transformations:

```
#include <tf/transform_listener.h>
…
ros::init(argc, argv, "transform_listening_node");
tf::TransformListener listener;
tf::StampedTransform robot_pose;
...
listener.lookupTransform("/odom", "/base_footprint",
ros::Time(0), robot_pose);
```

See http://wiki.ros.org/tf/Tutorials

Python

ROS tf

`tf` is a library that

- Listens to the `/tf` topic for a node.
- Automatically composes transformations:

```
import tf
…
rospy.init_node("transform_listening_node")
listener = tf.TransformListener()
...
robot_pose = listener.lookupTransform("/odom",
"/base_footprint",  rospy.Time(0))
```

See http://wiki.ros.org/tf/Tutorials

Example: StampedTransform data structure

C++

```
listener.lookupTransform("/odom", "/base_footprint",
ros::Time(0), robot_pose);


cout<<"Robot X:"<<robot_pose.getOrigin().x()<<endl;

cout<<"Robot Y:"<<robot_pose.getOrigin().y()<<endl;


tf::Quaternion rotation = robot_pose.getRotation();

tf::Vector3 axis = rotation.getAxis();

cout<<"Robot theta:"<<rotation.getAngle()*axis[2]<<endl;
```

Example: StampedTransform data structure

Python

```
(translation,orientation) =
listener.lookupTransform("/odom", "/base_footprint",
rospy.Time(0));


cout<<”Robot X:”<<translation[0]<<endl;

cout<<”Robot Y:”<<translation[1]<<endl;


r_xorient, r_yorient, r_zorient =
transformations.euler_from_matrix(transformations.quater
nion_matrix(orientation))
```

Example: StampedTransform for transforming points

C++

```
listener.lookupTransform("/odom", "/base_footprint",
ros::Time(0), robot_pose);


tf::Vector3 point_1_R(3,5,0);

tf::Vector3 point_1_I = robot_pose * point_1_R;


tf::Vector3 point_2_I(1,-3,0);

tf::Vector3 point_2_R = robot_pose.inverse() * point_2_I;
```

Example: transformPoint

Python

```
from geometry_msgs.msg import PointStamped


point_1_R = PointStamped()
point_1_R.point.x=3
point_1_R.point.y=5
point_1_R.header.frame_id="/base_footprint"


point_1_I  = listener.transformPoint("/odom", point_1_R)


point_2_I = PointStamped()
point_2_I.point.x=1
point_2_I.point.y=-3
point_2_I.header.frame_id="/odom"


point_2_R  = listener.transformPoint("/base_footprint", point_2_I)
```

BLG456E
Robotics
ROS TF Library

Lecture Contents:

- Reference frames in ROS.
- /tf topic.
- tf library.
- Visualising/debugging TF.

Lecturer:	Damien Jade Duff
Email:	djduff@itu.edu.tr
Office:	EEBF 2316
Schedule:	http://djduff.net/my-schedule
Course web:	Ninova
Slides:	Damien Duff

Tools for debugging tf

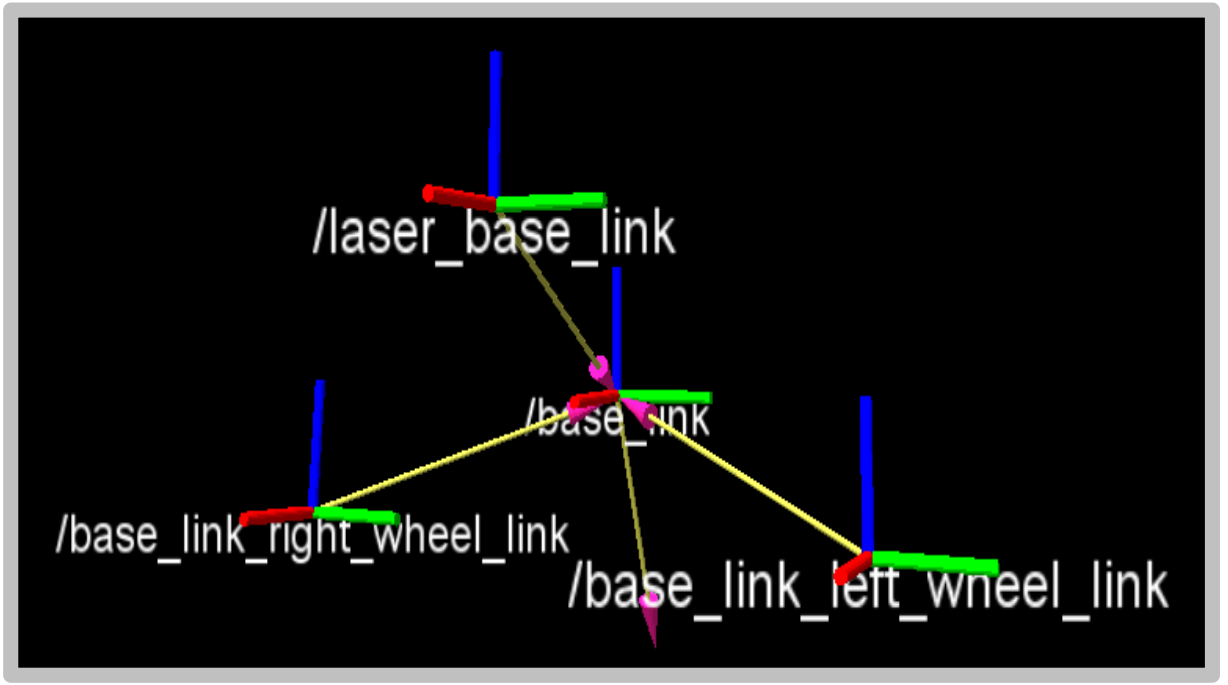
To visualise use:

```
rosrun tf view_frames
```

(produces frames.pdf)

(note: transformation graph (tf) ≠ topic graph (rxgraph))

Or add tf to rviz.



Tools for debugging tf

For fun, try:

```
rostopic echo /tf
```

To see transform between “odom” and “base_link” frames:

```
rosrun tf tf_echo /odom /base_link
```

Typical output:

```
At time 38.850
- Translation: [0.463, -0.002, 0.010]
- Rotation: in Quaternion [0.000, 0.000, 0.642, 0.766]
            in RPY [0.000, -0.000, 1.395]
```

Experiment with transforms

```
cd /opt/ros/kinetic/lib/python2.7/dist-packages/tf/
ipython
import transformations
import math
help(transformations.rotation_matrix)
M=transformations.rotation_matrix(math.pi/2,(0,1,0))
Q=transformations.quaternion_from_matrix(M)
E=transformations.euler_from_matrix(M)
Q2=transformations.quaternion_from_euler(*E)
M2=transformations.quaternion_matrix(Q2)
R2=transformations.rotation_from_matrix(M2)
```