

# BLG456E

## Robotics

### ROS TF Library

#### Lecture Contents:

- /tf topic
- tf library
- Visualising TF
- Debugging TF

<b>Lecturer:</b>	Damien Jade Duff
<b>Email:</b>	<a href="mailto:djduff@itu.edu.tr">djduff@itu.edu.tr</a>
<b>Office:</b>	EEBF 2316
<b>Schedule:</b>	<a href="http://djduff.net/my-schedule">http://djduff.net/my-schedule</a>
<b>Course web:</b>	Ninova
<b>Slides:</b>	Damien Duff

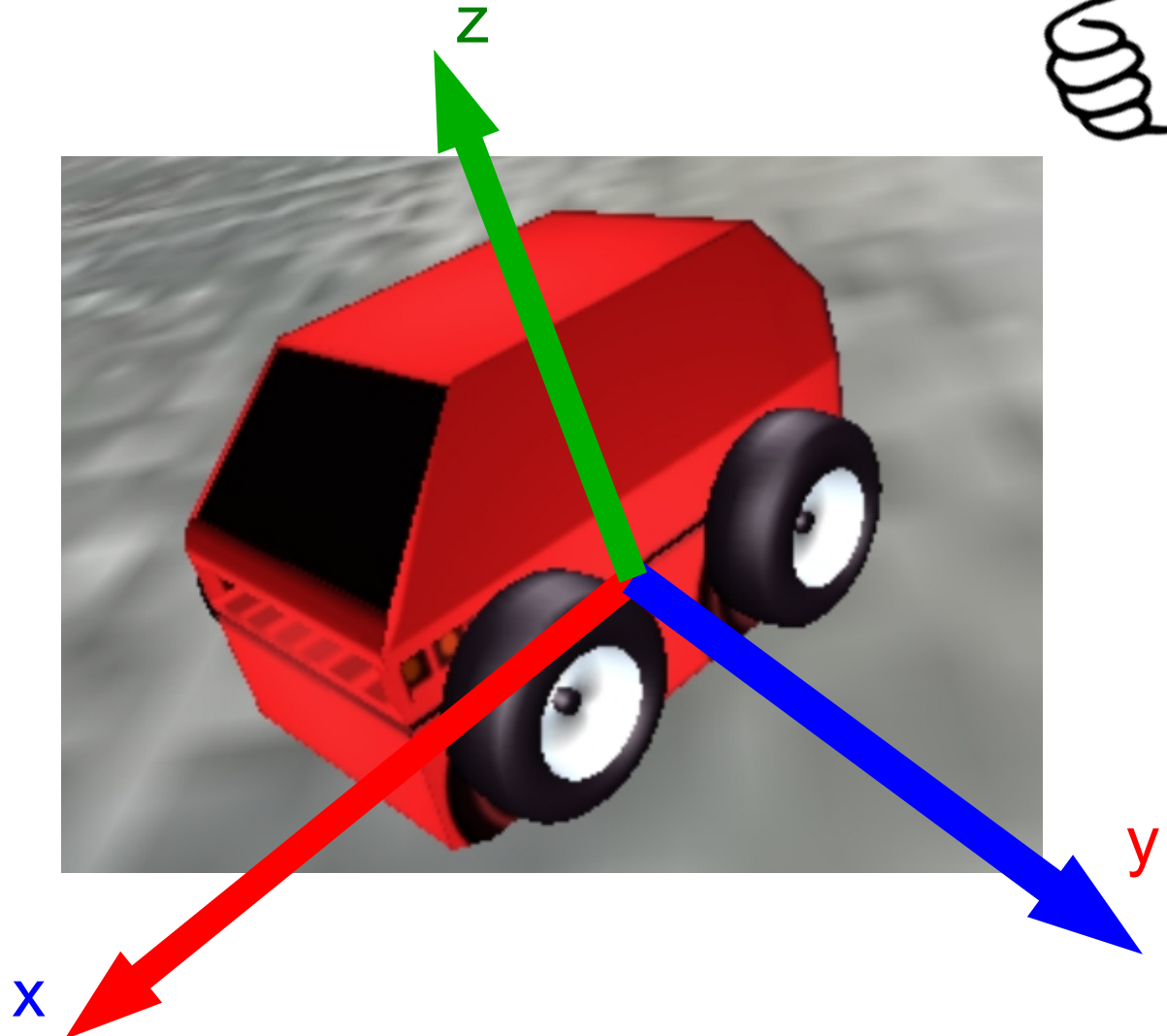
# ROS robot coordinate frames convention



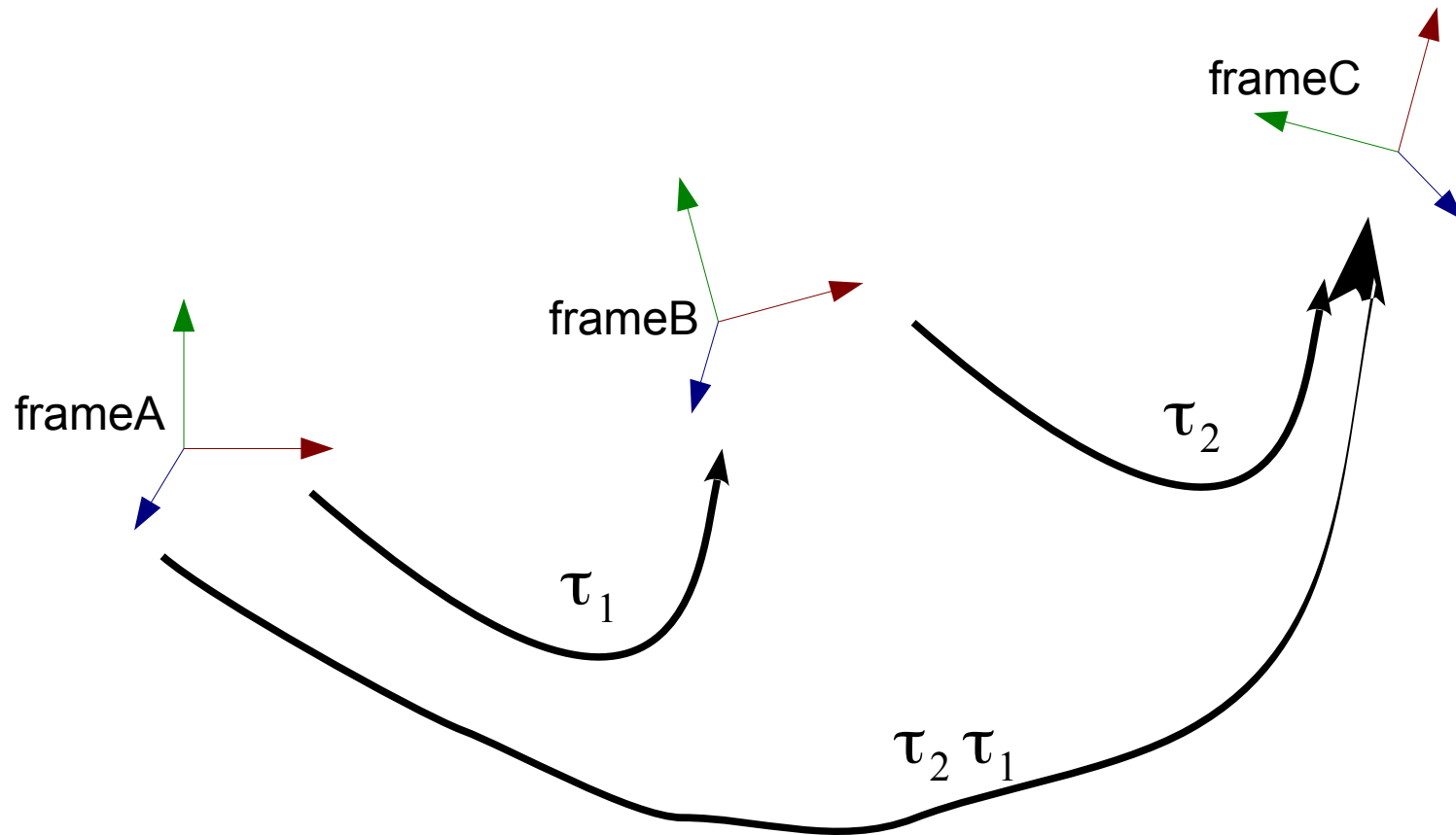
Mobile robots:

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

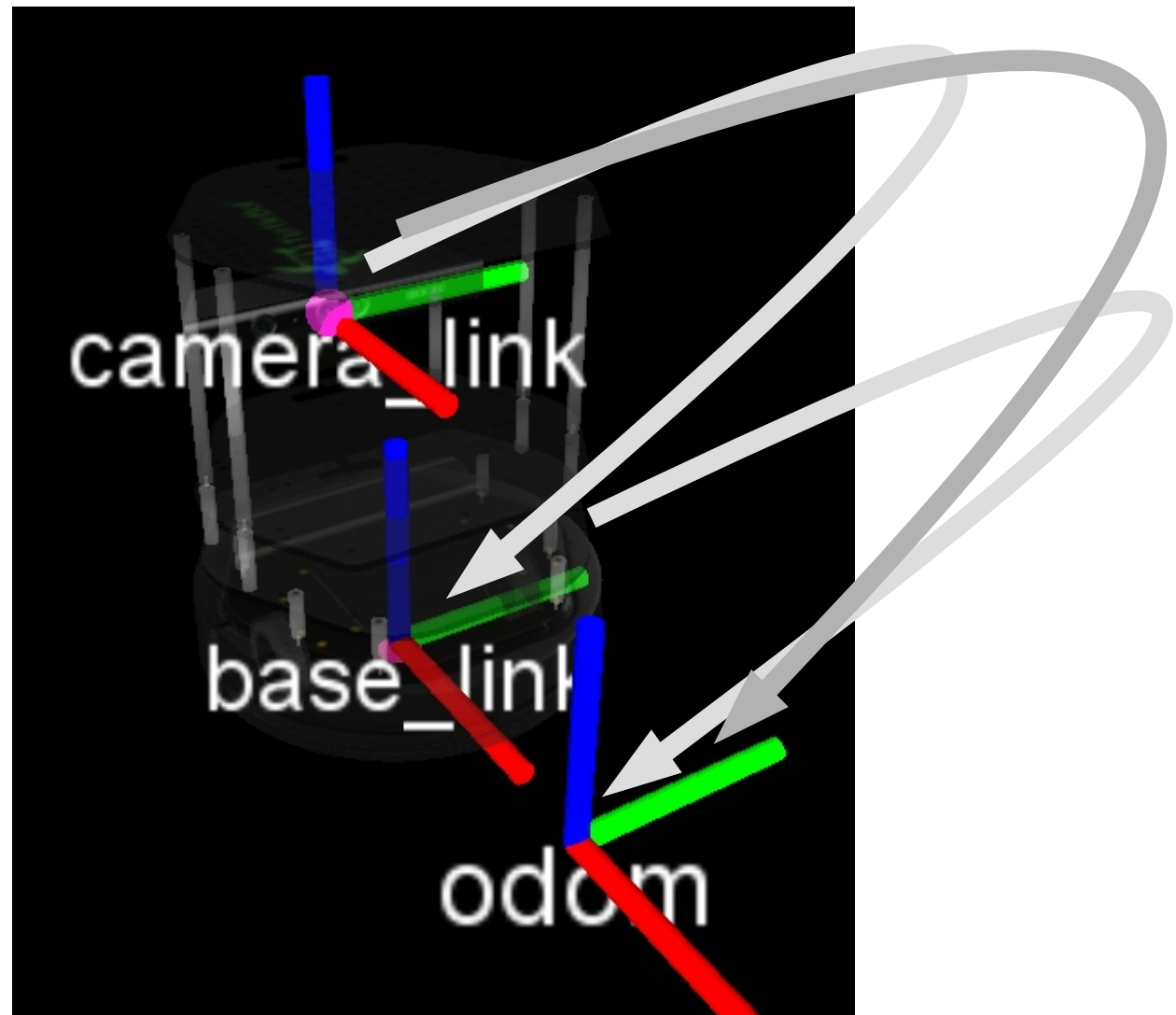
("right-handed coordinate system")



Recall: transformations  
between multiple frames



# Reference frames / transforms in ROS (TF)



# ROS `tf` & `/tf`

(note: transformation graph  $\neq$  topic graph)

`/tf`

- A topic.
- Any node can publish transformations 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\_link

transform:

translation:

x: -0.135

y: 0.0

z: 0.009

rotation:

x: -0.707106781185

y: 0.0

z: 0.0

w: 0.707106781188

# ROS tf & /tf

(note: transformation graph  $\neq$  topic graph)

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 transf;
...
listener.lookupTransform("base_link", "arm_link_3",
ros::Time(0), transf);
```

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

# Example: StampedTransform data structure

```
double robot_x = transf.getOrigin().x();  
double robot_y = transf.getOrigin().y();  
tf::Quaternion rotation = transf.getRotation();  
tf::Vector3 axis = rotation.getAxis();  
double angle = rotation.getAngle();  
double robot_theta = angle*axis[2];
```



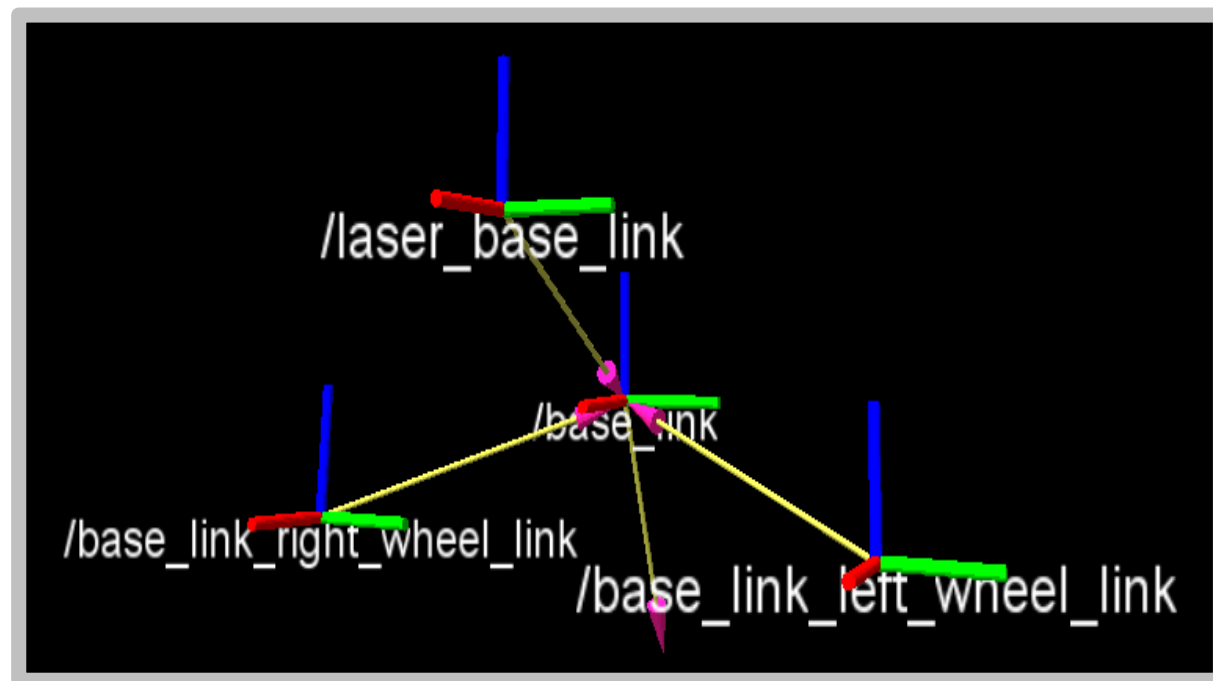
# Tools for debugging tf

To visualise use:

```
roslaunch tf view_frames
```

(produces frames.pdf)

Or add tf to rviz.



# Tools for debugging tf

For fun, try:

```
rostopic echo /tf
```

To see transform between “odom” and “base\_link” frames:

```
roslaunch 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/indigo/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)
```