CS 309: Autonomous Robots
FRI I

TF2

Instructor: Justin Hart

http://justinhart.net/teaching/2020_spring_cs309/

What is TF2

- TF2 is the ROS transform library
- It keeps track of spatial transformations between coordinate frames
- This allows ROS nodes to know the spatial poses of objects

- You do not subscribe to ROS topics to use TF2
- Instead, TF2 has a class that monitors transform data
- You request transform information from TF2

A quick overview of TF classes and methods

- tf::TransformListener listener
  - Listens on behalf of the TF library so it can compute transforms between frames
- tf::StampedTransform transform
  - A spatial transformation (timestamped)

```
listener.lookupTransform("odom", "base_link",
        ros::Time::now(), transform);
```

- Looks up the transform between "base_link" and "odom"

A quick overview of TF classes and methods

```
listener.waitForTransform("odom", "base_link", ros::Time(0),
        ros::Duration(4));
```

- TF listens to ROS topic information for spatial transforms
- It may not have heard the information needed to compute a transform
- If you request a transform that TF cannot compute, it throws an error
- This waits until the information is available before proceeding, to prevent the error

You can also send transform data

```
tf:: TransformBroadcaster br
br.sendTransform(
        tf::StampedTransform(
                transform, ros::Time::now(),
                fromFrame, toFrame));
```

- tf::StampedTransform
  - A spatial transformation (timestamped)
- ros::Time::now()
  - Makes the timestamp now
- fromFrame, toFrame
  - The names of the frames transformed between

Poses & Transformations

- Pose
  - Position and orientation of an object
  - Generally expressed as a position (x,y,z) and a rotation matrix
- Transformation
  - The relationship between two poses
  - Generally expressed as a translation (a position) & a rotation
- ROS has types for both
- Both are “different” in theory
- But both contain the same types of data, and are kind of interchangeable, conceptually.

geometry_msgs::Pose pose

pose.position
- pose.position.x
- pose.position.y
- pose.position.z

pose.orientation
- pose.orientation.x
- pose.orientation.y
- pose.orientation.z
- pose.orientation.w

Orientation is expressed as a quaternion

We will translate other representations into quaternions

tf::transform transform

transform.getOrigin()
- transform.getOrigin.x()
- transform.getOrigin.y()
- transform.getOrigin.z()

tf::vector3 origin(x,y,z)

transform.setOrigin(origin)

tf::Quaternion q(x,y,z,w)

transform.setRotation(q)

tf::Quaternion also supports
- Euler Angles
- Axis and Angle

Frames in the BWIBot Simulation

roscore

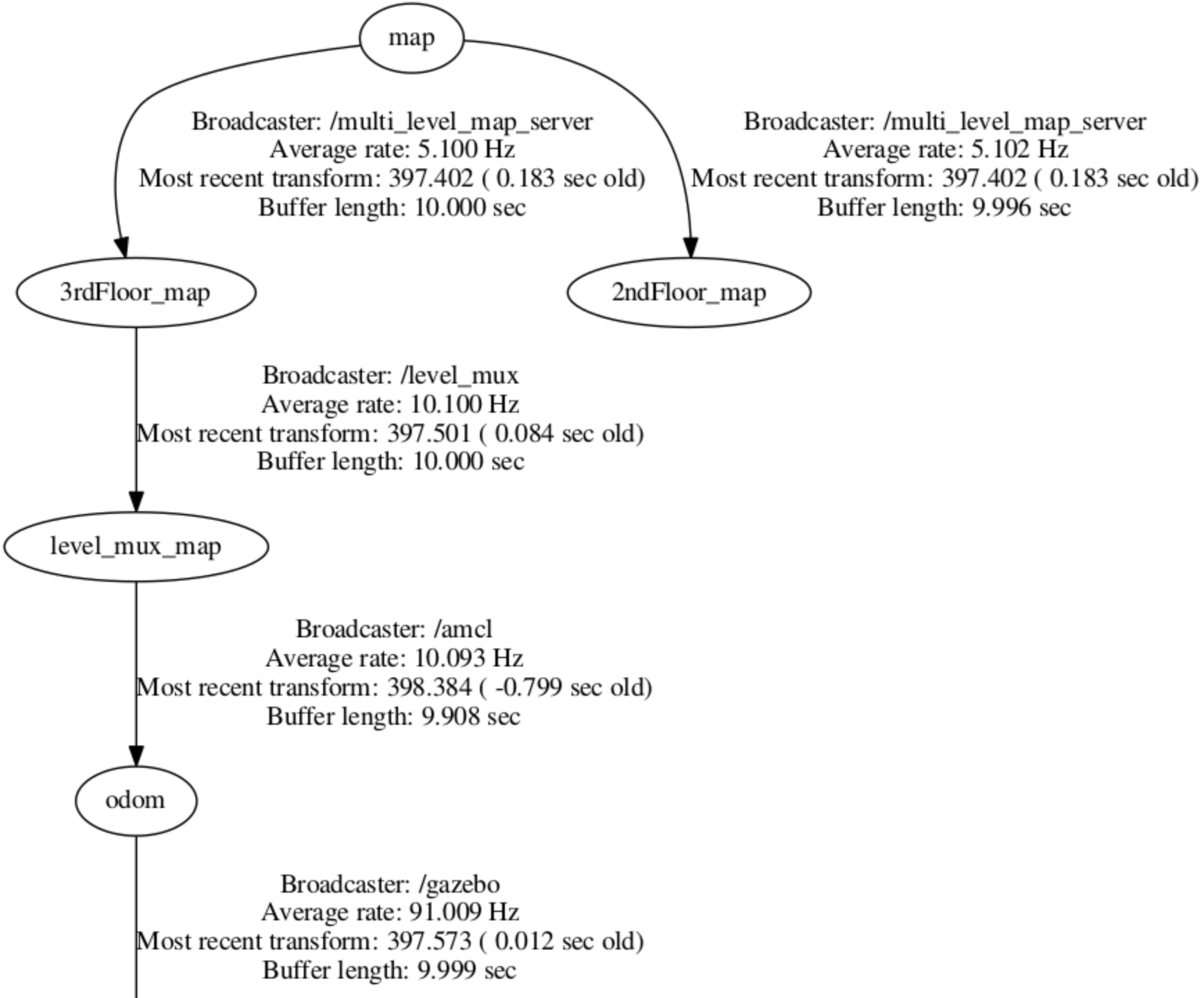
roslaunch bwi_launch simulation_v2.launch

To focus on the robot, pick “base_footprint” as the “fixed_frame” in the left column in rviz.

TF view_frames

rosrun tf view_frames

This generates a PDF of the current TF tree


map
Broadcaster: /multi_level_map_server
Average rate: 5.100 Hz
Most recent transform: 397.402 (0.183 sec old)
Buffer length: 10.000 sec
Broadcaster: /multi_level_map_server
Average rate: 5.102 Hz
Most recent transform: 397.402 (0.183 sec old)
Buffer length: 9.996 sec
3rdFloor_map
2ndFloor_map
Broadcaster: /level_mux
Average rate: 10.100 Hz
Most recent transform: 397.501 (0.084 sec old)
Buffer length: 10.000 sec
level_mux_map
Broadcaster: /amcl
Average rate: 10.093 Hz
Most recent transform: 398.384 (-0.799 sec old)
Buffer length: 9.908 sec
odom
Broadcaster: /gazebo
Average rate: 91.009 Hz
Most recent transform: 397.573 (0.012 sec old)
Buffer length: 9.999 sec

“map”

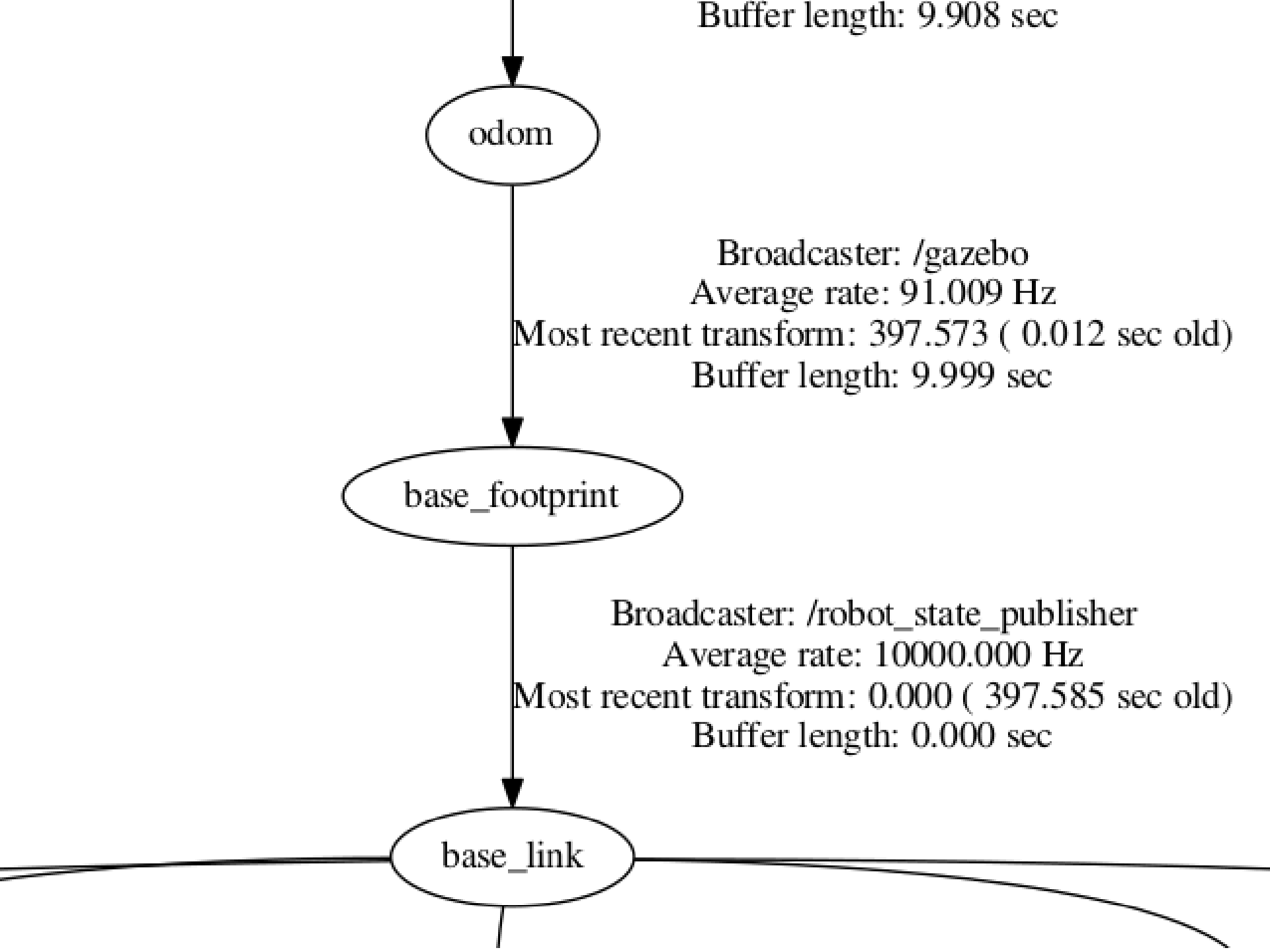
- Our “global” frame. Sometimes called the “inertial frame”
- The top frame.

“3rdFloor_map” and “level_mux_map”

- Part of the BWI map service. You can safely ignore them for now.

“odom”

- Short for “odometry.” Measuring the robot’s motion.
- Odom is placed in a fixed position.
- The robot’s motion is tracked relative to this.
- This fixed position is relative to the map.


Buffer length: 9.908 sec
odom
Broadcaster: /gazebo
Average rate: 91.009 Hz
Most recent transform: 397.573 (0.012 sec old)
Buffer length: 9.999 sec
base_footprint
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (397.585 sec old)
Buffer length: 0.000 sec
base_link

“base_footprint”

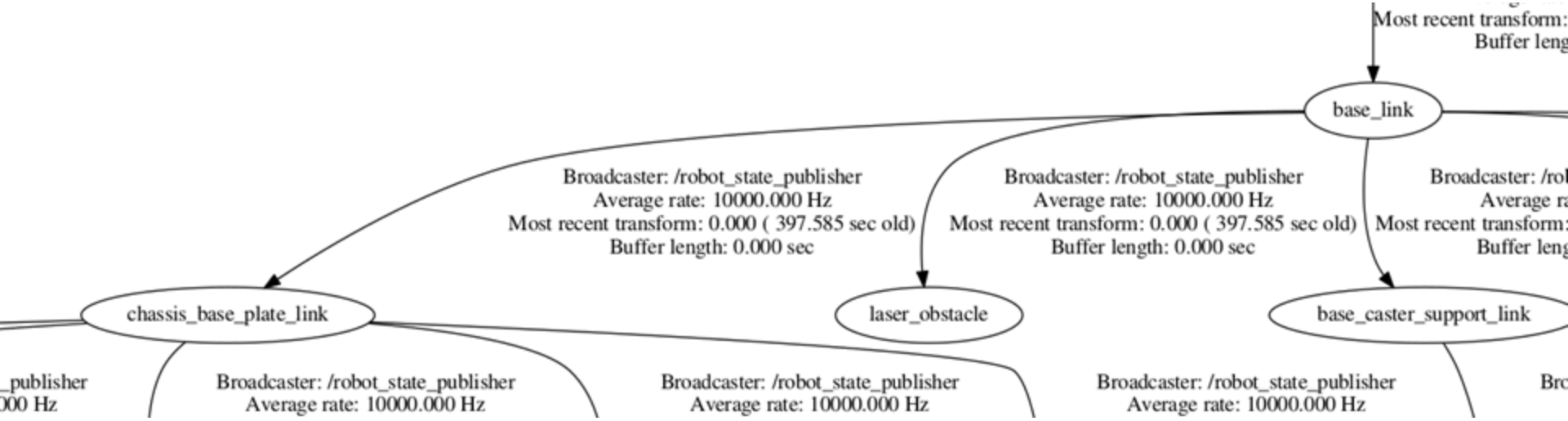
- Where the robot is
- base_footprint is at (0,0,0) relative to base_footprint
- So we track relative to the odom frame

We can watch this from the command line

rosrun tf tf_echo /base_footprint /odom

rosrun segbot_bringup teleop_twist_keyboard

We can see the translation changing!


Most recent transform:
Buffer leng
base_link
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (397.585 sec old)
Buffer length: 0.000 sec
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (397.585 sec old)
Buffer length: 0.000 sec
Broadcaster: /rob
Average ra
Most recent transform:
Buffer leng
chassis_base_plate_link
laser_obstacle
base_caster_support_link
_publisher
000 Hz
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Broadcaster: /robot_state_publisher
Average rate: 10000.000 Hz
Bro

The robot has many frames

“base_link”

- base_footprint is a position in the plane of the floor
- base_link is where the physical, mechanical base of the robot is

“chasis_base_plate_link”

- Where parts are mounted on the BWIBot

“laser_obstacle”

- Where the laser sees the nearest obstacle

“caster_wheel_link, base_link_left_wheel, base_link_right_wheel”

- The robot’s wheels

Many many more links are used for everything we care about on the robot