CS 309: Autonomous Intelligent Robotics
FRI I

Lecture 18:
Coordinate Frames
TF2
Alvar

Instructor: Justin Hart

http://justinhart.net/teaching/2018_spring_cs309/

A couple of quick notes

- Homework 4
  - Due March 29
  - You need to do this **in front of a mentor**
  - Do it soon. You will run out of time.
- Robotics Study
  - If you’re free, we appreciate the help. See the Canvas announcement.

Looking forward

- Homework 5
  - Goes out March 29
  - Due April 12
- Homework 6
  - Also Due April 12
  - Final project prospectus!

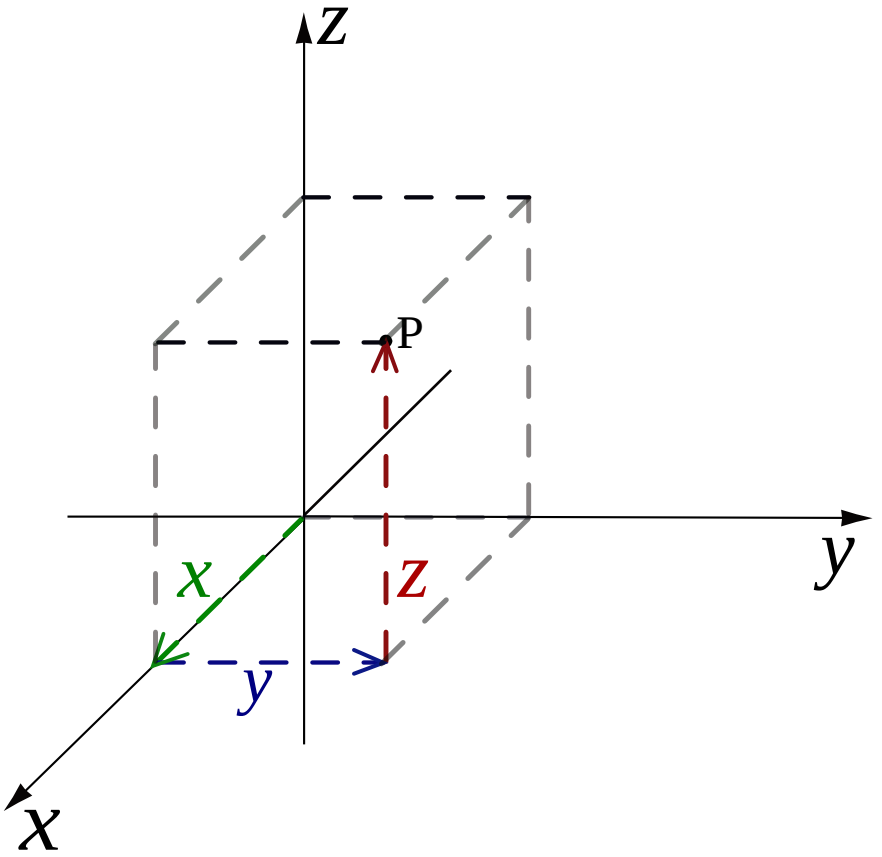
Today

- Coordinate Frames
- ROS TF2

Coordinate Frames

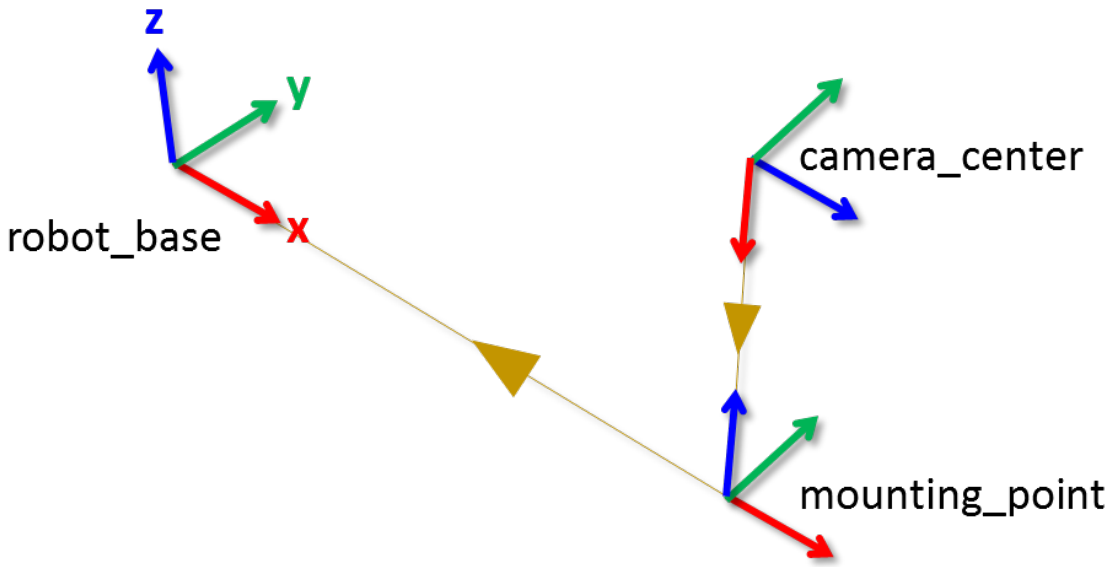
- We use them to discuss poses
  - The robot
  - Its parts
  - Tracked objects
  - Landmarks
- Two coordinate frames are related by a **transformation**
- Transformations are relative to pairs of coordinate frames
- Transformations are generally tracked as a hierarchy

3D Coordinate Frame



Coordinate Frames

- With robots we generally are discussing multiple coordinate frames relative to each other
- We often visualize them similar to this



TF2

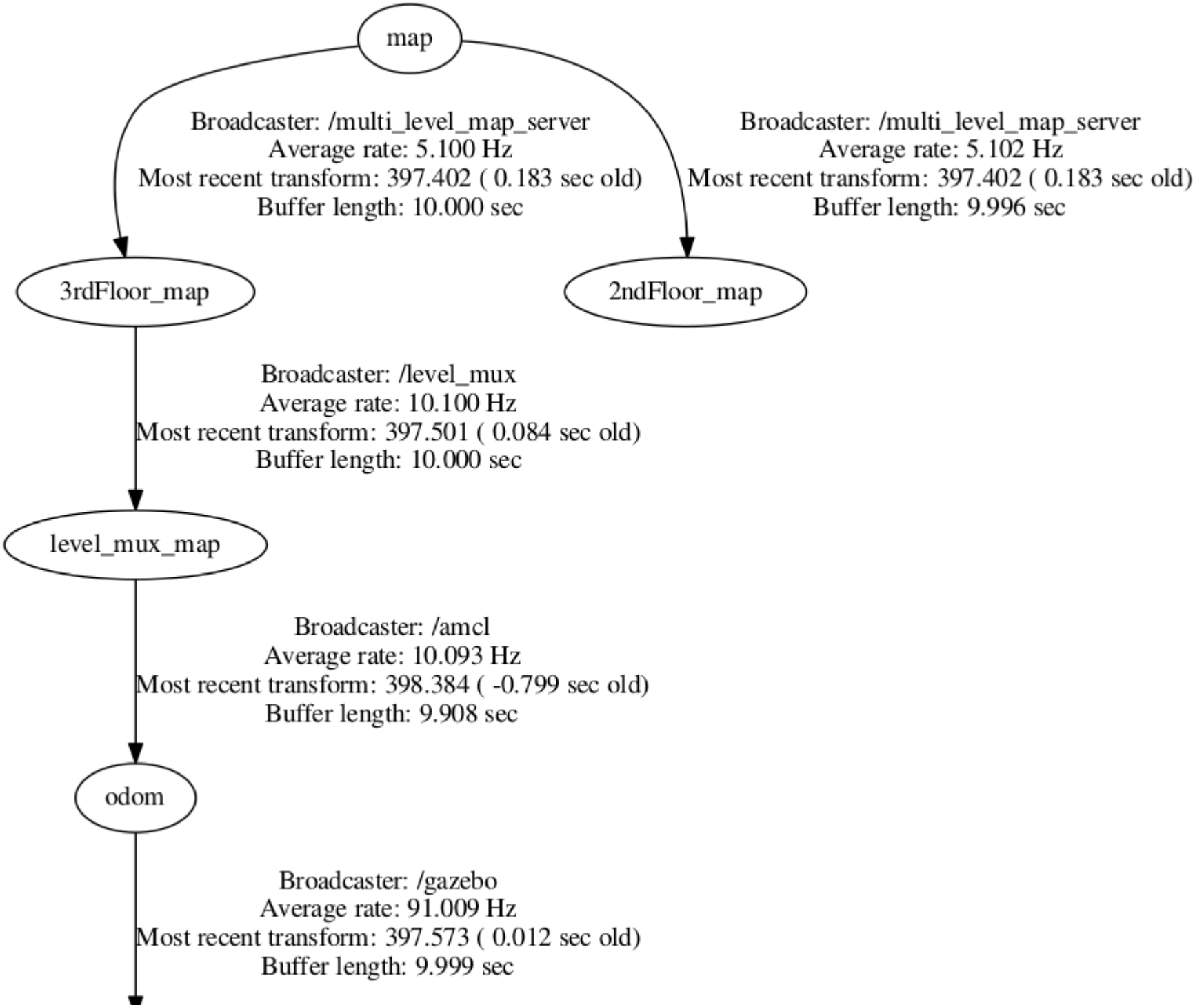
- The Transform Library
  - Replaces TF, hence being TF2
- Organizes transforms as a tree
  - Remember, transforms are relative to each other
  - The tree keeps track of how transforms relate to each other
  - Each transform tells you where something is
  - This is always relative to some other frame

Example: In simulation

- roscore
- roslaunch bwi_launch simulation_v2.launch
- If we select “Global Options” on the left, we can change “Fixed Frame” to base_footprint, telling us to move rviz so it is focusing on the robot

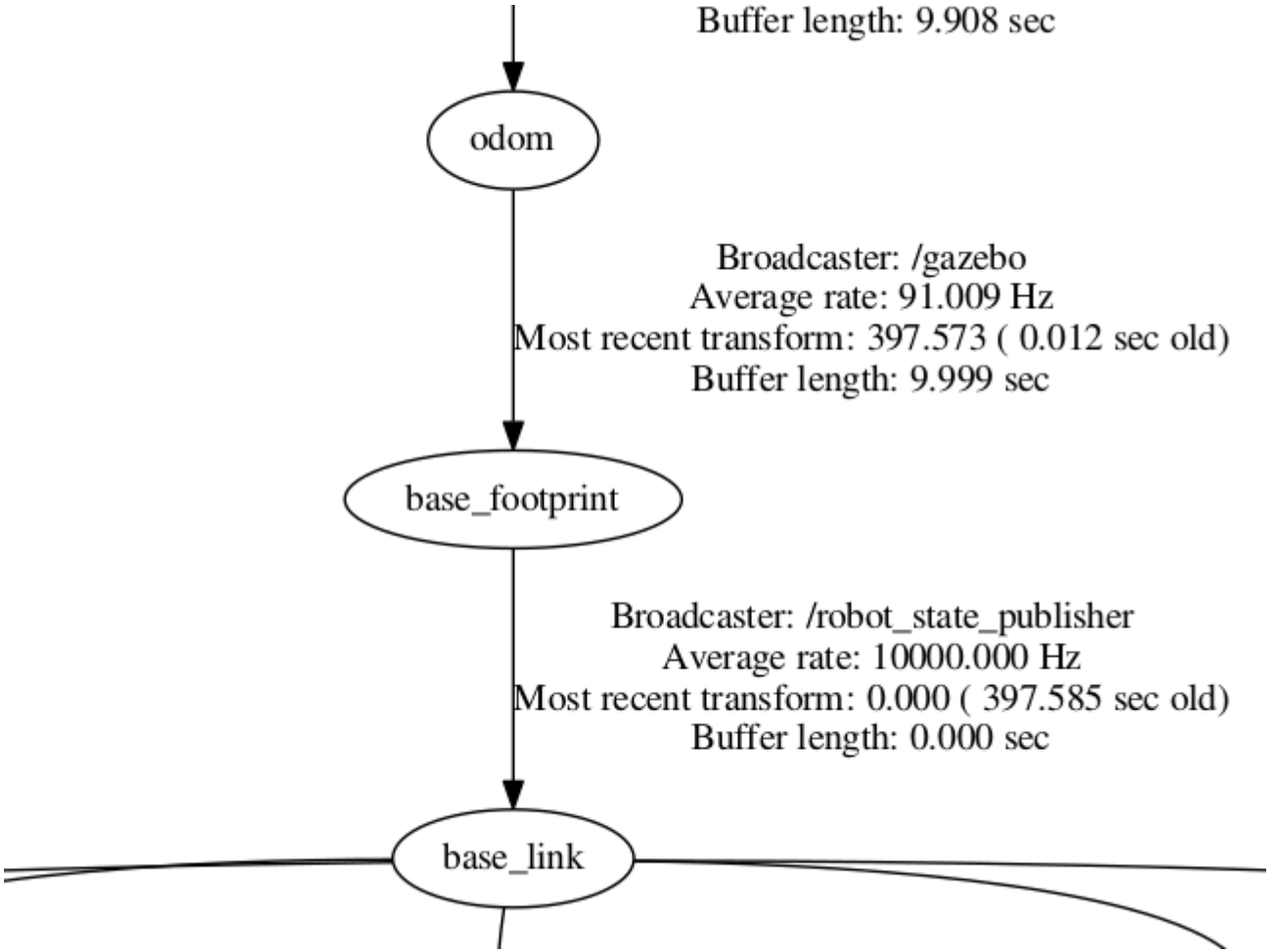
TF view_frames

- rosrun tf view_frames
  - This will generate 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

A few of these frames

- BWI uses a special map service to tell the robot which floor it is on
- “map” is our “global frame,” sometimes called the “inertial frame”
  - It is the top level, all frames are relative to it
- “3rdFloor_map” and “level_mux_map” are from the BWI map service.
  - You can safely ignore them
- “odom”
  - Short for “odometry.”
  - When we track the robot’s motion, it is relative to “odom.”


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

Where is the robot?

- odom
  - The robot’s motion is tracked relative to odom
  - odom is generally in a fixed position
- base_footprint
  - Where the robot is
  - Remember, base_footprint is at (0,0,0) in base_footprint’s frame
  - So we compare to odom to know where the robot is

Example

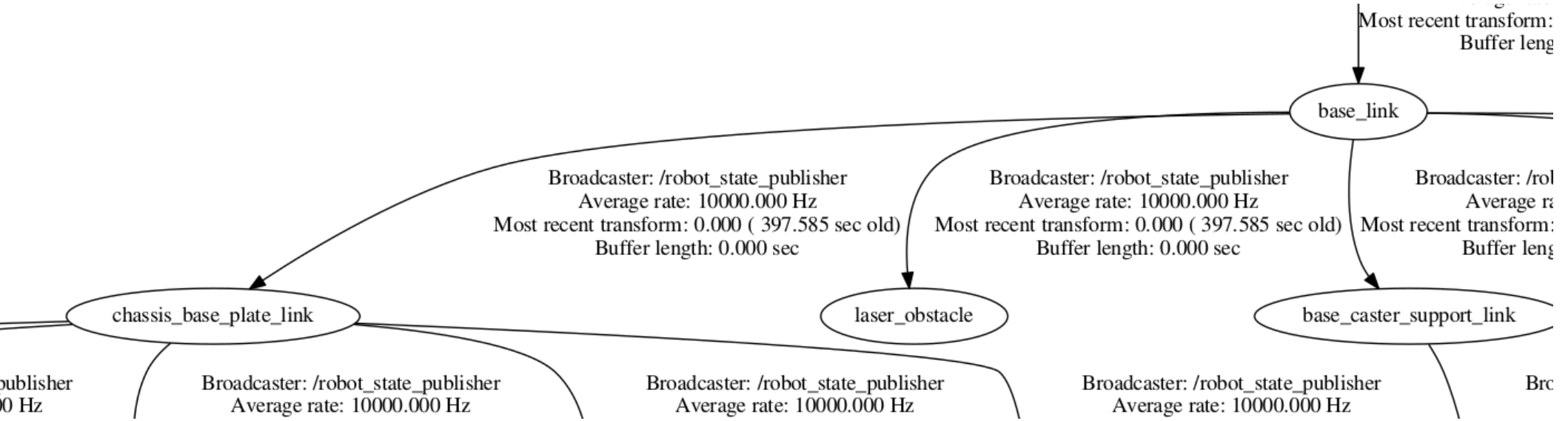
- We can watch this from the command line

- rosrun tf tf_echo /base_footprint /odom
- rosrun segbot_bringup teleop_twist_keyboard

- If we watch tf_echo, we can see the translation changing.
  - This is how our system knows where the robot is!

Example Code

- Work through tf_example


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: /rol
Average ra
Most recent transform:
Buffer leng
chassis_base_plate_link
laser_obstacle
base_caster_support_link
publisher
00 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

More frames

- The robot has MANY frames
- base_link
  - The physical base of the robot's mechanics
    - Rather than base_footprint, where it is on the floor
    - Decoupling the two simplifies modeling
- 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

The TF2 Library

AR Tags

- Let’s take a look at tracking an object and finding its coordinate frame.
- Alvar
  - An open source library for tracking AR Tags

Virtual Reality

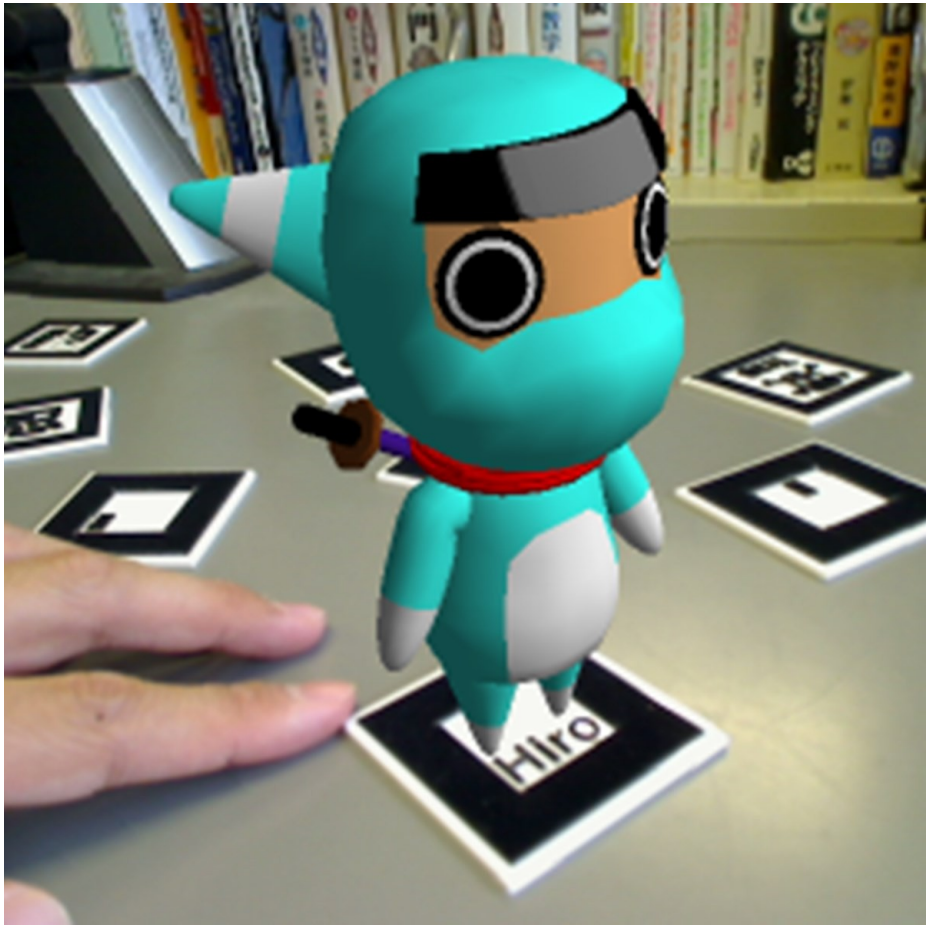
- The simulation of a 3D world that can be experienced first-hand
- 3D
  - Video games
  - Virtual worlds
  - Headsets

Augmented Reality

- Adds 3D content to images and perception of the real world
- The photo on the right merges real content and virtual content in a real-estate application

Augmented Reality Toolkit

- What is an AR Tag?
  - Augmented Reality Tag
  - The image on the right comes from a demonstration of Augmented Reality Tool Kit
  - Provides a coordinate frame to render 3D content onto
  - We will use a similar package called Alvar



How does this work?

- Recall our transforms
  - Translation
  - Rotation
- Augmented Reality uses *projective transformations*
  - Homographies
  - Projections

For your next homework..

- Work out the Direct Linear Transformation (DLT) of a homography to resolve its position in space
  - Just kidding
- From our perspective, AR Tags provide us with
  - Translation
  - Rotation
    - And work with TF2

ar_track_alvar

- ROS package for using AR Tags!
- Needs to be configured for your camera
  - This happens in a .launch file
- Generate AR Tags
- Print them
- Attach them to a rigid surface
- Your system can now track an AR Tag
- Example