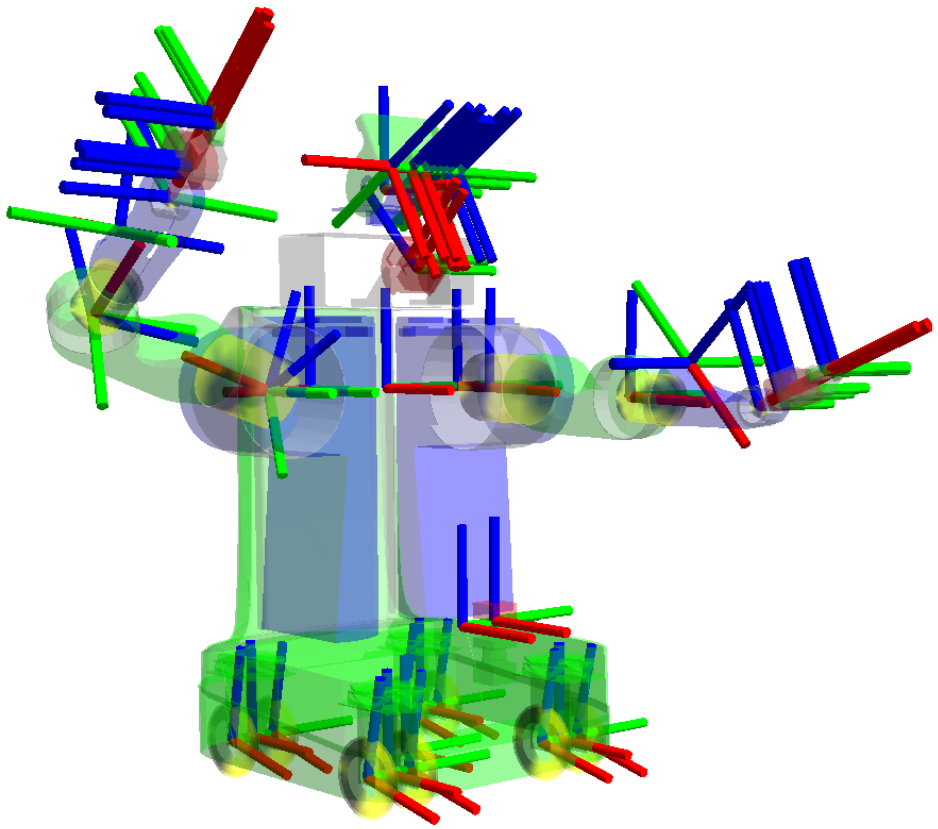
TF (transform) in ROS

ECET 49900/58100

Credit: PhD comics and Willow Garage

ROS tf (transform) package

- Goal: Maintain relationship between multiple coordinate frames overtime. Transform points, or vectors between two coordinates.



Source: http://wiki.ros.org/tf

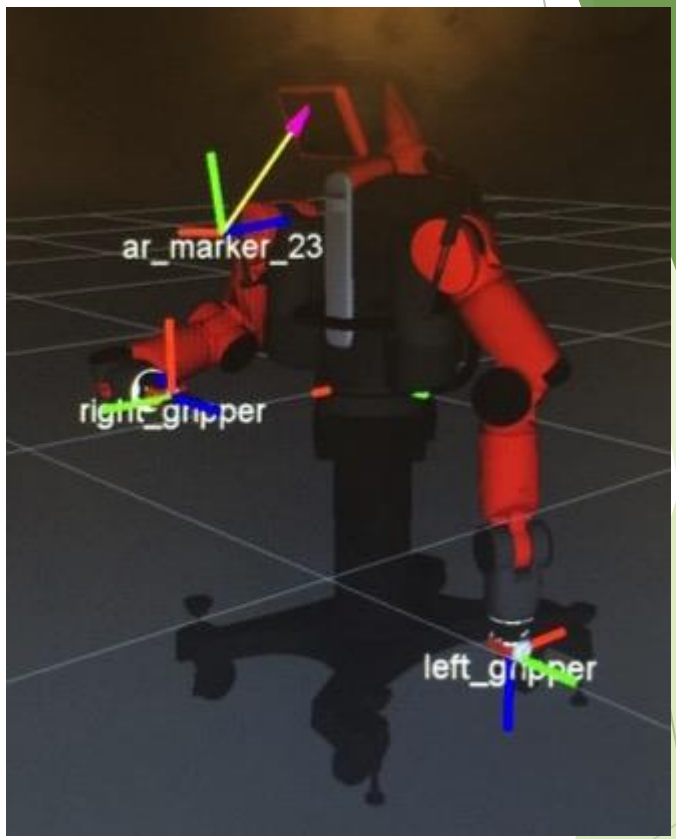
- Published to the system which can be accessed by any node subscribed to it!

- *note tf package is deprecated in favor of the more powerful **tf2_ros** package

(See! The cycle continues…)

ROS tf (transform) package

- What are we using it for?
  - Autonomous driving, transform sensor data to map coordinate



- Transform robot coordinates

Source: http://www.kendo.flippen.se/

Source: http://web.ics.purdue.edu/~rvoyles/Classes/ROSprogramming/index.html

Using ROS tf (transform) package to transform between coordinate frames

2 main tasks that users generally use tf for transform between coordinates: broadcasting and listening.

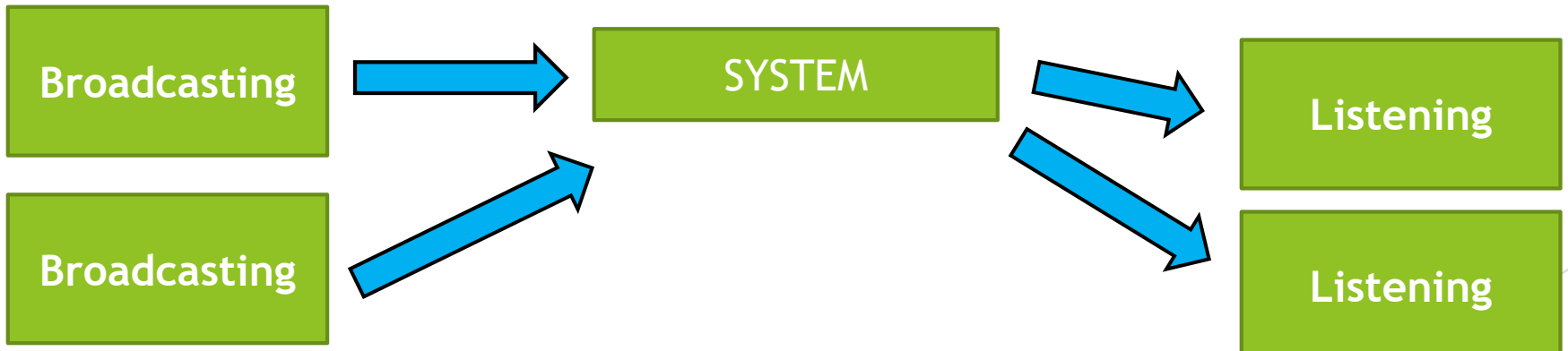
Broadcasting transforms:

Publish the relative pose and coordinate to the system

This allow us to setup our own relationship between two coordinate frames

Listening transforms:

Specify the published transform and query the specific transform between coordinate frames (not quite the same as Subscribing to a Topic)



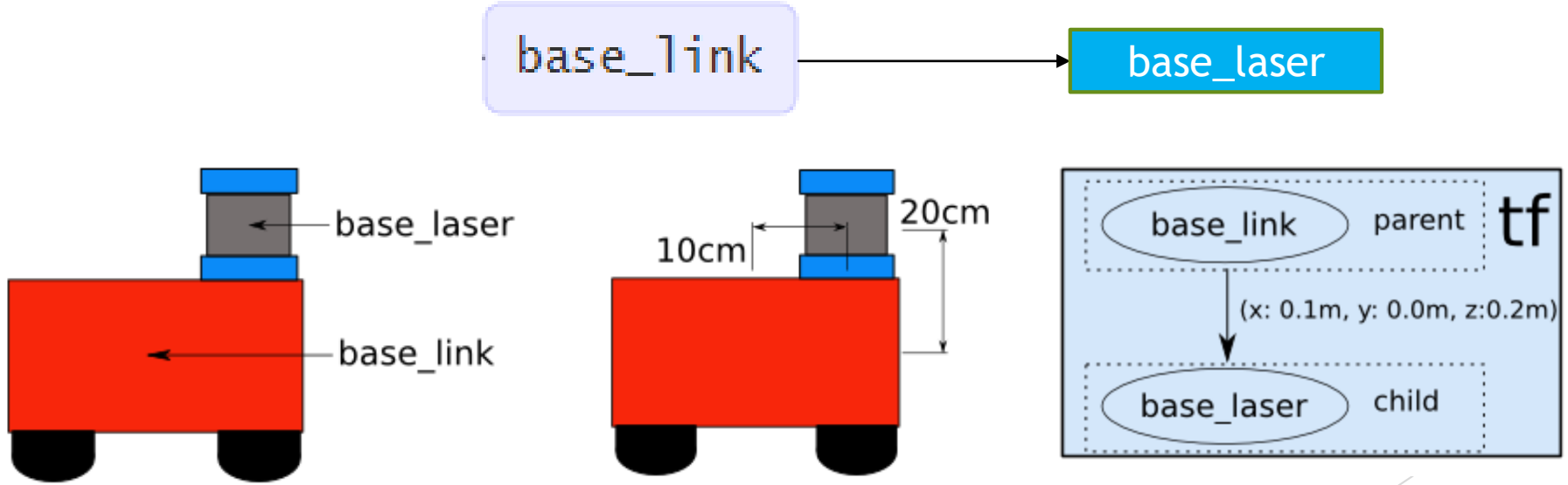
Source: http://wiki.ros.org/tf

Example of using TF broadcasting

to update the base_laser frame

Our goal of using the TF broadcasting is to define and establish the relationship between two different coordinate frames, `base_link` and `base_laser`, and build the relationship tree of the coordinate frames in the system.

First step, we have to first define the which is “parent” and “child” because TF defines the “forward transform” as transforming from parent to child. The “inverse transform” goes the other way (and we know how to specify both).



Source: http://wiki.ros.org/navigation/Tutorials/RobotSetup/TF

Cont: Example of using TF2 broadcasting

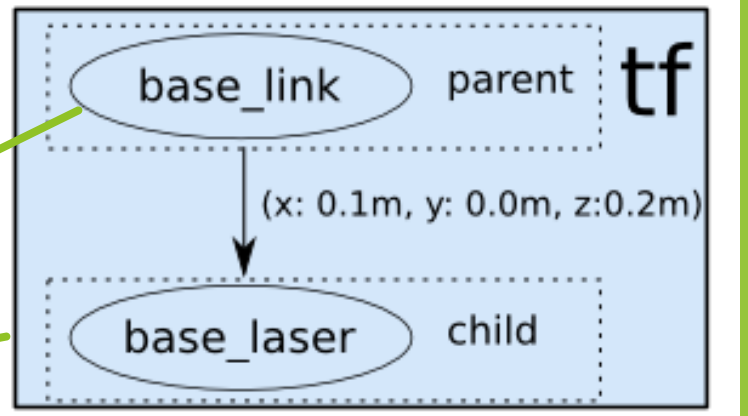
update base_laser frame whenever base_link gets updated

Then, we setup a broadcast:

```
import rospy, tf2_ros, geometry_msgs.msg
def callback(data):
    tf2Broadcast = tf2_ros.TransformBroadcaster()
    tf2Stamp = geometry_msgs.msg.TransformStamped()
    tf2Stamp.header.stamp = rospy.Time.now()
    tf2Stamp.header.frame_id = 'base_link'
    tf2Stamp.child_frame_id = 'base_laser'
    tf2Stamp.transform.translation = (0.1, 0.0, 0.2)
    tf2Stamp.transform.rotation = (0.0, 0.0, 0.0)
    tf2Broadcast.sendTransform(tf2Stamp)

if __name__ == "__main__":
    rospy.init_node("talker")
    rospy.Subscriber('topic_name', 'message_class', callback)
    rospy.spin()
```

*** See source for full implementation**



name of parent

name of child

Forward transform between them

Not “Published” as a separate Topic - “Looked up” instead

Topic that updates base_link

Source: http://wiki.ros.org/tf/Tutorials/Writing%20a%20tf%20broadcaster%20%28Python%29

Source: http://wiki.ros.org/navigation/Tutorials/RobotSetup/TF

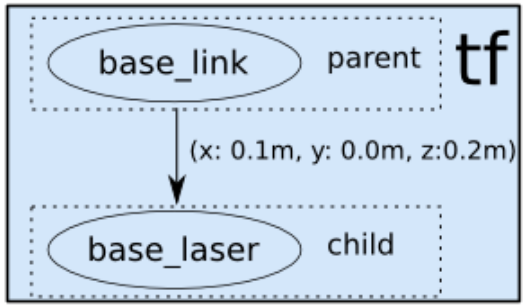
Example of using TF2 listening

TF Listener will access into the existing TF relationship tree and return the relationship between coordinate frames, or even transform points for you. Not the same as “Subscribing”
Then, we setup a listener:

```
import rospy, tf2_ros, geometry_msgs.msg
if __name__ == '__main__':
    rospy.init_node('listener')
    listener = tf2_ros.TransformListener()

    rate = rospy.Rate(10.0)
    while not rospy.is_shutdown():
        try:
            (trans,rot) = listener.lookupTransform('base_link',
            'base_laser', \
                rospy.Time(0))
    # This will give you the coordinate of the child in the parent frame
        except (tf2_ros.LookupException, tf2_ros.ConnectivityException,
                tf2_ros.ExtrapolationException):
            pass
        rate.sleep()
```

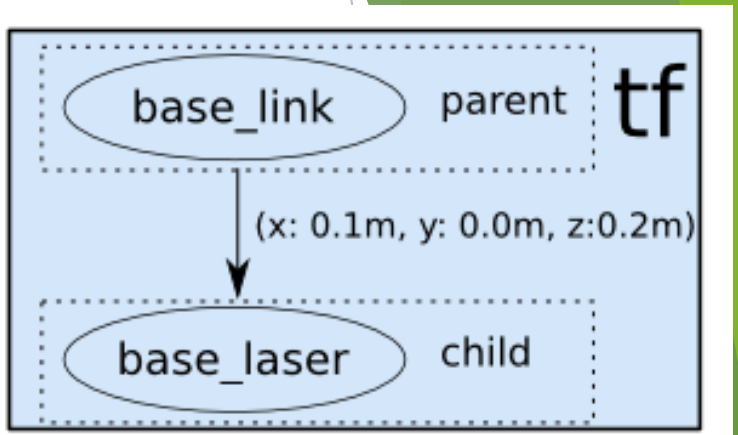
“Look up” the updated transform



*** See source for full implementation**

Source: http://wiki.ros.org/tf/Tutorials/Writing%20a%20tf%20listener%20%28Python%29
Source: http://mirror.umd.edu/roswiki/doc/diamondback/api/tf/html/python/tf_python.html

Example of using TF2 point listening



Then, we setup a listener:

```
import rospy, tf2_ros, geometry_msgs.msg
if __name__ == '__main__':
     rospy.init_node('listener')
     listener = tf2_ros.TransformListener()

     rate = rospy.Rate(10.0)
     while not rospy.is_shutdown():
          pointstamp = PointStamped()
          pointstamp.header.frame_id = 'base_laser'
          pointstamp.header.stamp = rospy.Time(0)
          pointstamp.point.x = 1.0
          pointstamp.point.y = 2.0
          pointstamp.point.z = 3.0
          try:
               listener.transformPoint('base_link', pointstamp)
          except (tf2_ros.LookupException, tf2_ros.ConnectivityException,
                    tf2_ros.ExtrapolationException):
               pass
          rate.sleep()

# This will give you what is the coordinate in parent coordinate frame for (1,2,3) in child.
```

Defining the reference frame…

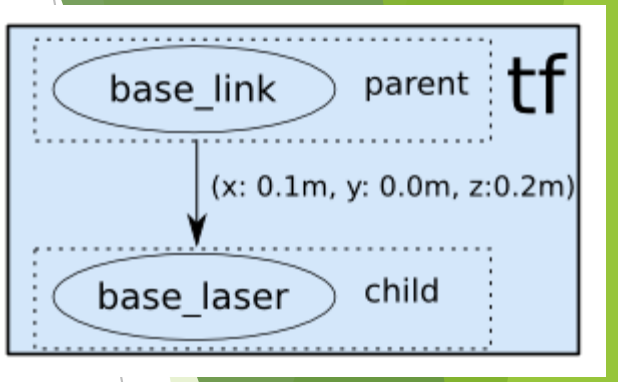
…of this point

Report it in this frame

*** See source for full implementation**

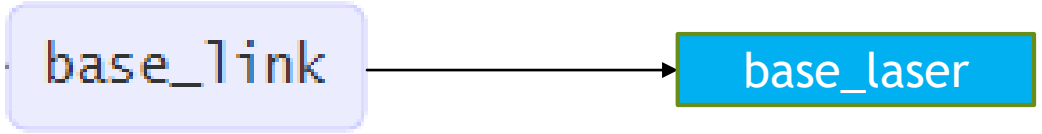
Source: http://wiki.ros.org/tf/Tutorials/Writing%20a%20tf%20listener%20%28Python%29
Source: http://mirror.umd.edu/roswiki/doc/diamondback/api/tf/html/python/tf_python.html

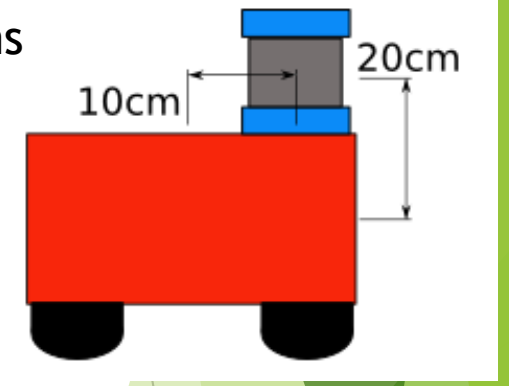
Example of using TF broadcasting staticTF



Another way to broadcast if the transformation is static?
Use “static_transform_publisher” in launch file

static_transform_publisher x y z yaw pitch roll frame_id child_frame_id period_in_ms





Example launch file command:

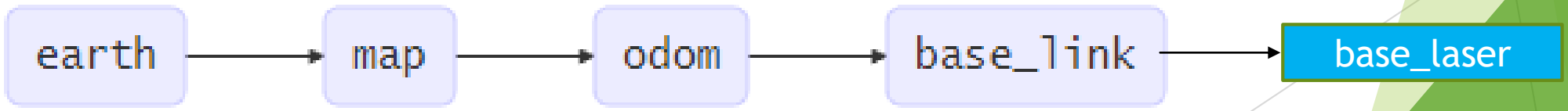
```
<node pkg="tf" type="static_transform_publisher"
name="base2laser_bcast" args= "0.1 0.0 0.2 0.0 0.0 0.0
base_link base_laser 100” />
```

This will transform the parent to the child according to the coordinate transformation input, and publish every 100 ms.

Source: http://wiki.ros.org/tf#static_transform_publisher

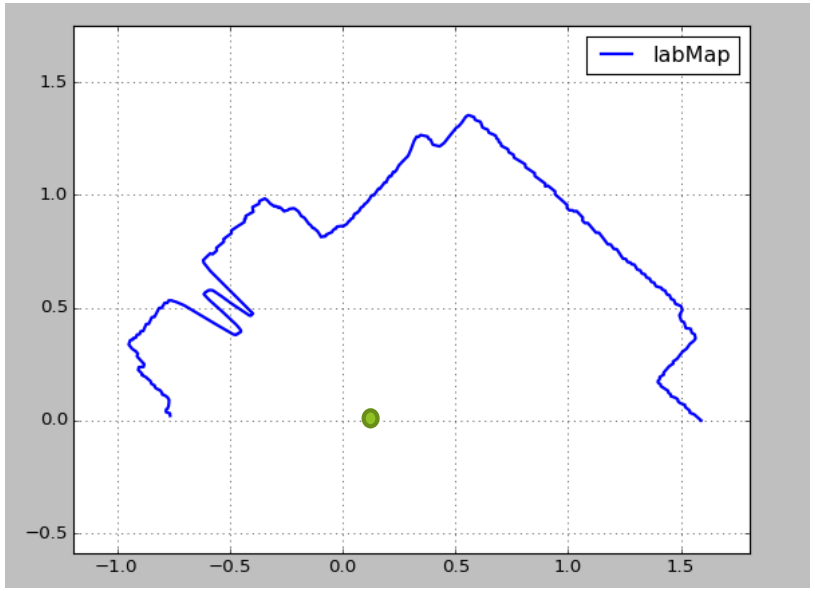
Coordinate frames for Laser Reading

- From ROS Enhancement Proposal (REP) #105
  - Base_link: **Rigidly attached to the mobile robot base.**
  - Odom: **World-fixed coordinate frame.**
    - The pose is continuous (no sudden jumps)
    - Accurate in short term, local reference! But accumulates errors in long term
  - Map: **World-fixed coordinate frame.**
    - Obtained from re-computing the position from sensor information
    - Not continuous (sudden jumps can occur!)

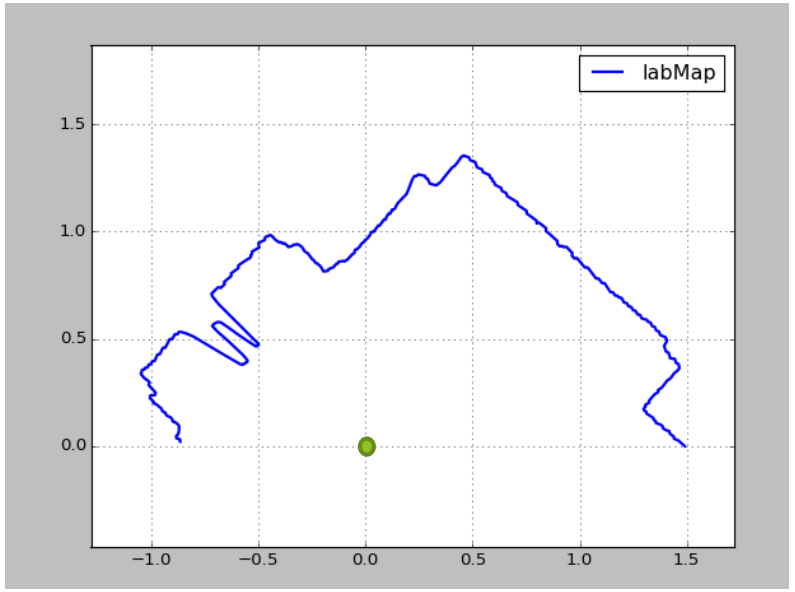


Source: http://www.ros.org/reps/rep-0105.html

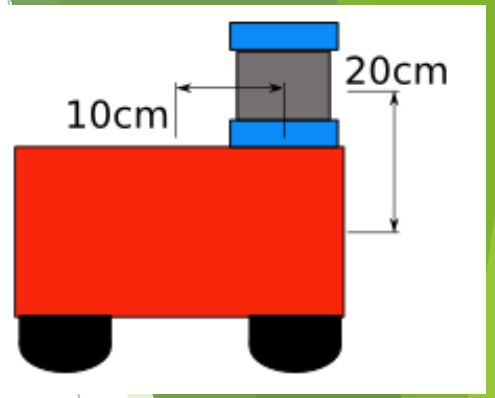
Coordinate frames for Laser Reading

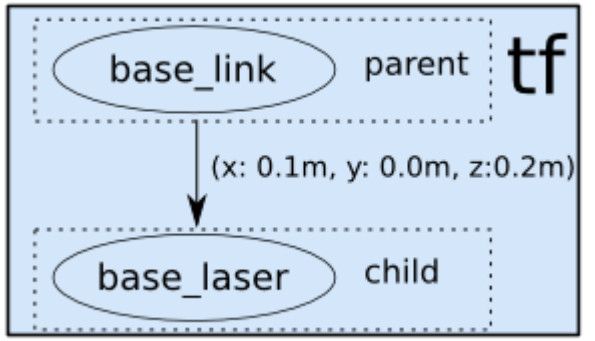


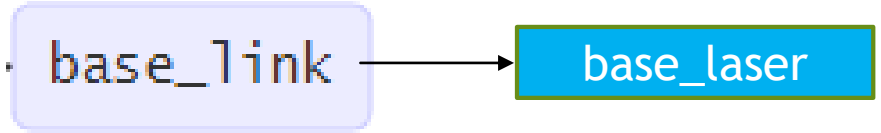
LIDAR reading in base_link coordinate frame



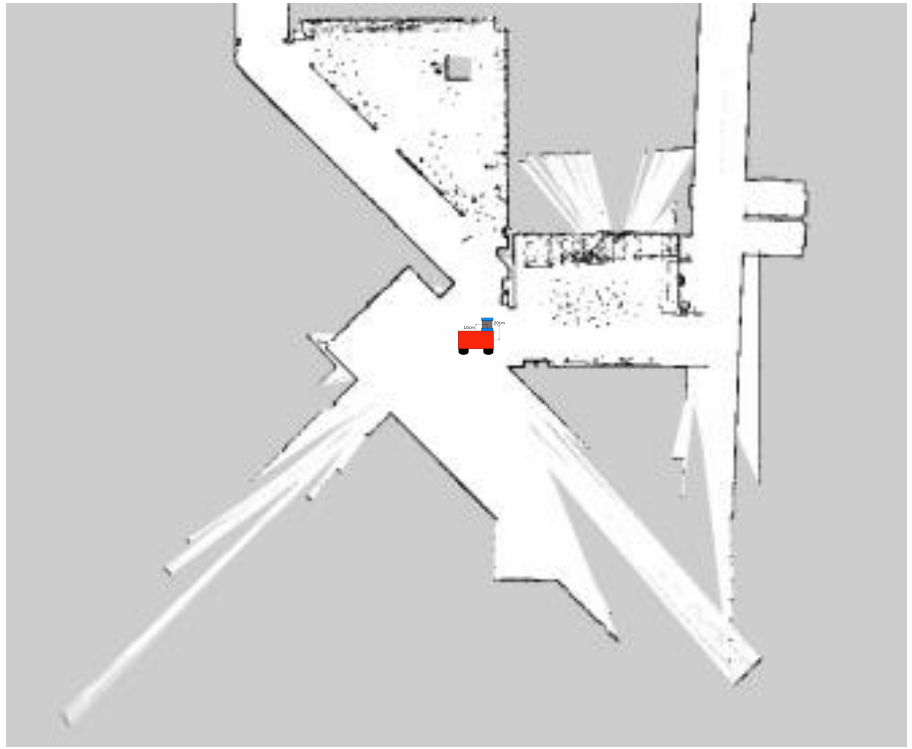
LIDAR reading in base_laser coordinate frame





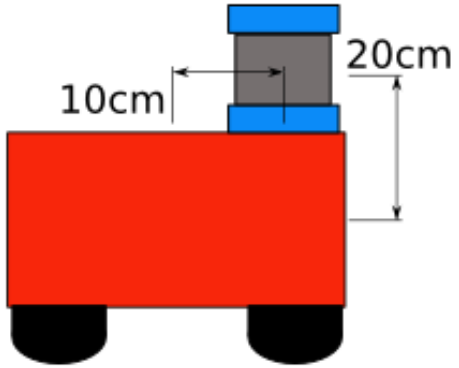


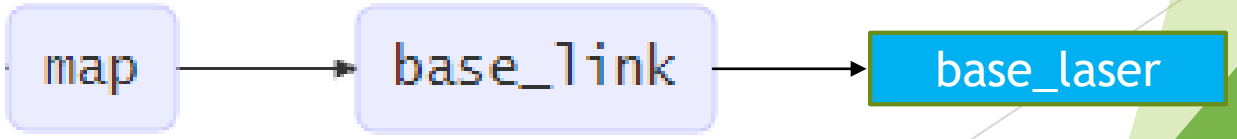
Coordinate frames for Laser Reading



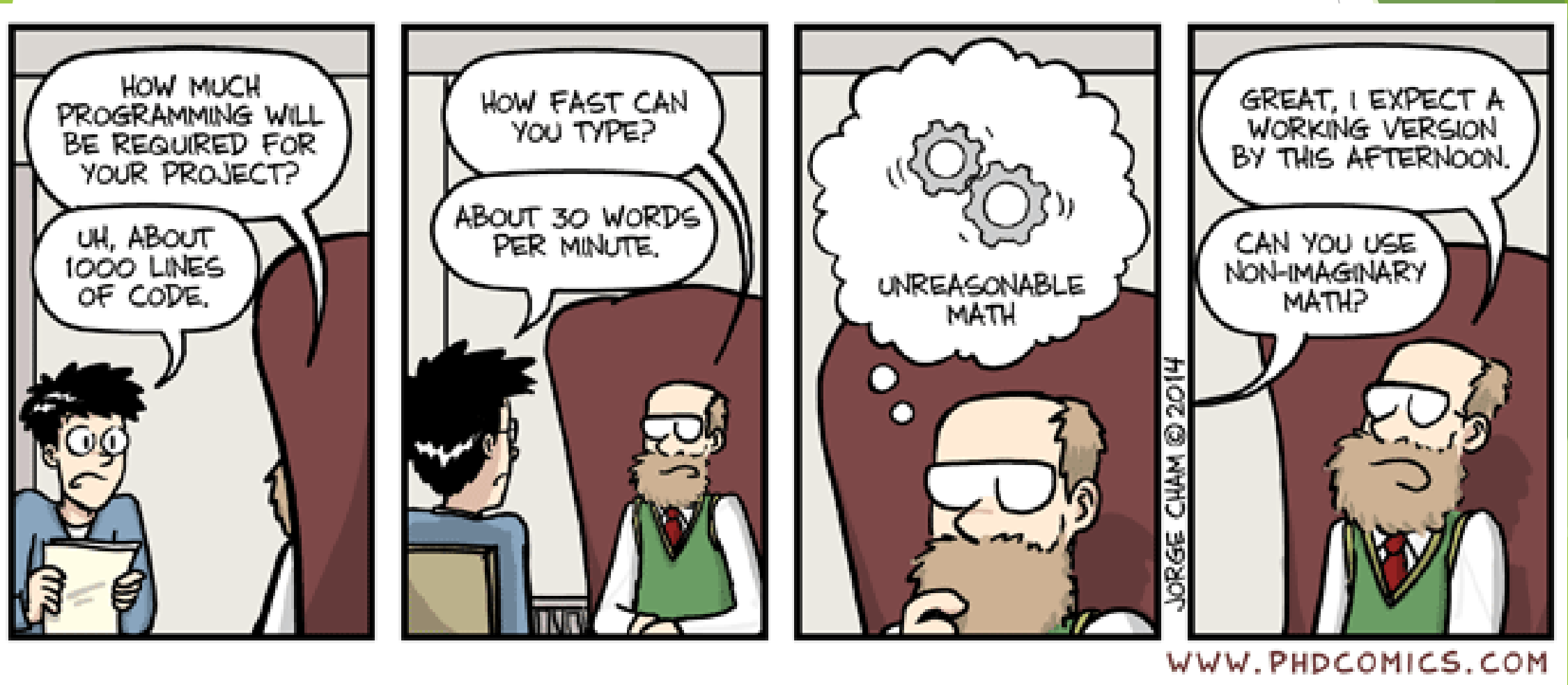
Adjust the TF broadcasting to “look up” the position/pose of base_link in map coordinates to know where the robot is in the map (localization).

[Later you will “Subscribe” to the AMCL topic that indicates when a new pose estimate is available.]





Now, we can implement tf!



Credit: PhD comics