Course of

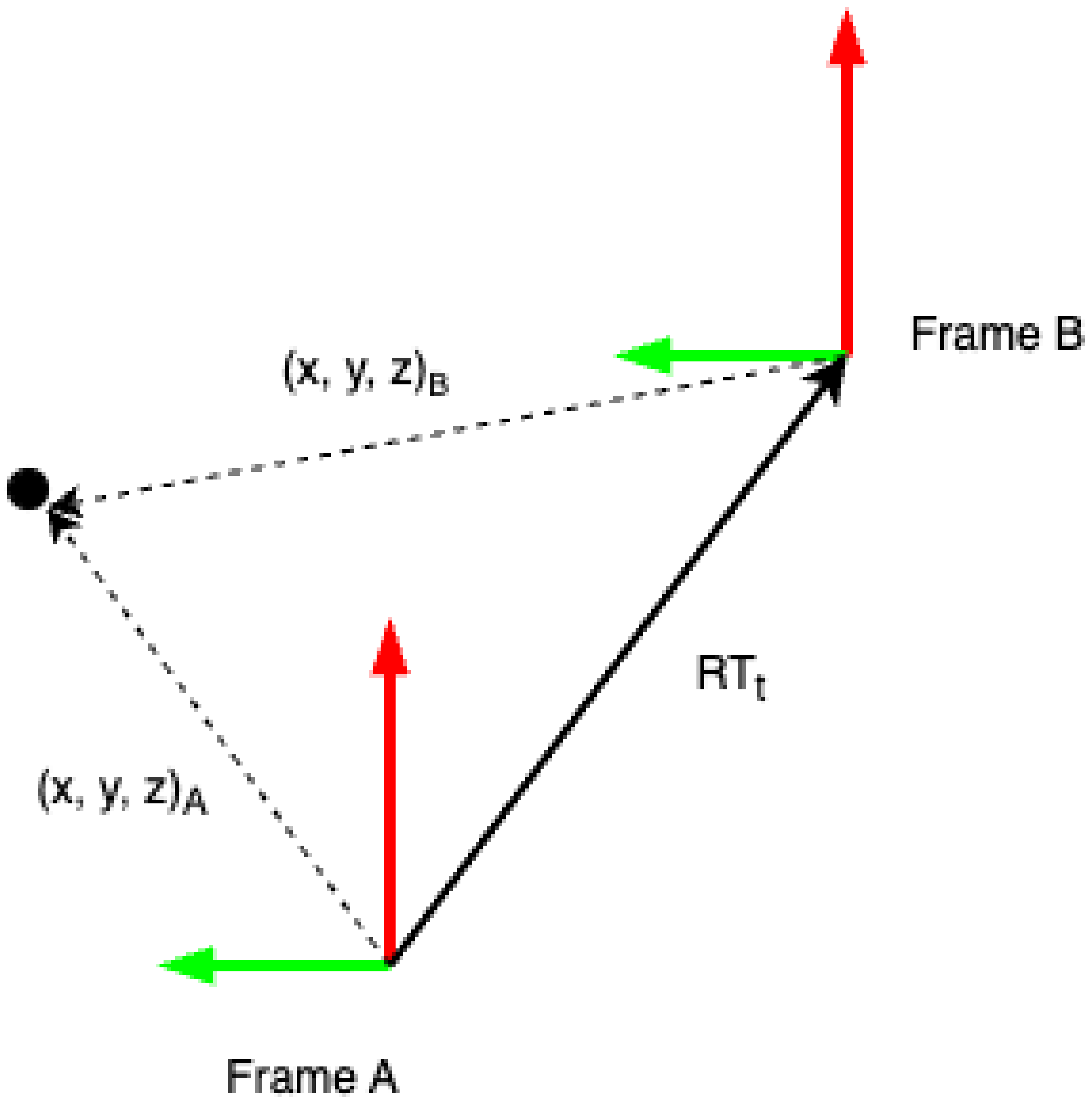
Robot Programming with ROS 2

Day 2

1. TFs System

Frames and Transformations

- One of the greatest treasures in ROS
- It allows to use and transform coordinates between different reference axes (**frames)**



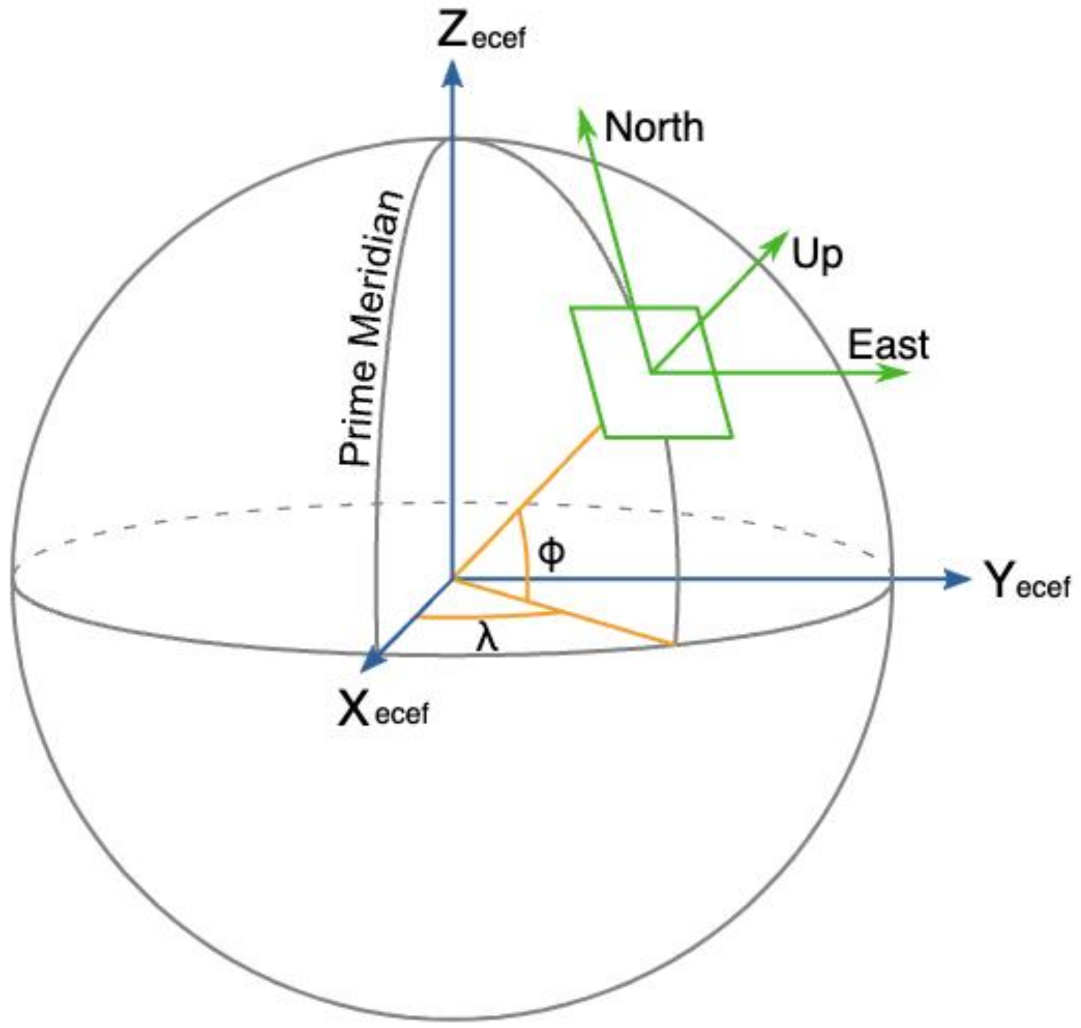
$$P_B = RT_{A\rightarrow B} * P_A$$

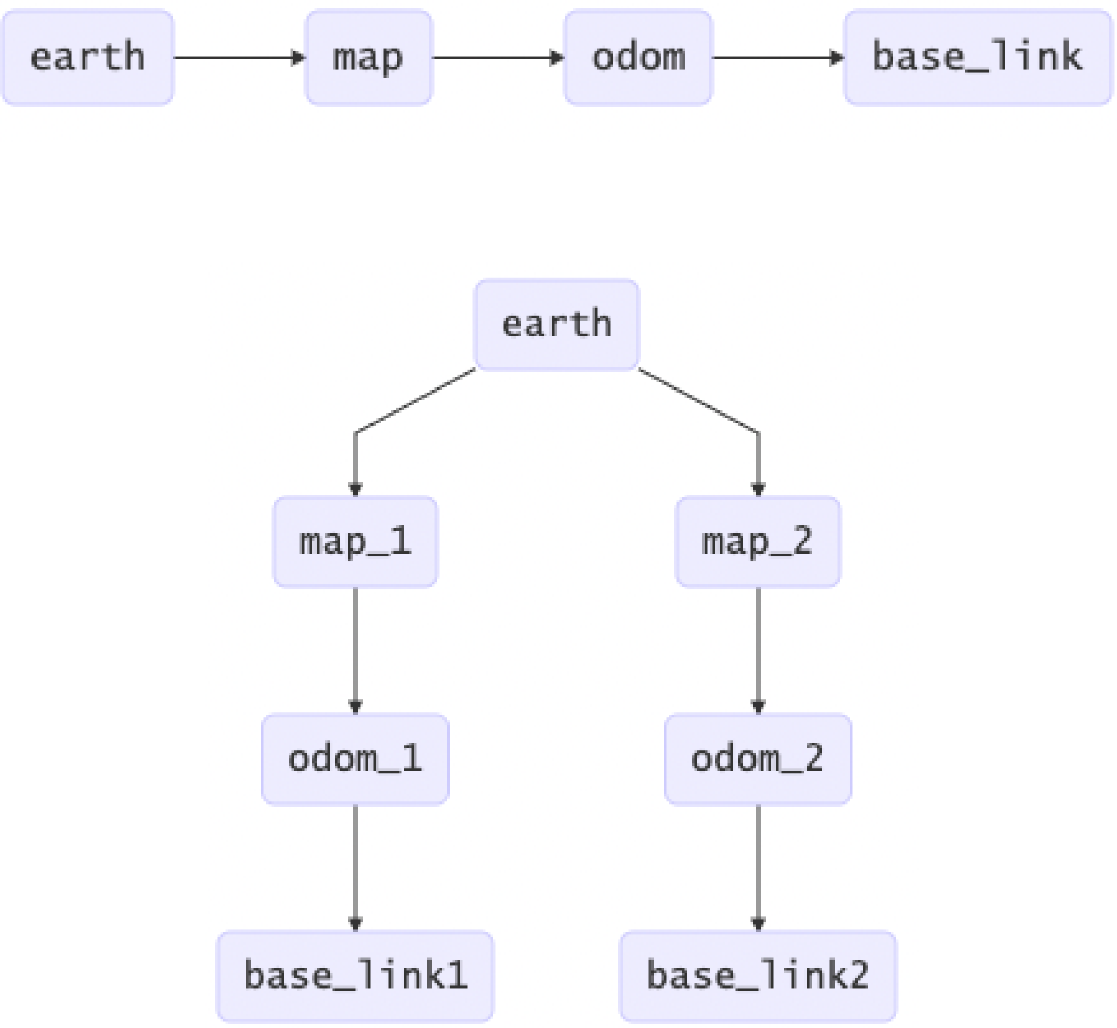
$$\begin{pmatrix} x_B \\ y_B \\ z_B \\ 1 \end{pmatrix} = \begin{pmatrix} R^{xx}_{A\rightarrow B} & R^{xy}_{A\rightarrow B} & R^{xz}_{A\rightarrow B} & T^{x}_{A\rightarrow B} \\ R^{yx}_{A\rightarrow B} & R^{yy}_{A\rightarrow B} & R^{yz}_{A\rightarrow B} & T^{y}_{A\rightarrow B} \\ R^{zx}_{A\rightarrow B} & R^{zy}_{A\rightarrow B} & R^{zz}_{A\rightarrow B} & T^{z}_{A\rightarrow B} \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x_A \\ y_A \\ z_A \\ 1 \end{pmatrix}$$

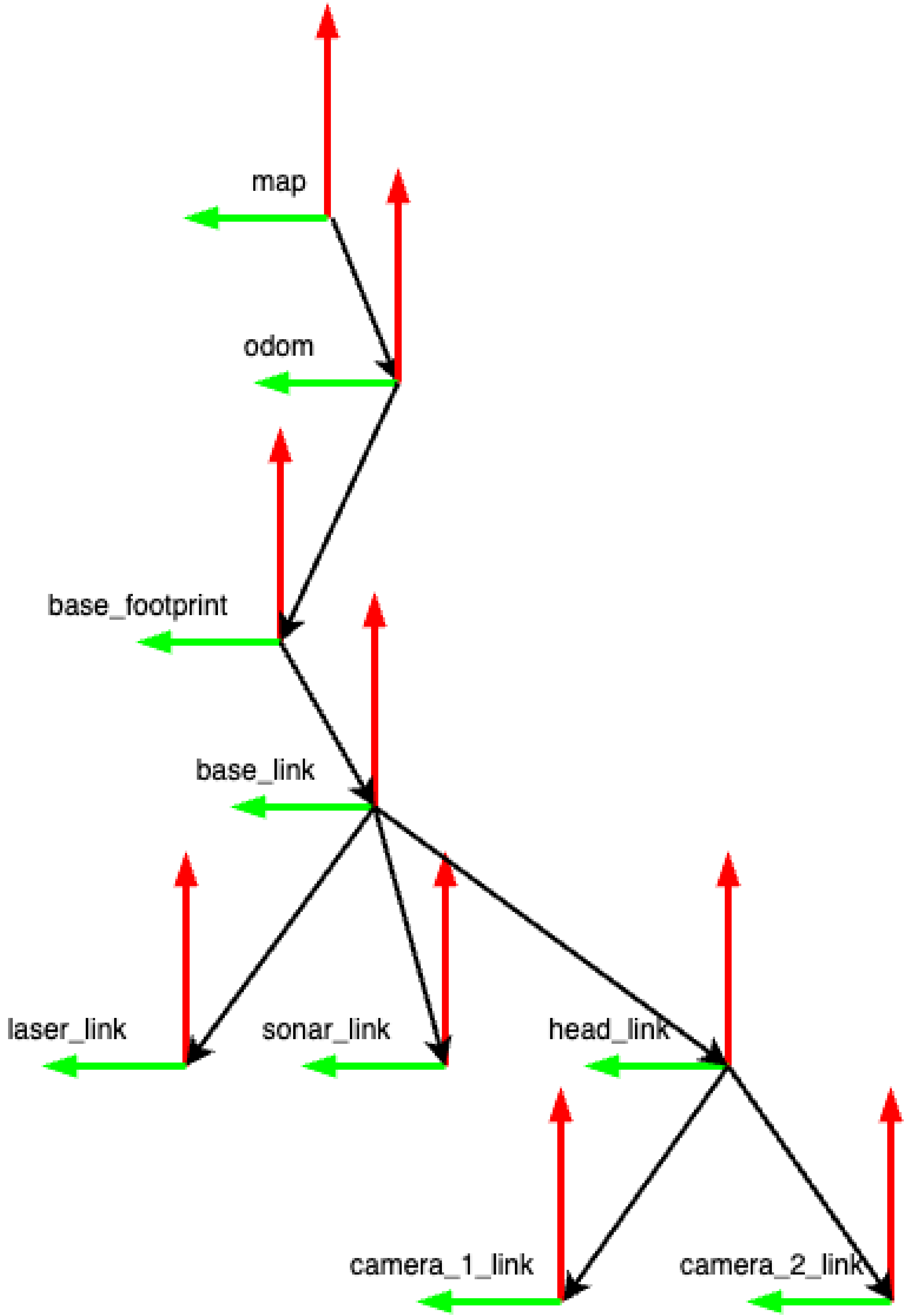
Intelligent Robotics Lab

Transforms Tree

- Transforms compose a Tree (1 parent / N children) with nodes linked by transforms
- Frame conventions specified in REP 105



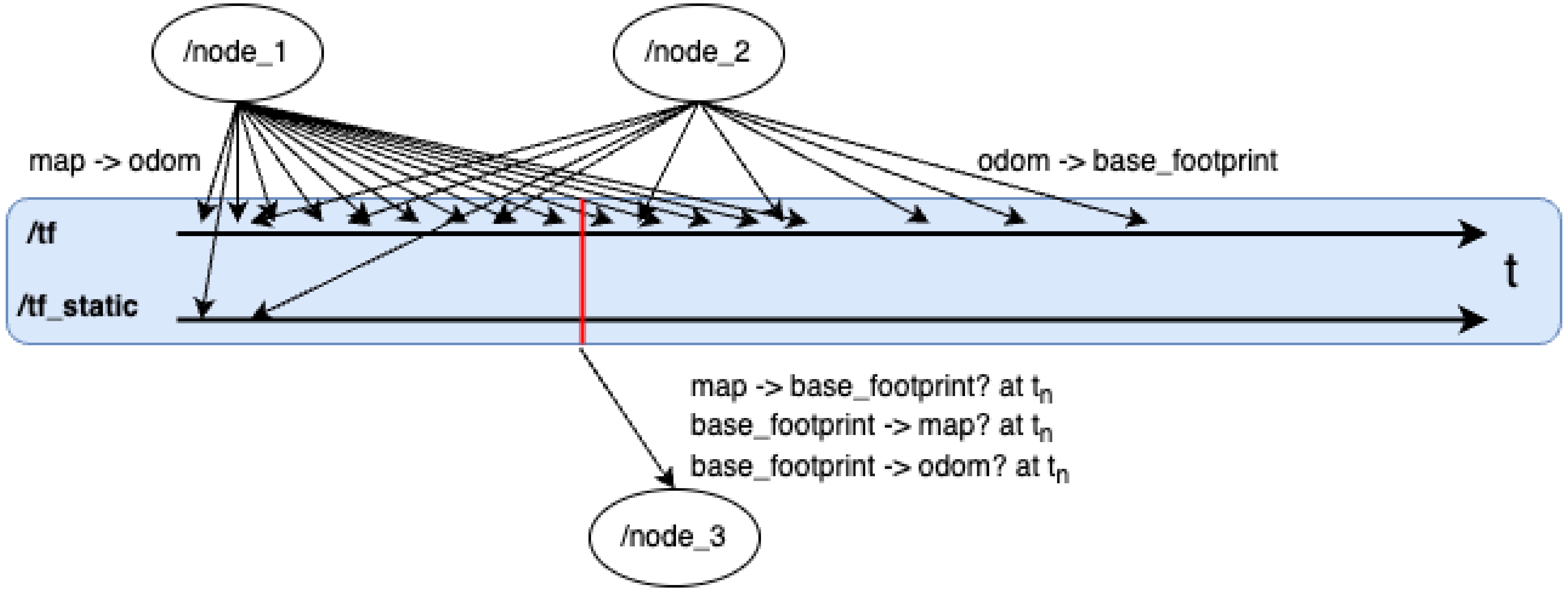


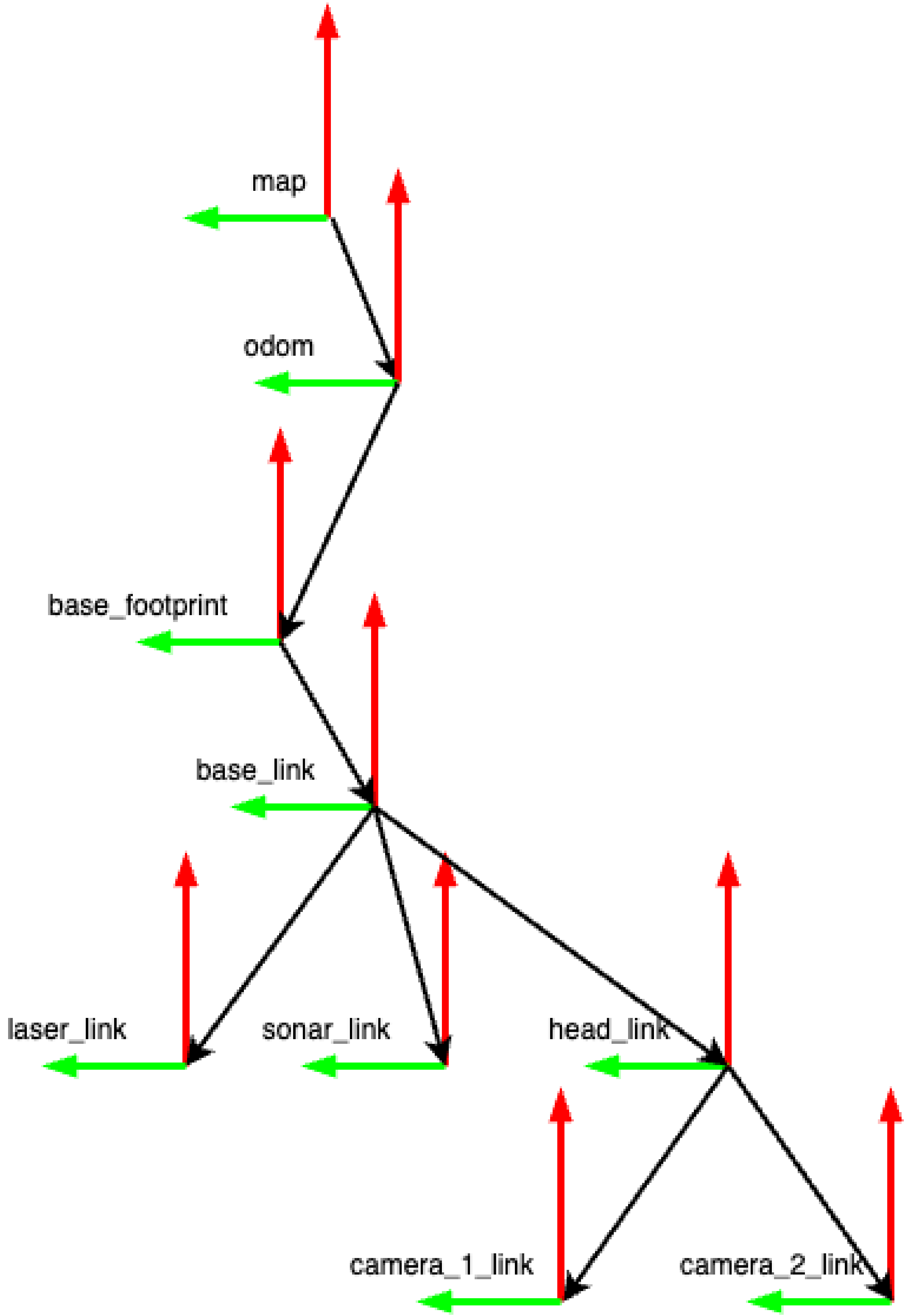


Transforms Tree

- TF System lets nodes to publish transforms:
- Dynamic (robot movement, joint, motor…)
- Static (robot geometry)

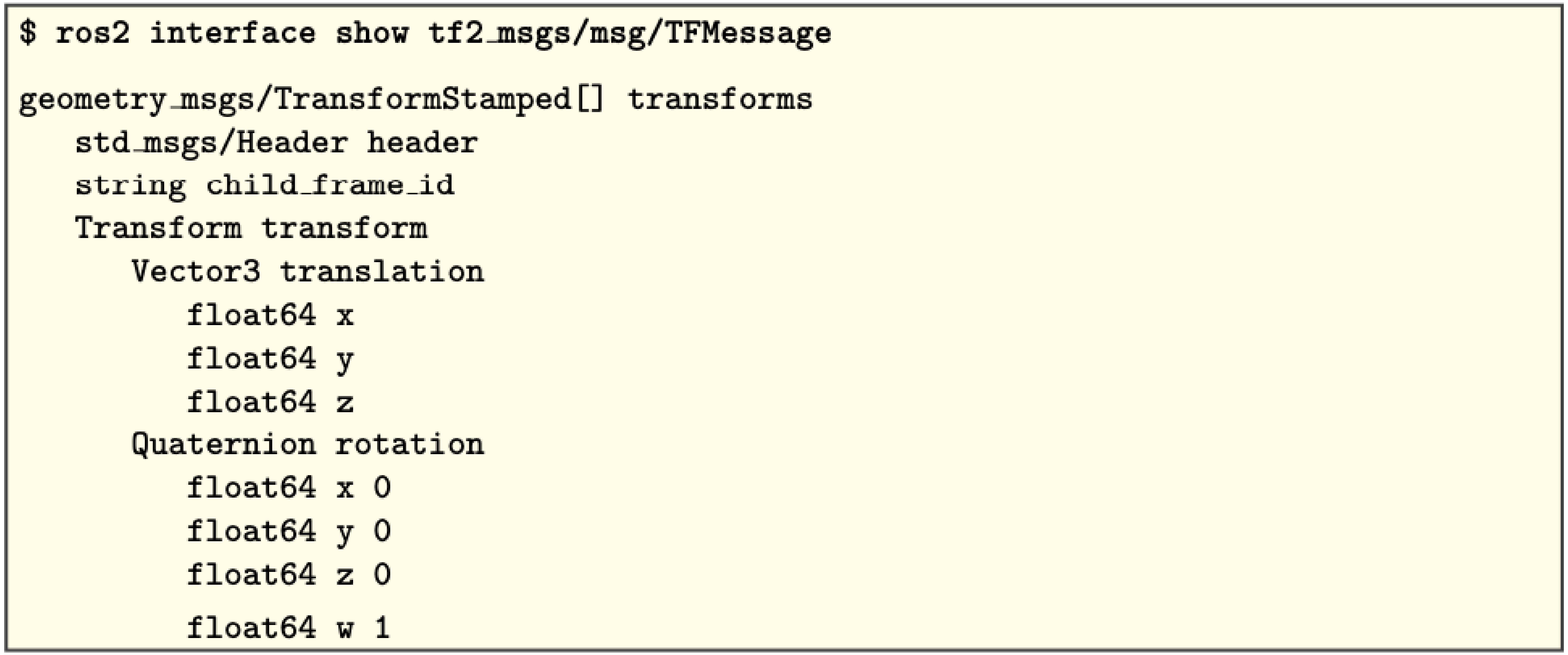
- TF System lets nodes to query for **any** transform at **any** time (except in the future)


/node_1
/node_2
map -> odom
odom -> base_footprint
/tf
/tf_static
t
map -> base_footprint? at t_n
base_footprint -> map? at t_n
base_footprint -> odom? at t_n
/node_3



Transforms interface

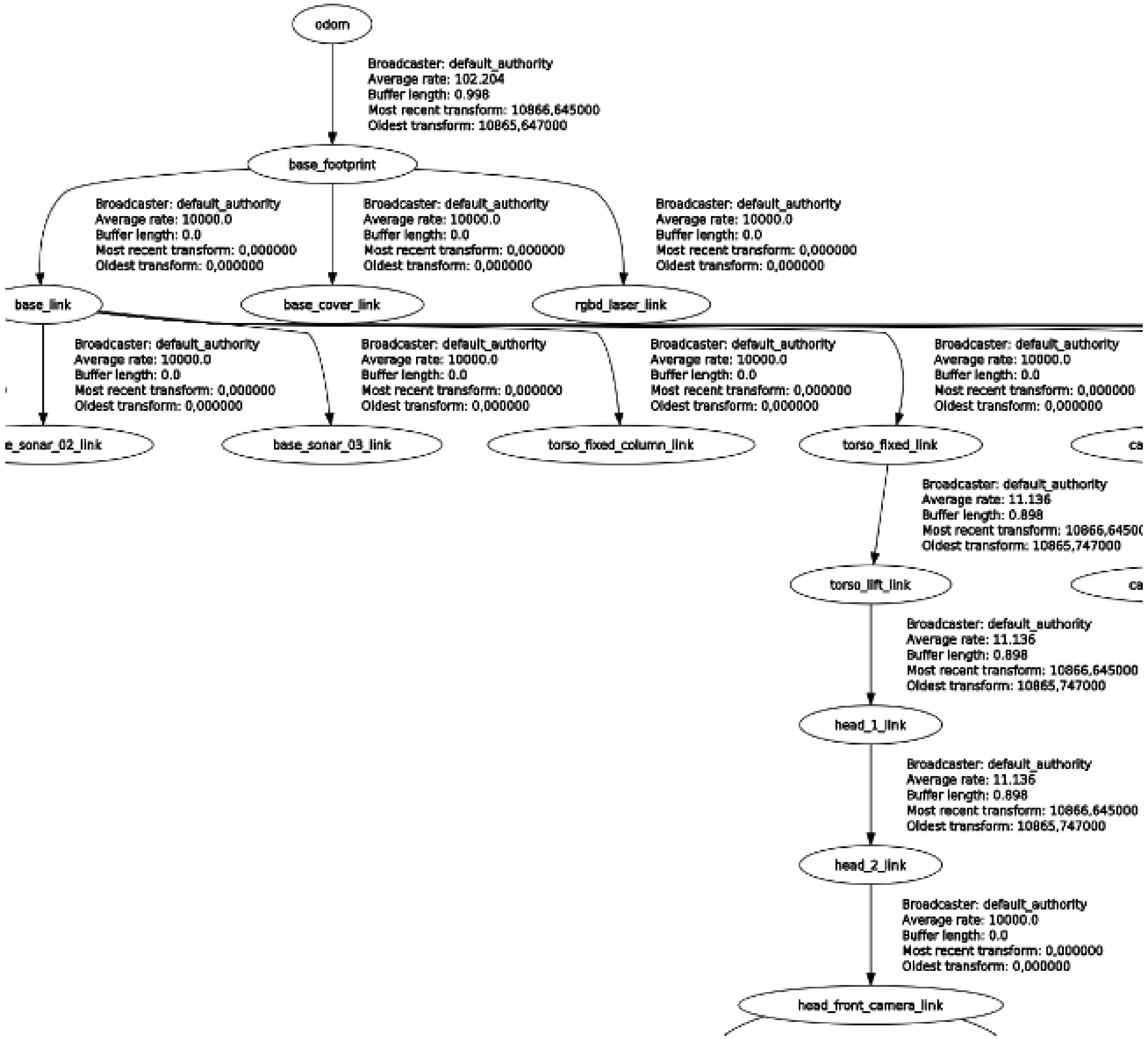
- Topics `/tf` and `/tf_static` (`tf2 msgs/msg/TFMessage`)

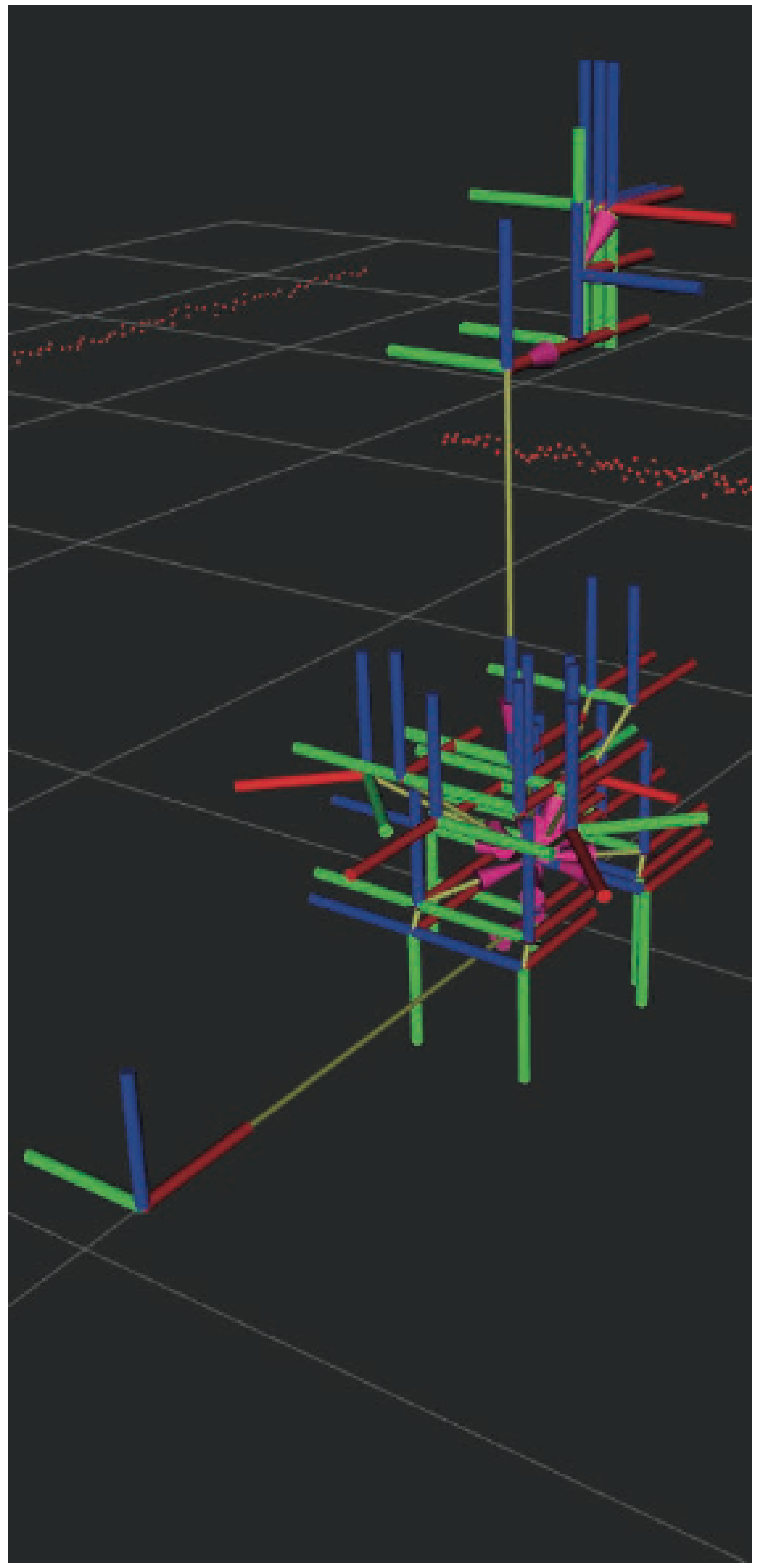


```
$ ros2 interface show tf2_msgs/msg/TFMessage

geometry_msgs/TransformStamped[] transforms
   std_msgs/Header header
   string child_frame_id
   Transform transform
      Vector3 translation
         float64 x
         float64 y
         float64 z
      Quaternion rotation
         float64 x 0
         float64 y 0
         float64 z 0
         float64 w 1
```

Visualizing Transforms

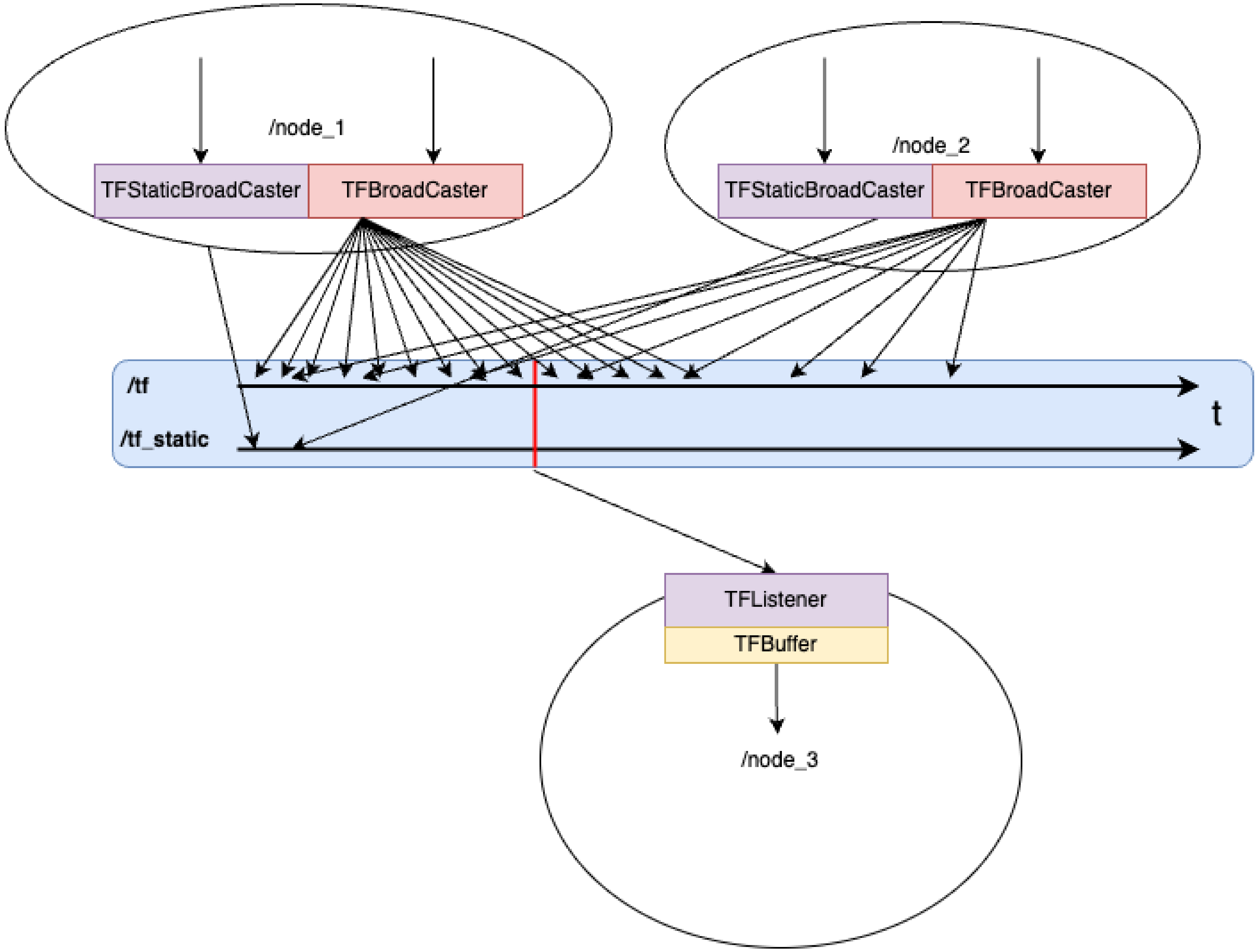
```
$ ros2 run rqt_tf_tree rqt_tf_tree
```


odom
Broadcaster: default_authority
Average rate: 102.204
Buffer length: 0.998
Most recent transform: 10866,645000
Oldest transform: 10865,647000
base_footprint
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
base_link
base_cover_link
rgbd_laser_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
e_sonar_02_link
base_sonar_03_link
torso_fixed_column_link
torso_fixed_link
ca
Broadcaster: default_authority
Average rate: 11.136
Buffer length: 0.898
Most recent transform: 10866,6450
Oldest transform: 10865,747000
torso_lift_link
ca
Broadcaster: default_authority
Average rate: 11.136
Buffer length: 0.898
Most recent transform: 10866,645000
Oldest transform: 10865,747000
head_1_link
Broadcaster: default_authority
Average rate: 11.136
Buffer length: 0.898
Most recent transform: 10866,645000
Oldest transform: 10865,747000
head_2_link
Broadcaster: default_authority
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0,000000
Oldest transform: 0,000000
head_front_camera_link



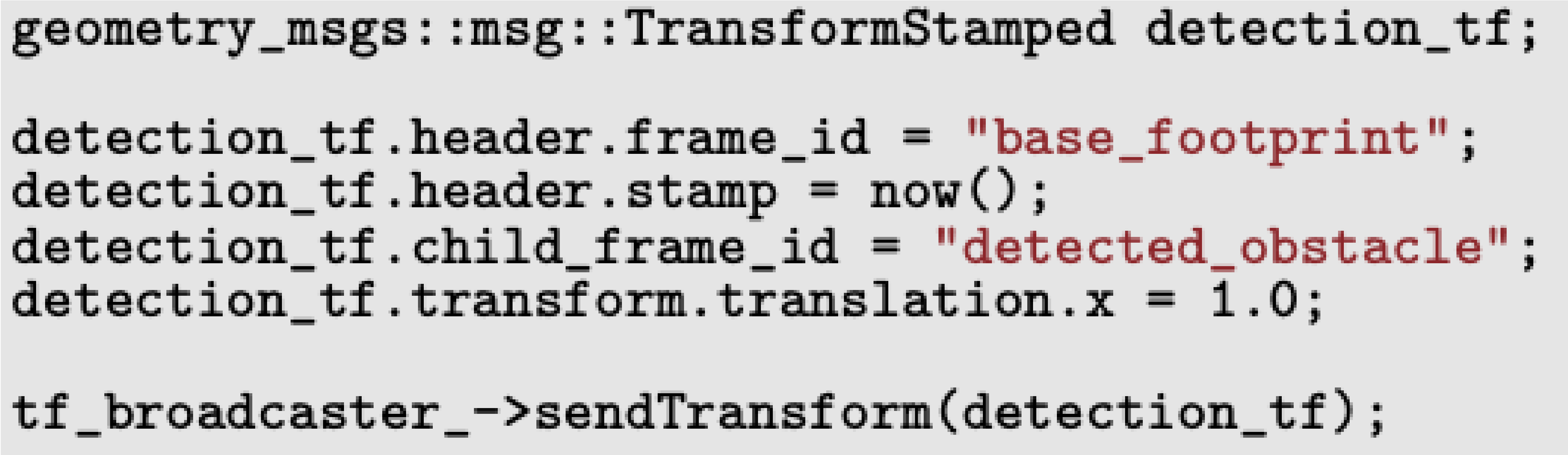
Programming with TFs

TF Listeners and Publishers

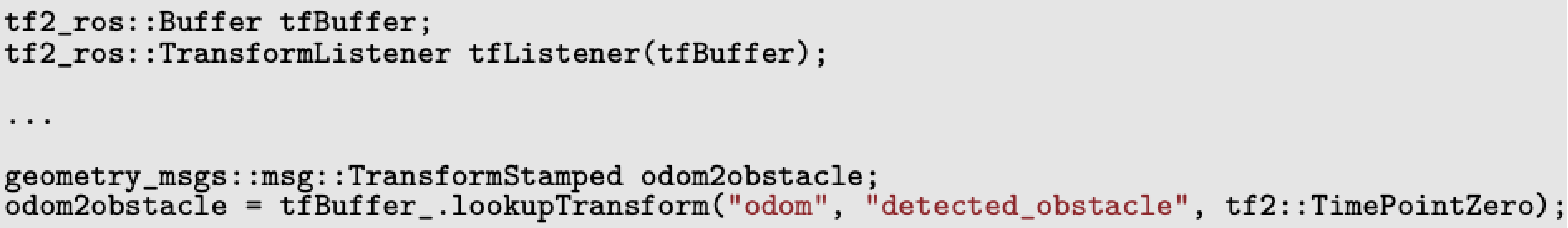


Programming with TFs

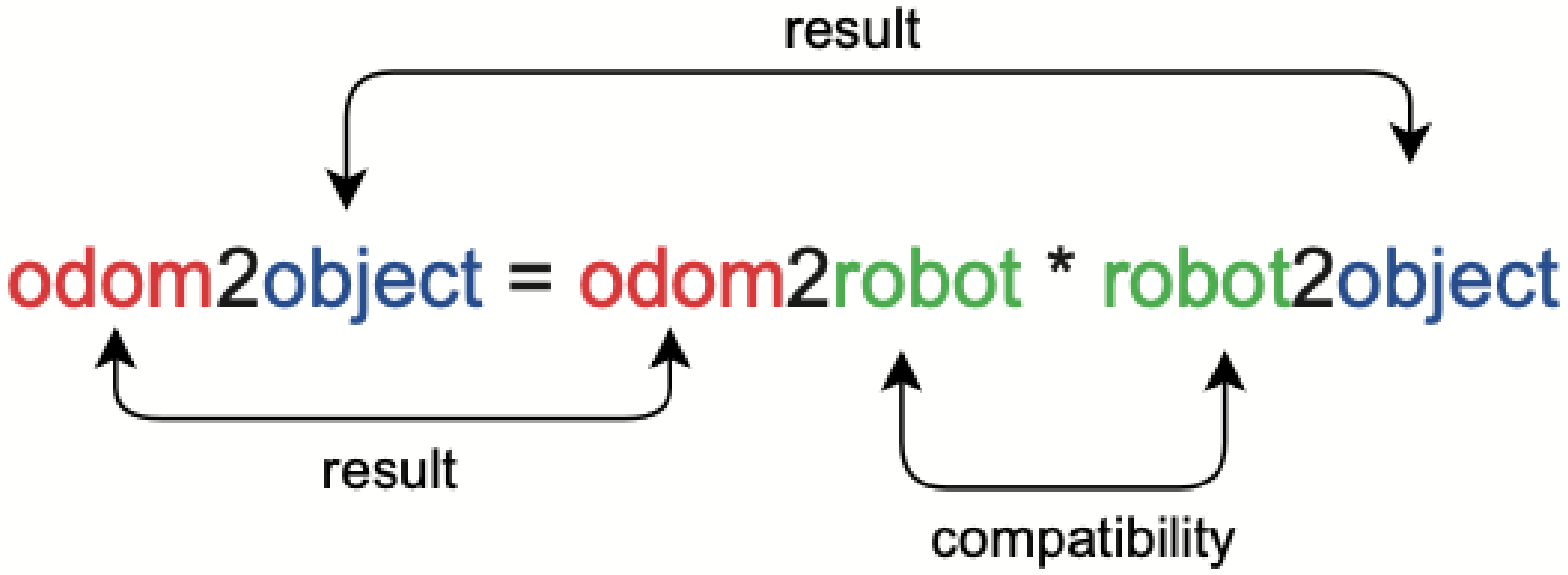
TF Listeners and Publishers



```
geometry_msgs::msg::TransformStamped detection_tf;

detection_tf.header.frame_id = "base_footprint";
detection_tf.header.stamp = now();
detection_tf.child_frame_id = "detected_obstacle";
detection_tf.transform.translation.x = 1.0;

tf_broadcaster_->sendTransform(detection_tf);
```

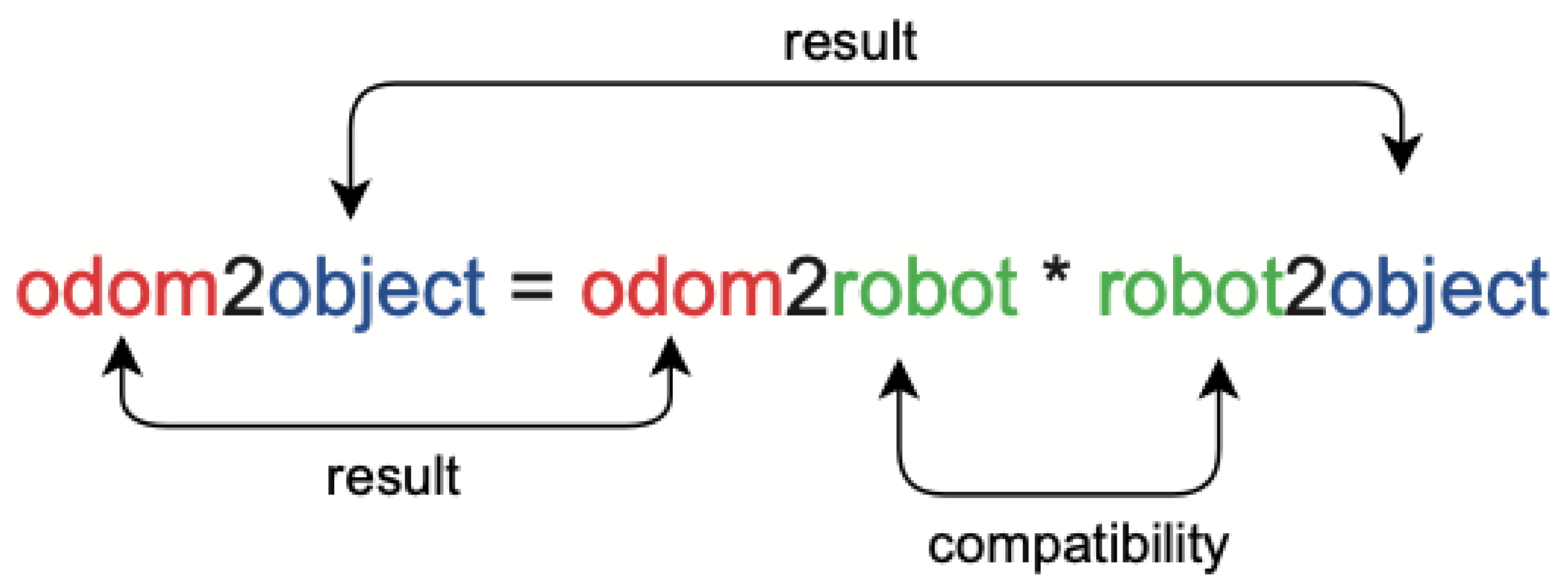


```
tf2_ros::Buffer tfBuffer;
tf2_ros::TransformListener tfListener(tfBuffer);

...

geometry_msgs::msg::TransformStamped odom2obstacle;
odom2obstacle = tfBuffer_.lookupTransform("odom", "detected_obstacle", tf2::TimePointZero);
```



Programming with TFs

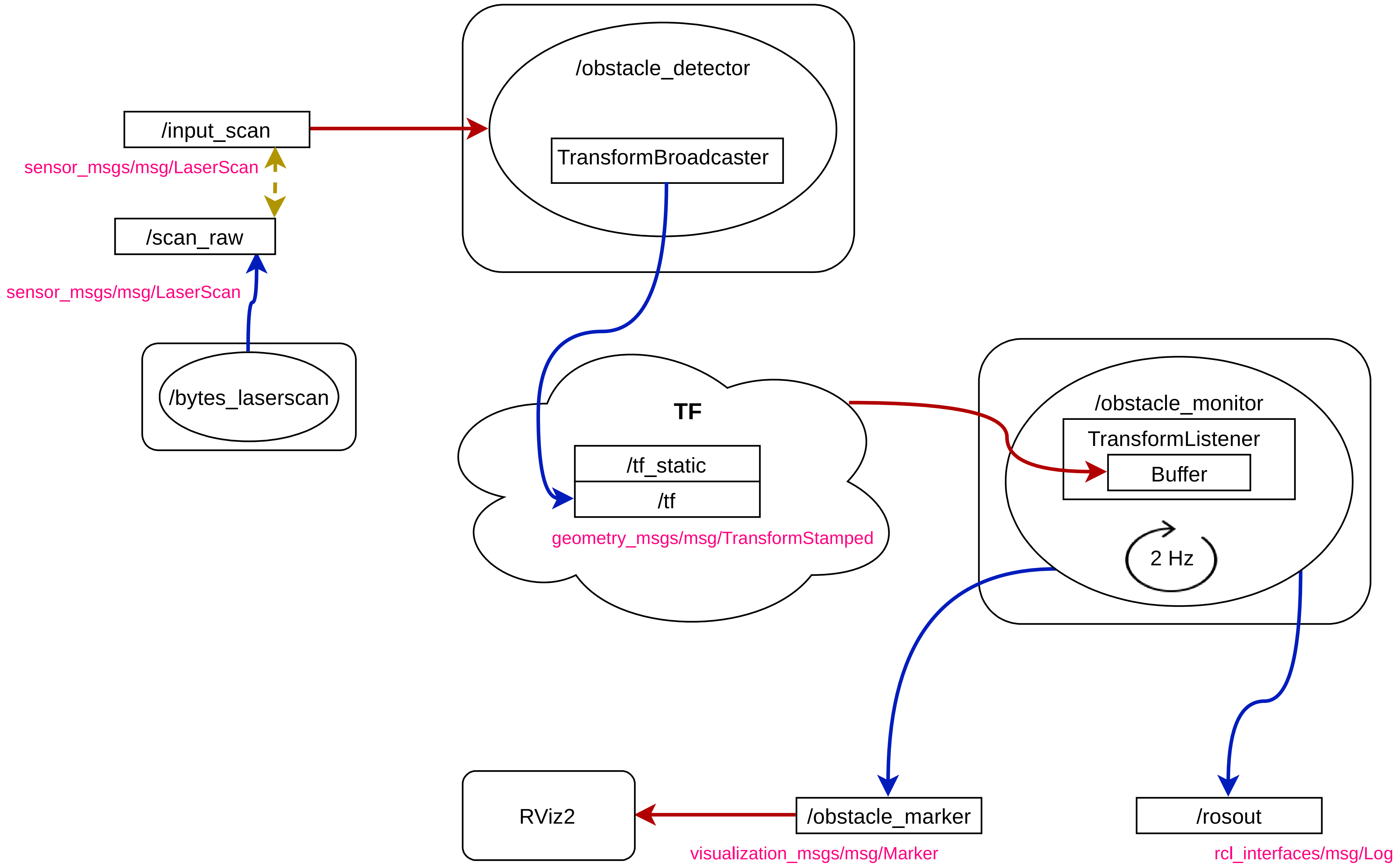
TF Listeners and Publishers

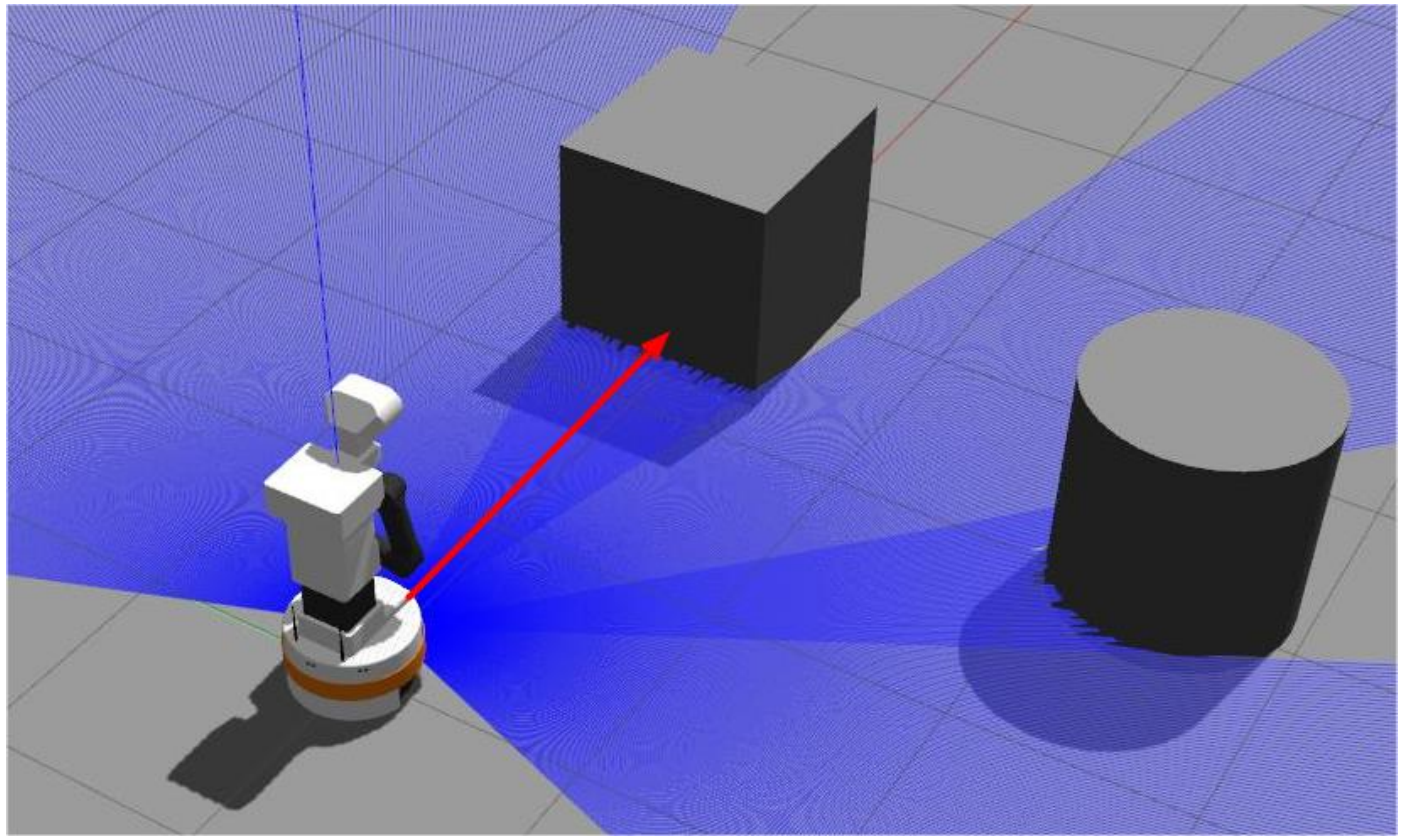


Programming with TFs

Problem Statement

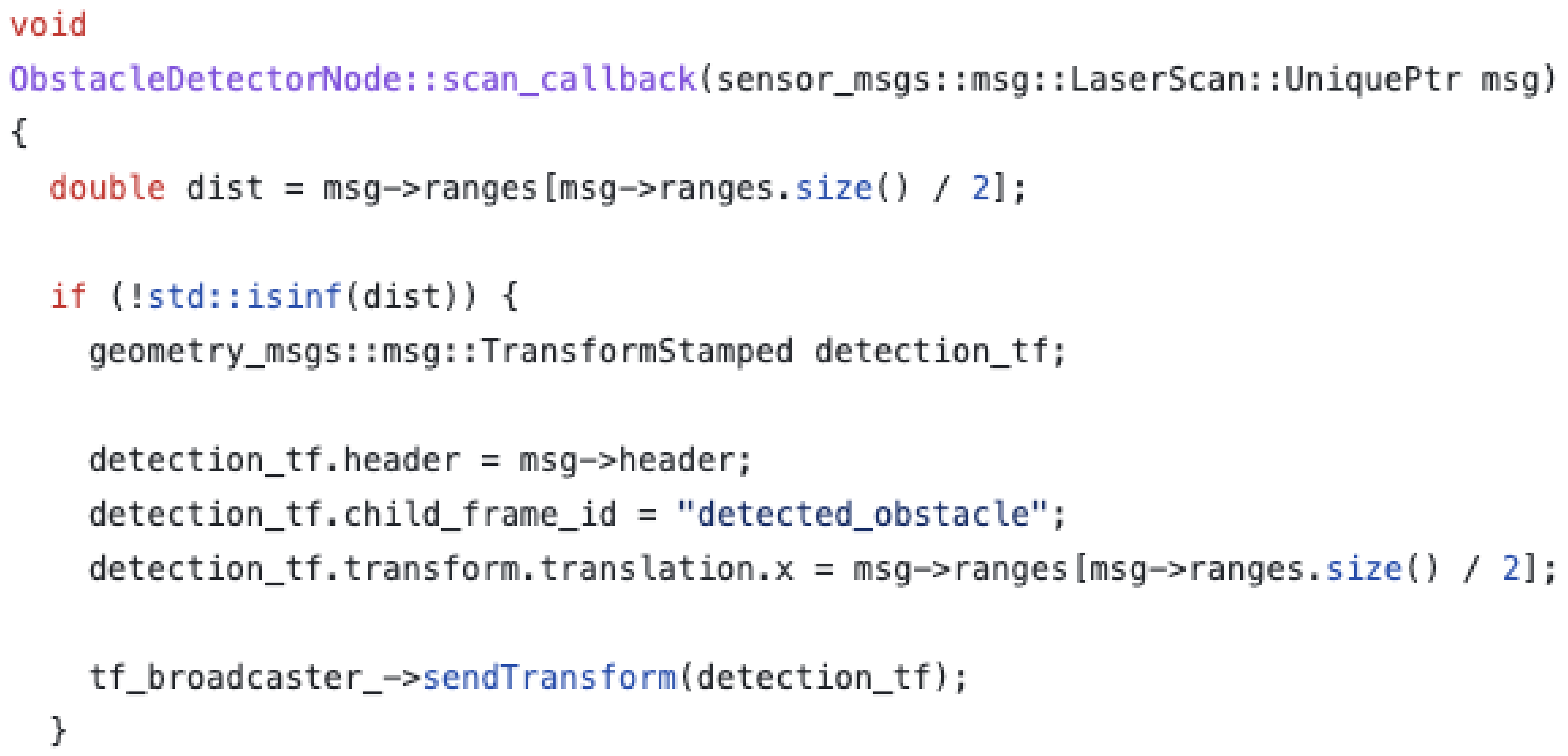
- A robot detects an obstacle while moving and stores it as a transform.
- At any time, it can reason where the last detected obstacle was



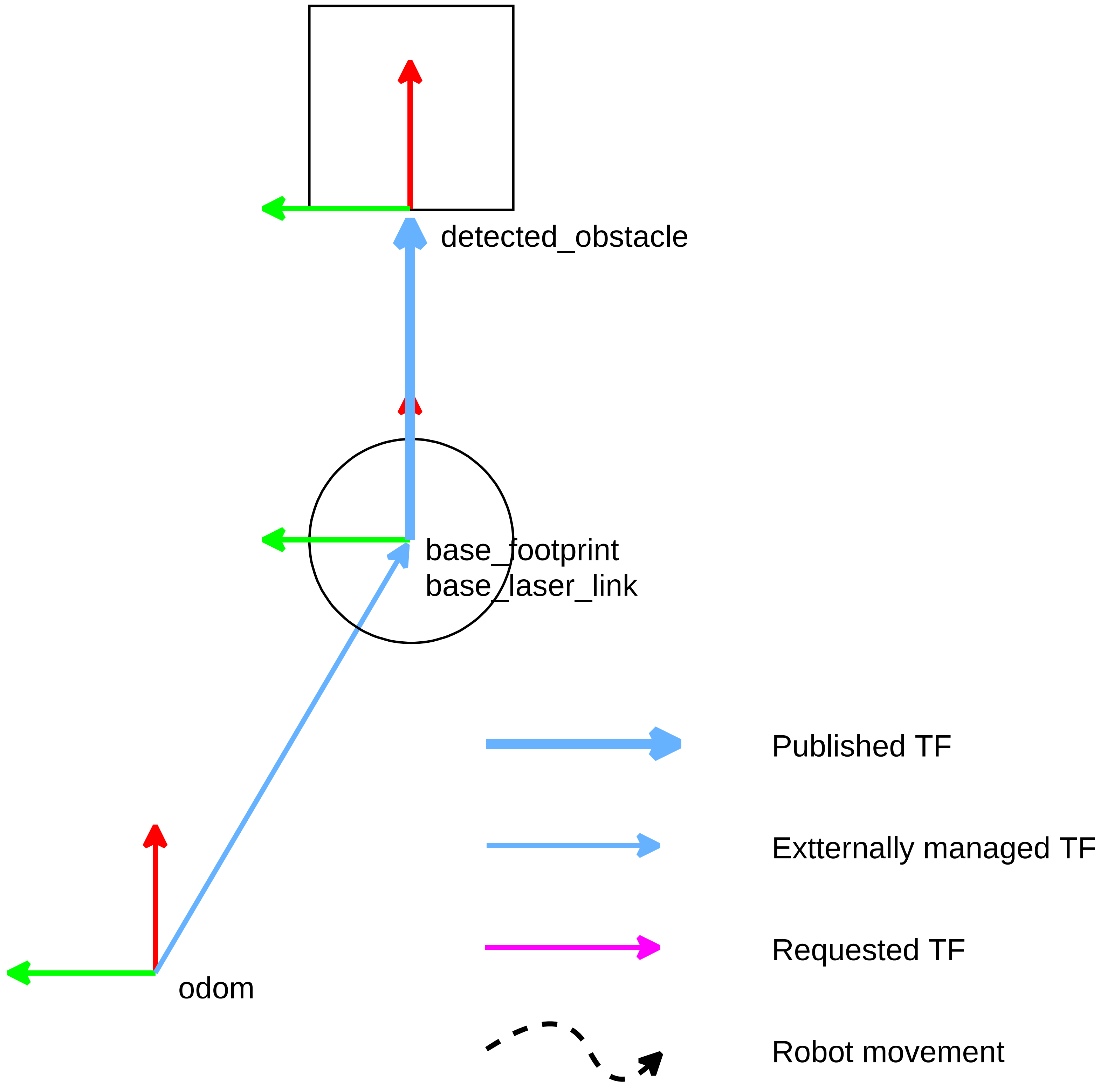


Programming with TFs

Obstacle Detection

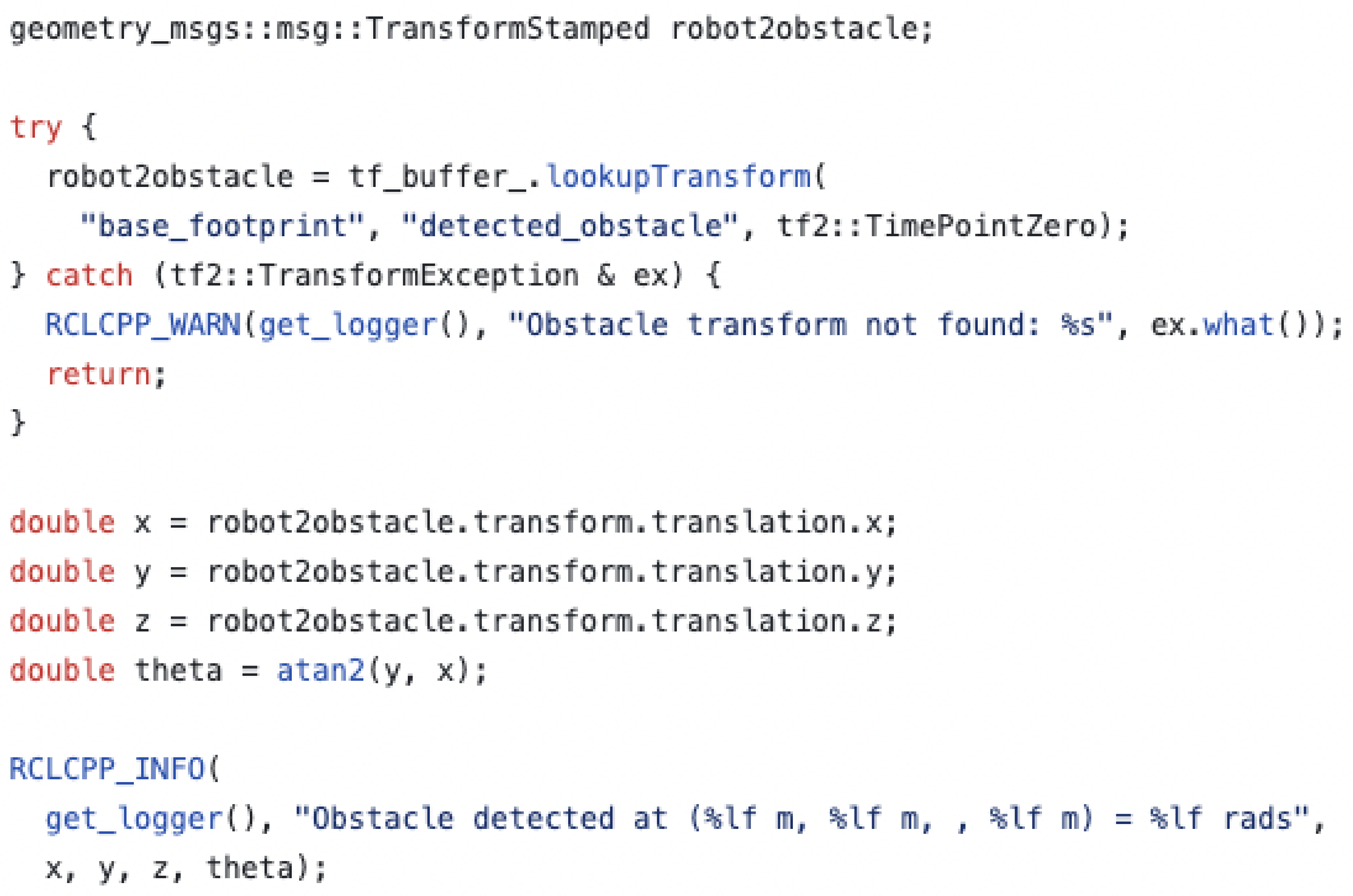


```
void
ObstacleDetectorNode::scan_callback(sensor_msgs::msg::LaserScan::UniquePtr msg)
{
  double dist = msg->ranges[msg->ranges.size() / 2];

  if (!std::isinf(dist)) {
    geometry_msgs::msg::TransformStamped detection_tf;

    detection_tf.header = msg->header;
    detection_tf.child_frame_id = "detected_obstacle";
    detection_tf.transform.translation.x = msg->ranges[msg->ranges.size() / 2];

    tf_broadcaster_->sendTransform(detection_tf);
  }
```

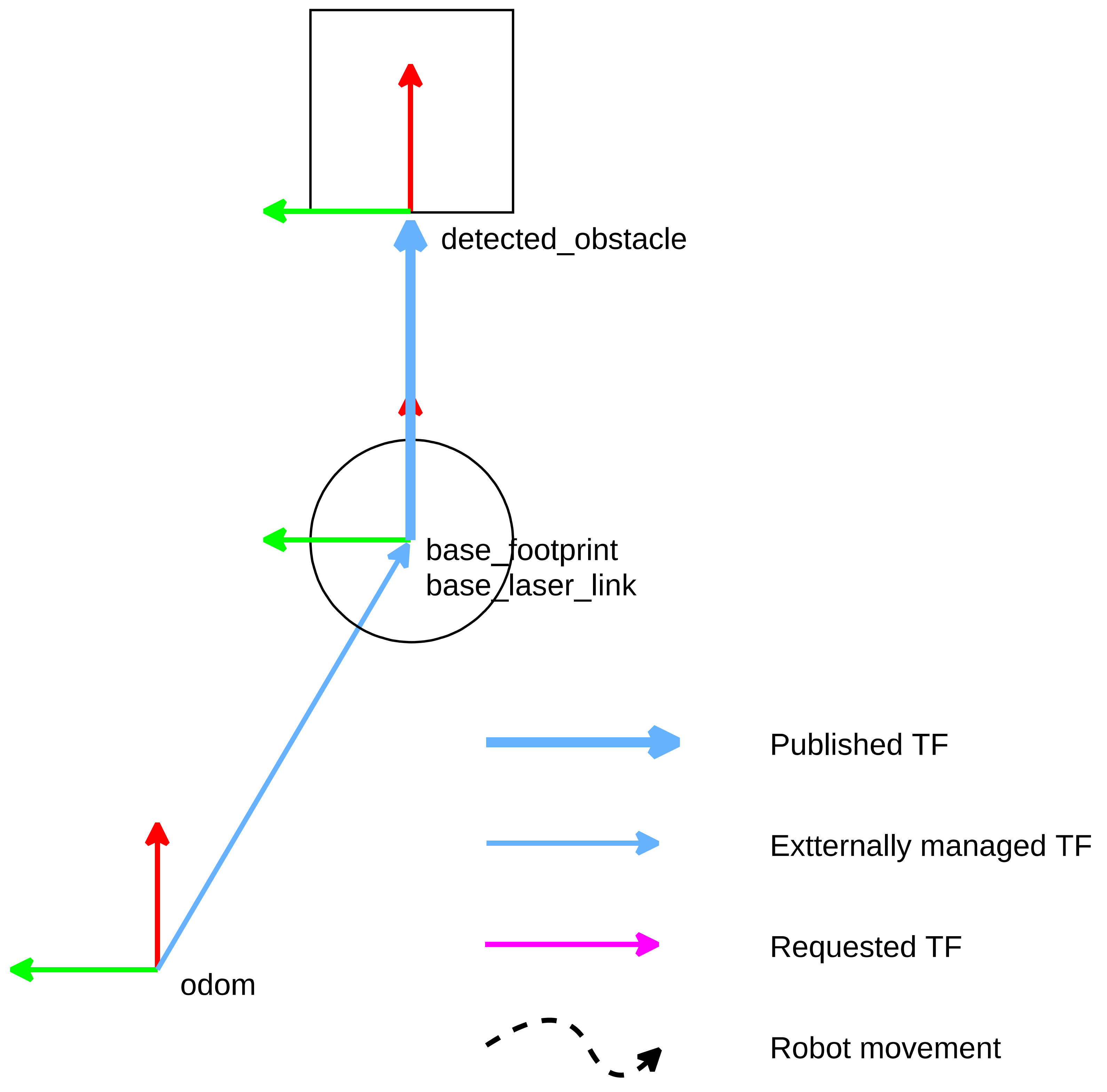


Programming with TFs

ObstAcle Monitoring

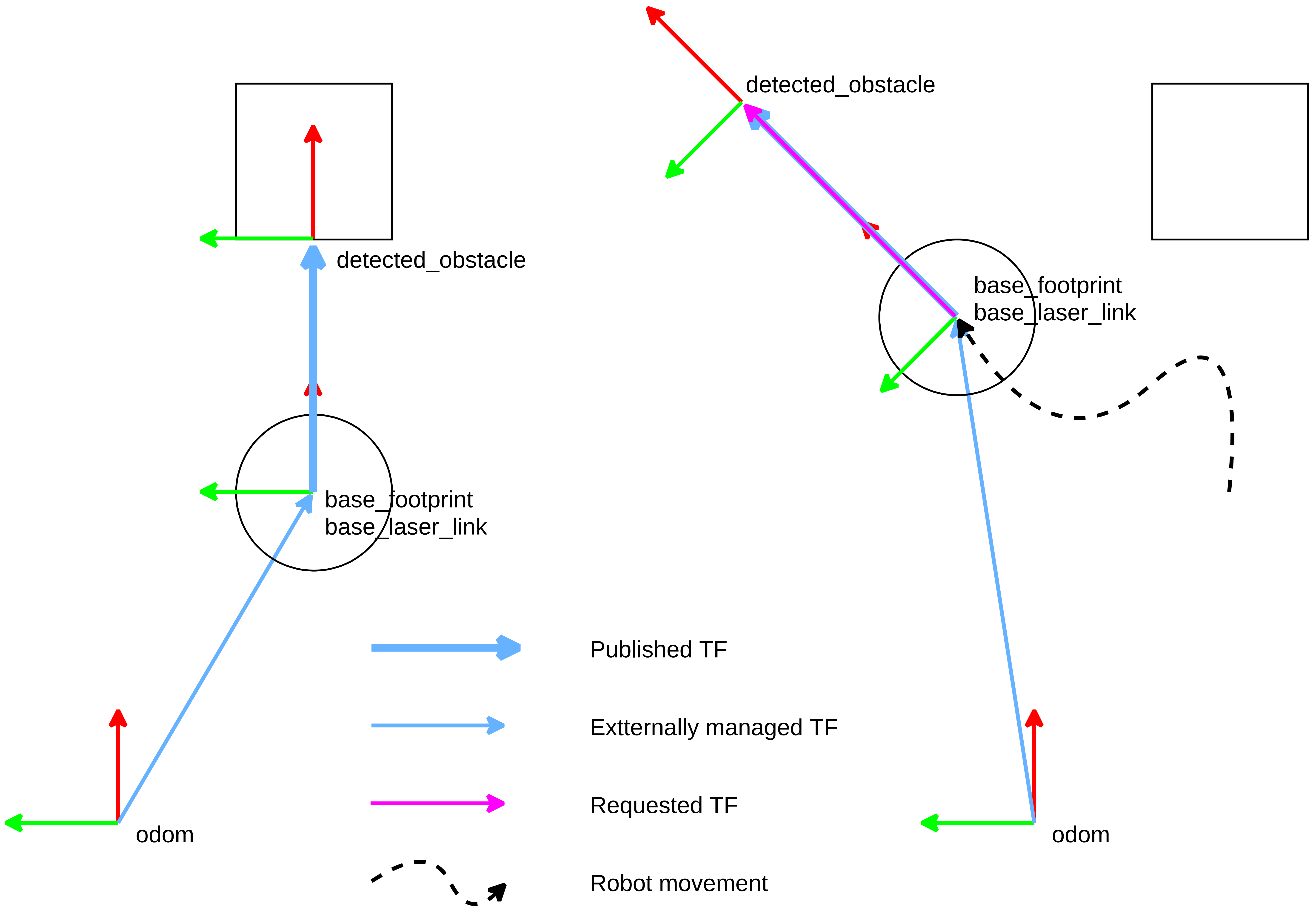


```
geometry_msgs::msg::TransformStamped robot2obstacle;

try {
  robot2obstacle = tf_buffer_.lookupTransform(
    "base_footprint", "detected_obstacle", tf2::TimePointZero);
} catch (tf2::TransformException & ex) {
  RCLCPP_WARN(get_logger(), "Obstacle transform not found: %s", ex.what());
  return;
}

double x = robot2obstacle.transform.translation.x;
double y = robot2obstacle.transform.translation.y;
double z = robot2obstacle.transform.translation.z;
double theta = atan2(y, x);

RCLCPP_INFO(
  get_logger(), "Obstacle detected at (%lf m, %lf m, , %lf m) = %lf rads",
  x, y, z, theta);
```



Programming with TFs

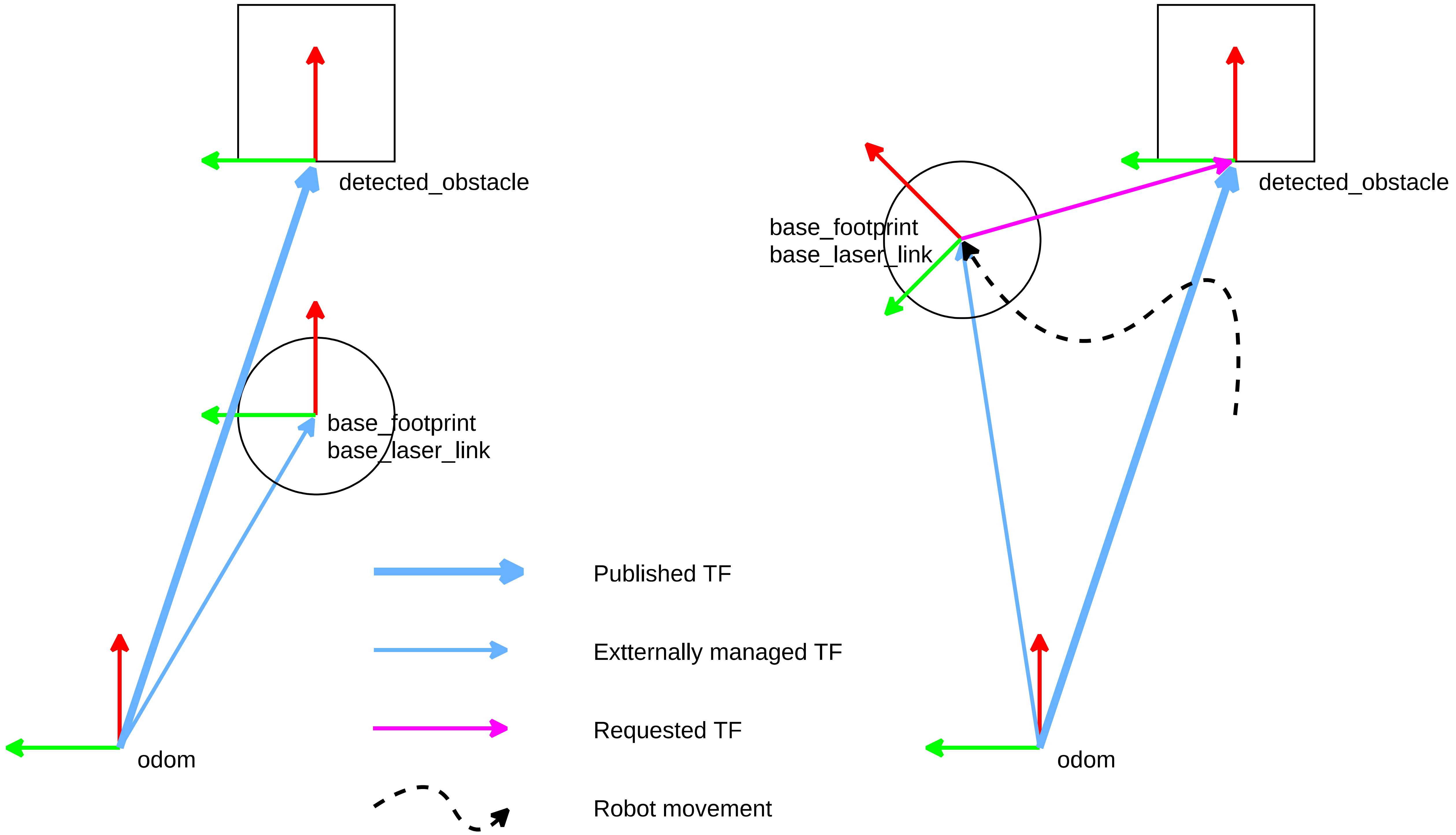
Yes, but..

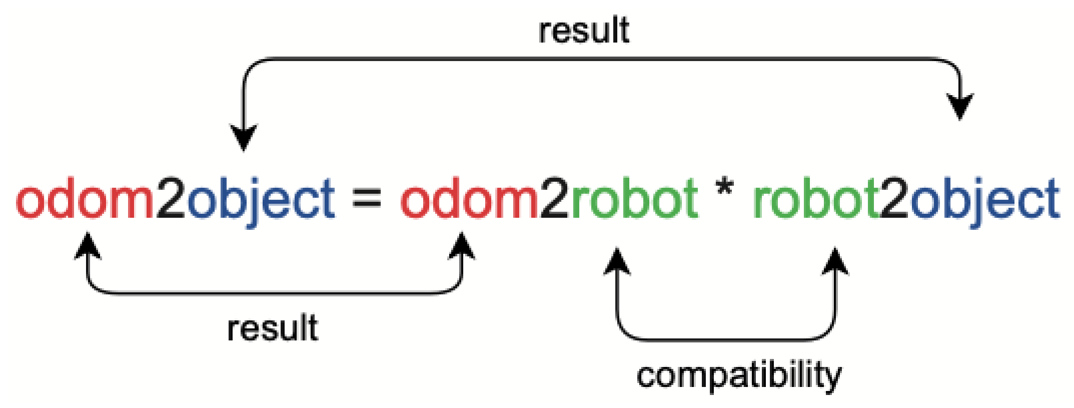


Programming with TFs

Improved Detection

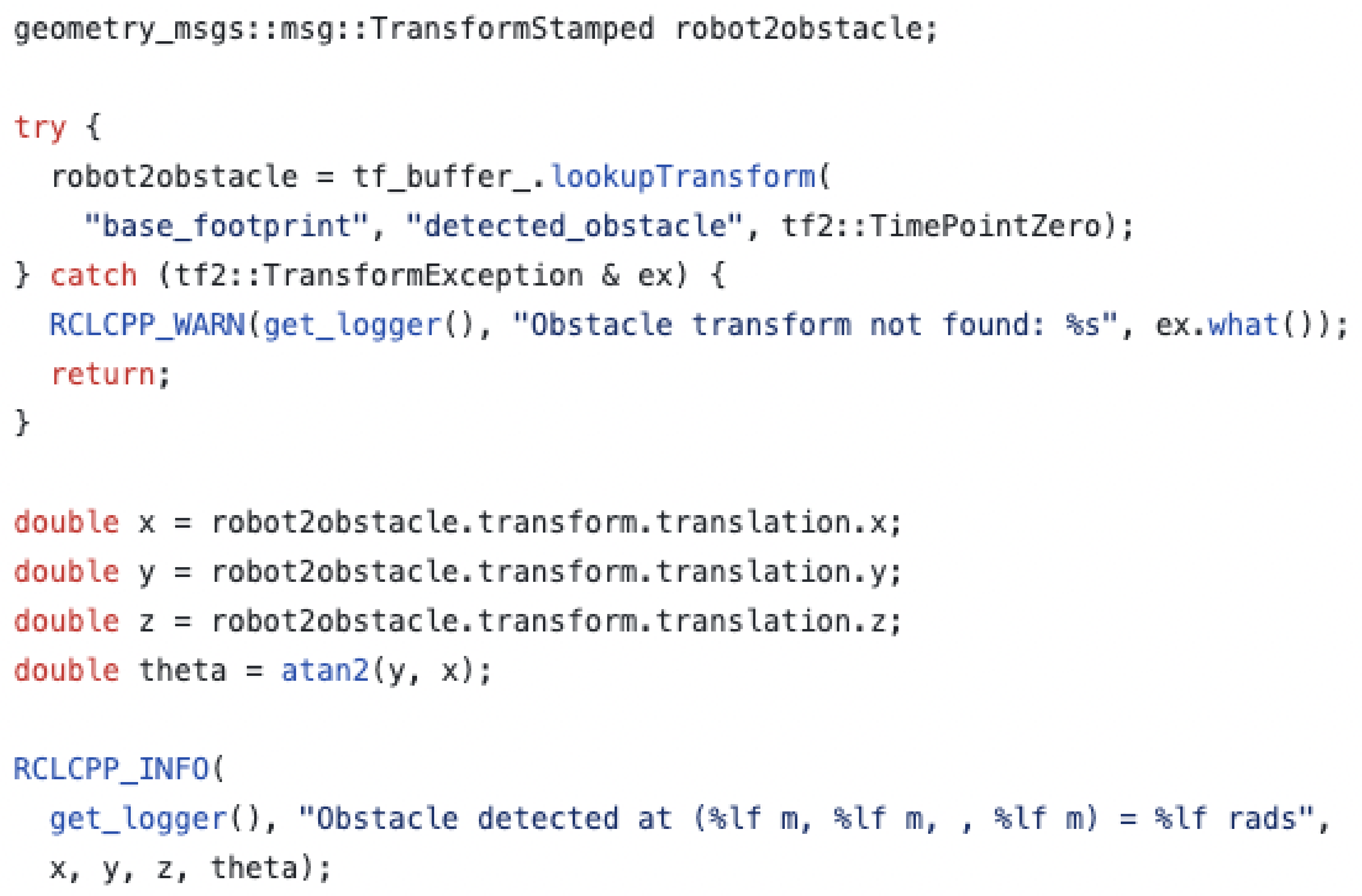
```
if (!std::isinf(dist)) {
  tf2::Transform laser2object;
  laser2object.setOrigin(tf2::Vector3(dist, 0.0, 0.0));
  laser2object.setRotation(tf2::Quaternion(0.0, 0.0, 0.0, 1.0));

  geometry_msgs::msg::TransformStamped odom2laser_msg;
  tf2::Stamped<tf2::Transform> odom2laser;
  try {
    odom2laser_msg = tf_buffer_.lookupTransform(
      "odom", "base_laser_link",
      tf2::timeFromSec(rclcpp::Time(msg->header.stamp).seconds() - 0.3));
    tf2::fromMsg(odom2laser_msg, odom2laser);
  } catch (tf2::TransformException & ex) {
    RCLCPP_WARN(get_logger(), "Obstacle transform not found: %s", ex.what());
    return;
  }

  tf2::Transform odom2object = odom2laser * laser2object;

  geometry_msgs::msg::TransformStamped odom2object_msg;
  odom2object_msg.transform = tf2::toMsg(odom2object);

  odom2object_msg.header.stamp = msg->header.stamp;
  odom2object_msg.header.frame_id = "odom";
  odom2object_msg.child_frame_id = "detected_obstacle";

  tf_broadcaster_->sendTransform(odom2object_msg);
```



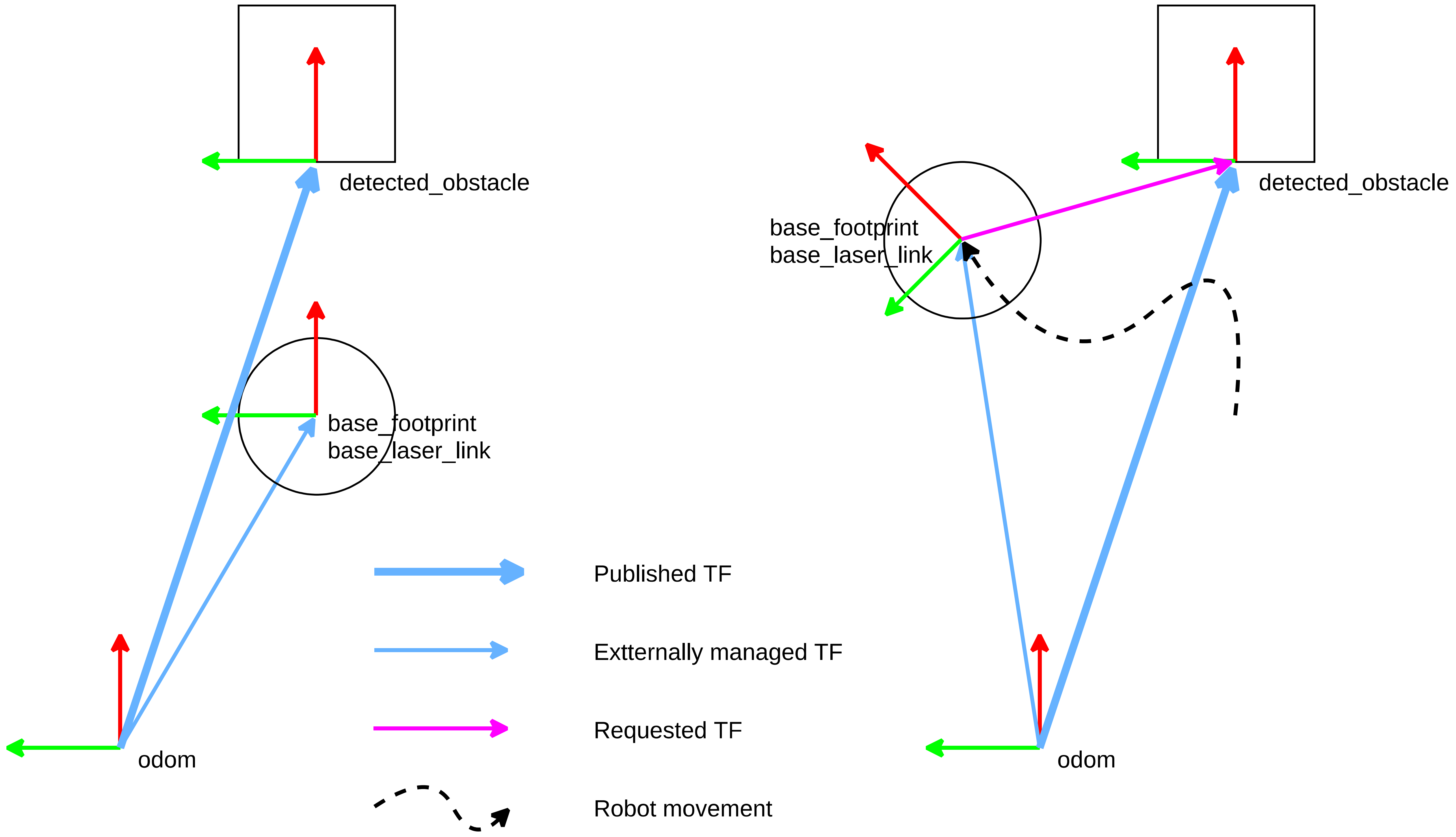


Programming with TFs

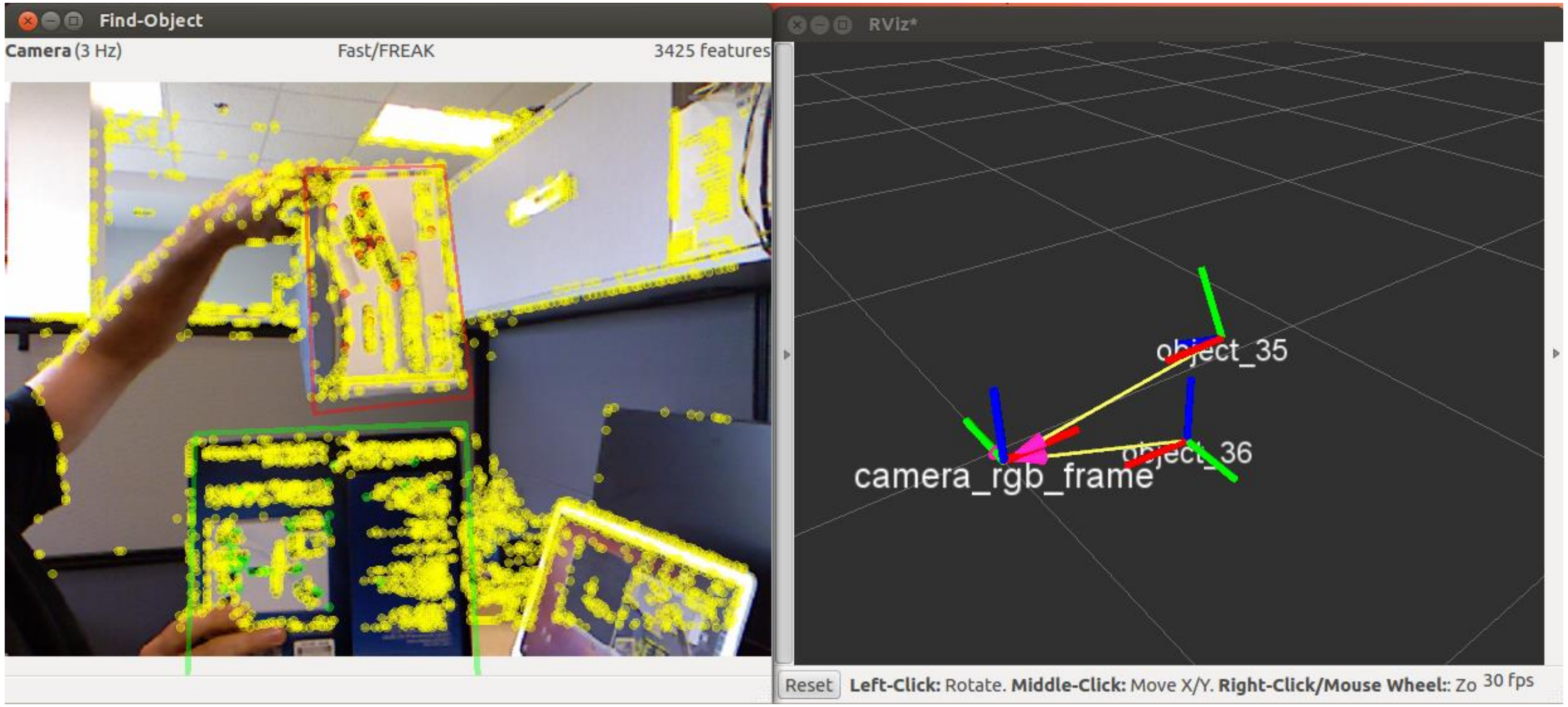
Obstacle Monitoring



```
geometry_msgs::msg::TransformStamped robot2obstacle;

try {
  robot2obstacle = tf_buffer_.lookupTransform(
    "base_footprint", "detected_obstacle", tf2::TimePointZero);
} catch (tf2::TransformException & ex) {
  RCLCPP_WARN(get_logger(), "Obstacle transform not found: %s", ex.what());
  return;
}

double x = robot2obstacle.transform.translation.x;
double y = robot2obstacle.transform.translation.y;
double z = robot2obstacle.transform.translation.z;
double theta = atan2(y, x);

RCLCPP_INFO(
  get_logger(), "Obstacle detected at (%lf m, %lf m, , %lf m) = %lf rads",
  x, y, z, theta);
```

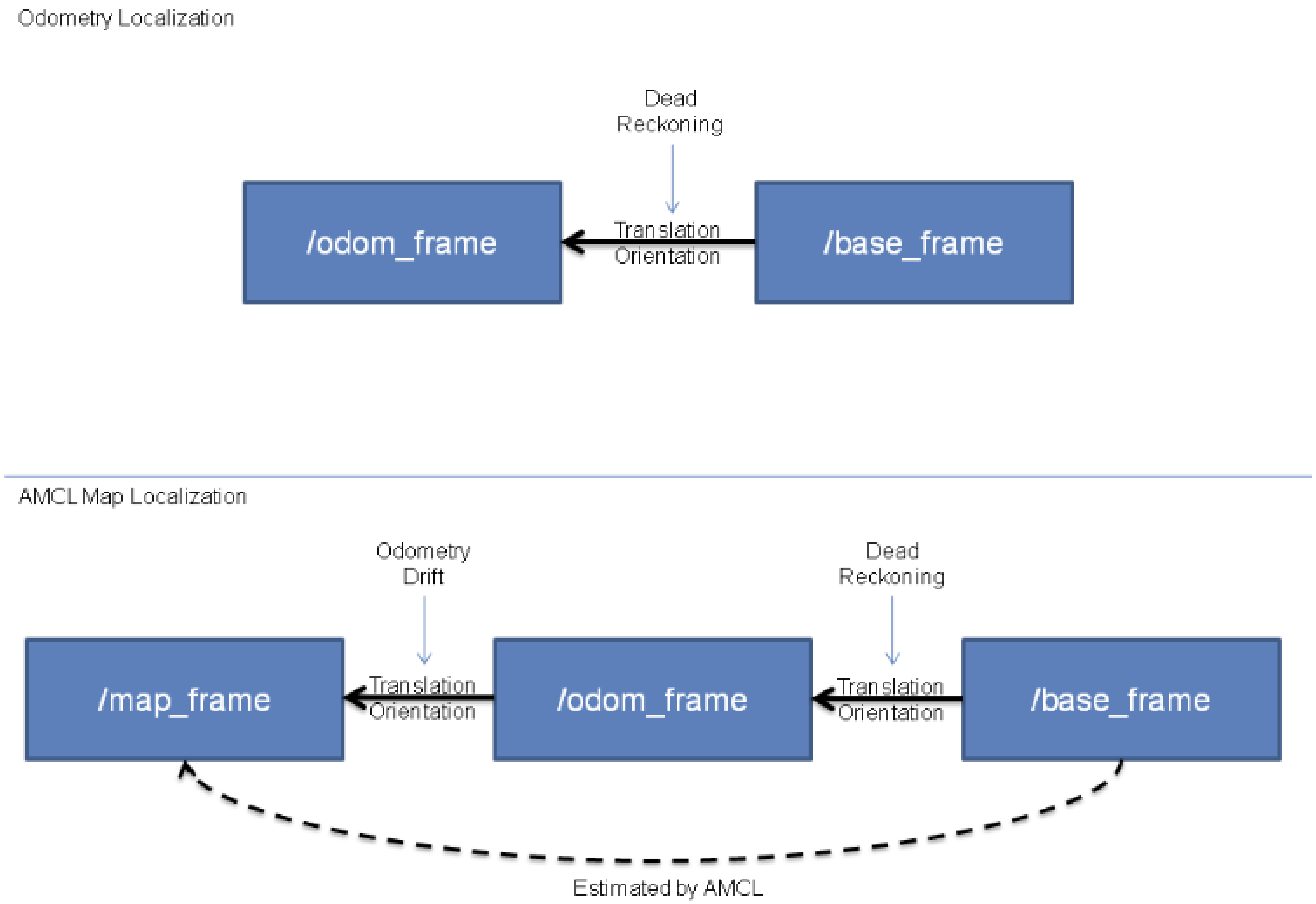


Examples of TF Applications



http://wiki.ros.org/find_object_2d

Examples of TF Applications



Odometry Localization

Dead Reckoning

/odom_frame

Translation
Orientation

/base_frame

AMCL Map Localization

Odometry Drift

Dead Reckoning

/map_frame

Translation
Orientation

/odom_frame

Translation
Orientation

/base_frame

Estimated by AMCL

Examples of TF Applications