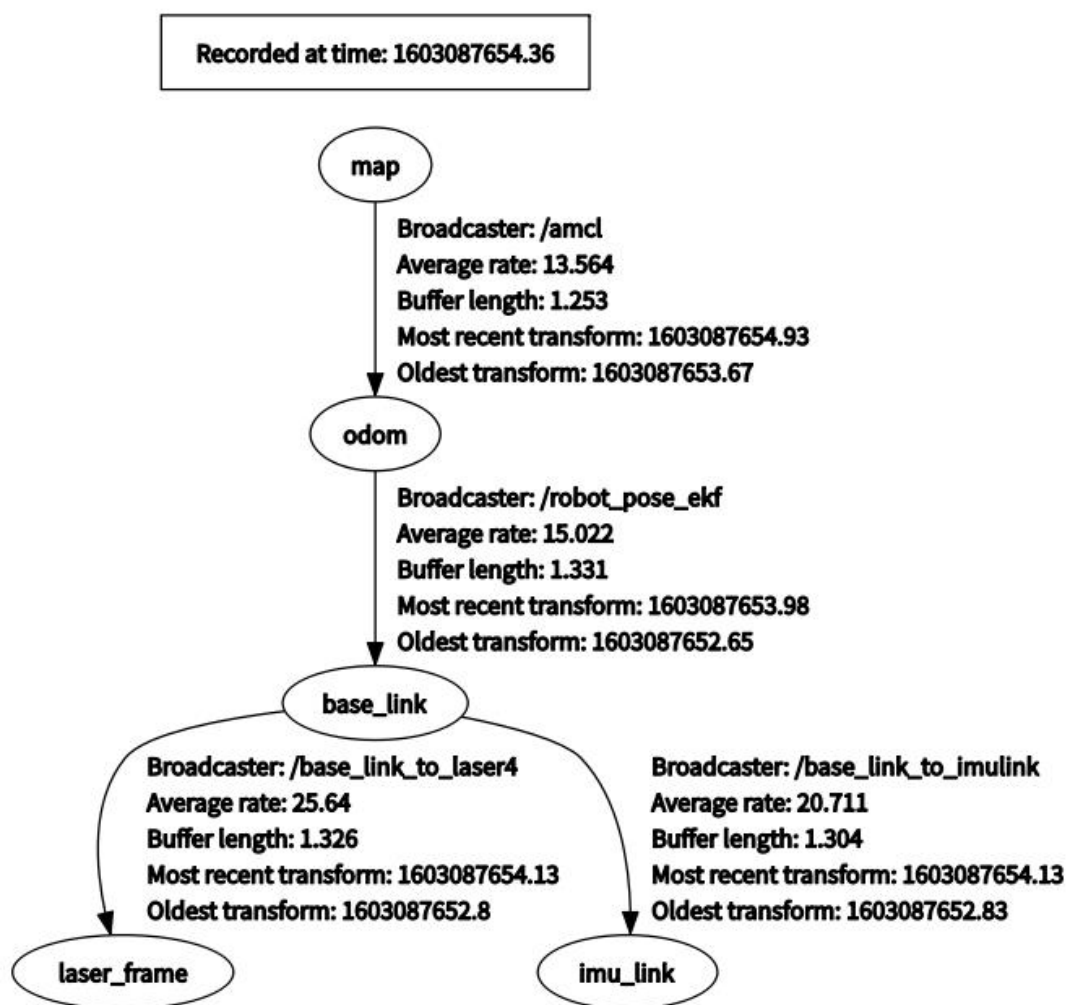


轻舟机器人 TF 变换讲解

AI 航团队

tf 是一个让用户随时跟踪多个参考系的功能包，它使用一种树型数据结构，根据时间缓冲并维护多个参考系之间的坐标变换关系，可以帮助用户在任意时间，将点、向量等数据的坐标，在两个参考系中完成坐标变换。

如下图所示，是轻舟机器人在运动过程中所产生的 TF 树结构。



轻舟机器人上的主要传感器为 IMU 模块和激光雷达，两个传感器相对于车体位置都是静态的，所以示例程序采用 static_transform_publisher 工具的功能是发布两个参考系之间的静态坐标变换，两个参考系一般不发生相对位置变化。命令的格式如下：

```
static_transform_publisher x y z yaw pitch roll frame_id child_frame_id period_in_ms
static_transform_publisher x y z qx qy qz qw frame_id child_frame_id period_in_ms
```

以上两种命令格式，需要设置坐标的偏移和旋转参数，偏移参数都使用相对于 xyz 三轴的坐标位移，而旋转参数第一种命令格式使用以弧度为单位的 yaw/pitch/roll 三个角度（yaw

是围绕 x 轴旋转的偏航角, $pitch$ 是围绕 y 轴旋转的俯仰角, $roll$ 是围绕 z 轴旋转的翻滚角), 而第二种命令格式使用四元数表达旋转角度。发布频率以 ms 为单位, 一般 $100ms$ 比较合适。该命令不仅可以在终端中使用, 还可以在 `launch` 文件中使用, 使用方式如下:

```
<launch>

<node pkg="tf" type="static_transform_publisher" name="link1_broadcaster" args="1 0 0 0 0 0 1 link1_parent link1 100" />

</launch>
```

根据以上静态 `TF` 发布方式, 可将轻舟机器人 `TF` 变换写入 `launch` 文件中, 底盘到激光雷达的 `TF` 变化如下:

```
<node pkg="tf" type="static_transform_publisher" name="base_link_to_laser4"
  args="0.12 0.0 0.08 0.0 0.0 0.0 /base_link /laser_frame 40" />
```

底盘到 `IMU` 的 `TF` 变化如下:

```
<node pkg="tf" type="static_transform_publisher" name="base_link_to_imulink"
  args="0.40 0.0 0.08 0.0 0.0 0.0 /base_link /imu_link 50" />
```