Integrating an Inertial Navigation System with the ROS Navigation Stack

Presenters:
Sarika Ramroop & Shivan Ramdhanie

THE UNIVERSITY OF THE WEST INDIES

VIRTANA

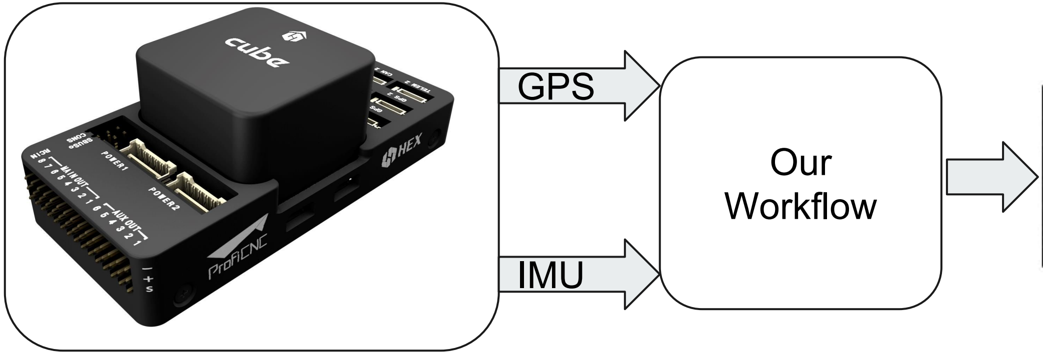
Overview

- Highlighting an approach for integrating an Inertial Navigation System (INS) as a localisation source
- AMCL
  - Sensor data used for localisation
- Our use case: Outdoor scenario and large map
- Our Robot:
  - Large ground robot operating in an airfield
  - Sensors: velodyne, SiCK lasers, pixhawk or similar

Fig. 1: Airfields have few obstacles

Overview cont'd

- INS (GPS + IMU) data is used to generate transforms between various reference frames.



$$\begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2: System Concept

- Converts GPS readings from latitude, longitude, altitude format to the map's cartesian coordinate frame and uses heading information from IMU readings to discern orientation.

System I/O

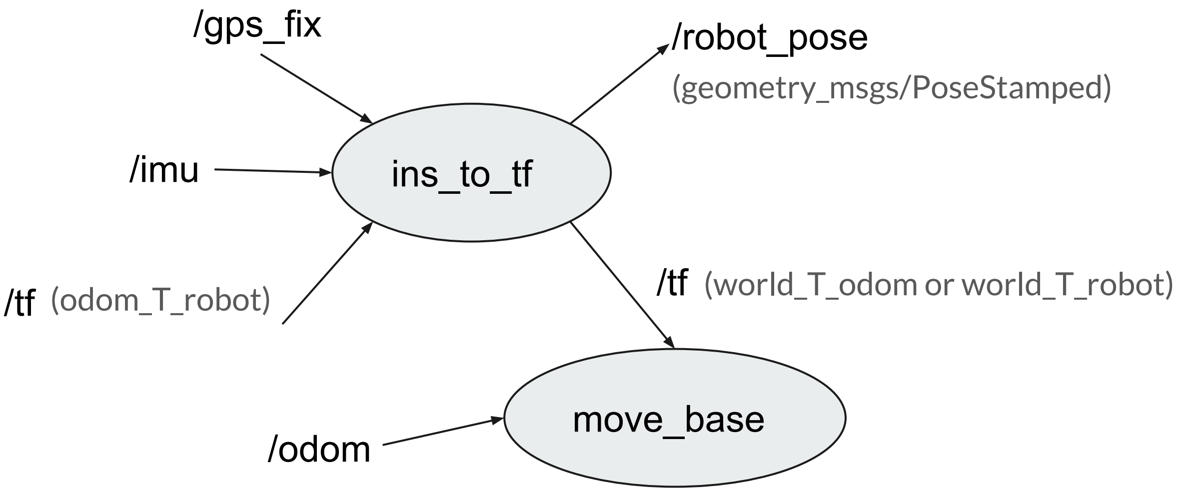


Fig. 3: Node inputs and outputs

Frame Relations

- If both INS and Odometry data are available, the transform tree is:

  Map->Odom->Robot

  where transforms between the map and odom are provided by our node.

- If odometry data is not available, the transform tree is:

  Map->Robot

  where the node publishes the map to robot base_link transform.

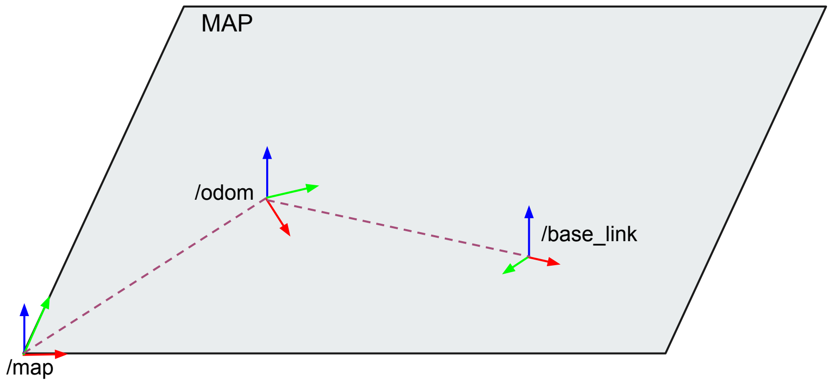


Fig. 4: Frame relations when INS and odom are available

Frame Relations

- If both INS and Odometry data are available, the transform tree is:

  Map->Odom->Robot

  where transforms between the map and odom are provided by our node.

- If odometry data is not available, the transform tree is:

  Map->Robot

  where the node publishes the map to robot base_link transform.

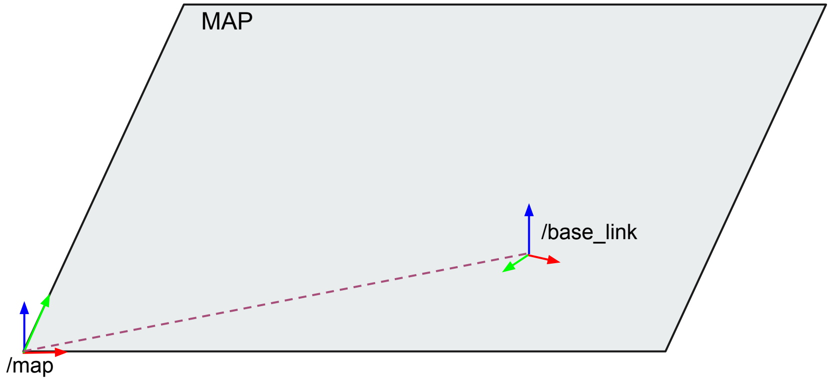


Fig. 5: Frame relations when only INS data is available

Rationale

1. To allow for navigation in the global frame (map) when a goal is specified
2. To allow the local frame (odom) to be placed within the global frame
3. To allow the robot to be placed in the global frame such that it is able to locate it position.
4. Robot is able to navigate using the global and local planners in move_base

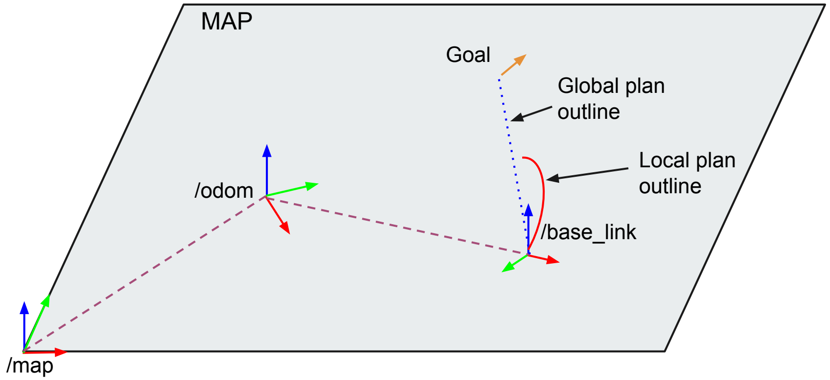


Fig. 6: Using the nav stack with these frame relations

TF tree comparison

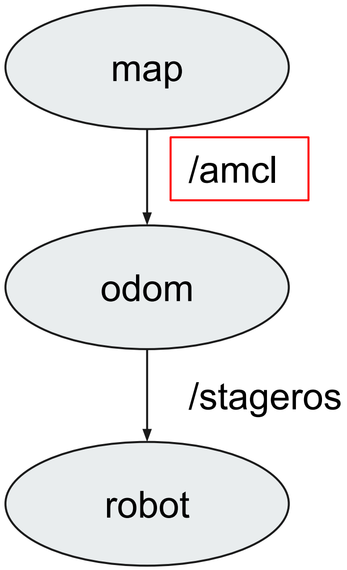


Fig. 7a: TF tree for turtlebot using amcl for navigation

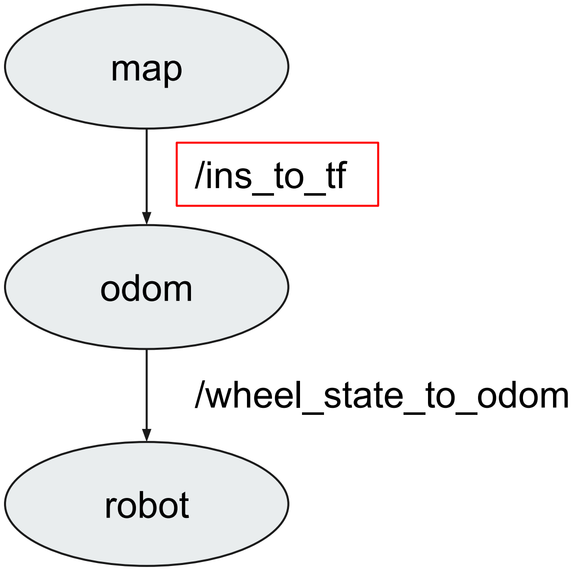


Fig. 7b:TF tree using INS data for navigation

INS data without odometry

- Generates nav_msgs/Odometry messages on the “odom” topic.
- Velocity calculated based on the change in pose over time between successive messages.
- Robot uses the map as both local and global frames.

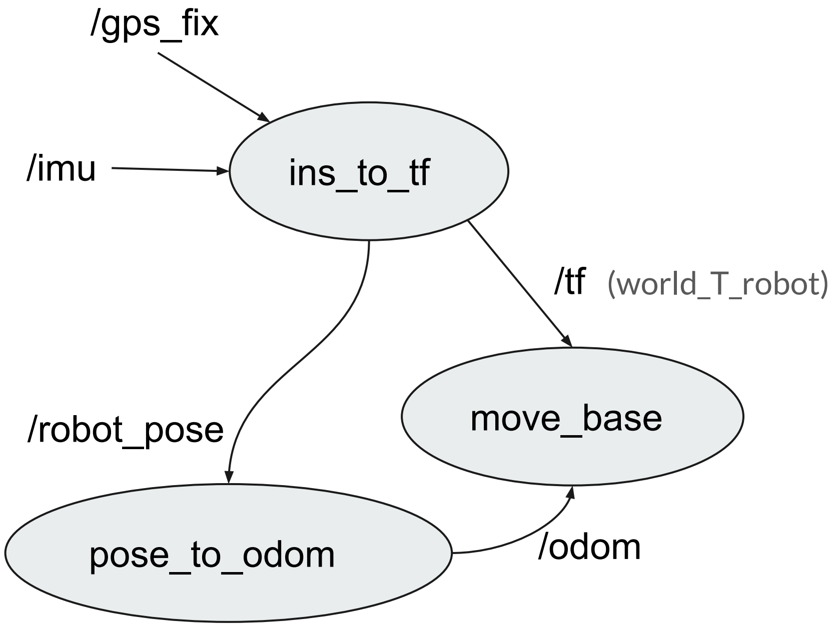


Fig. 8: Node inputs and outputs

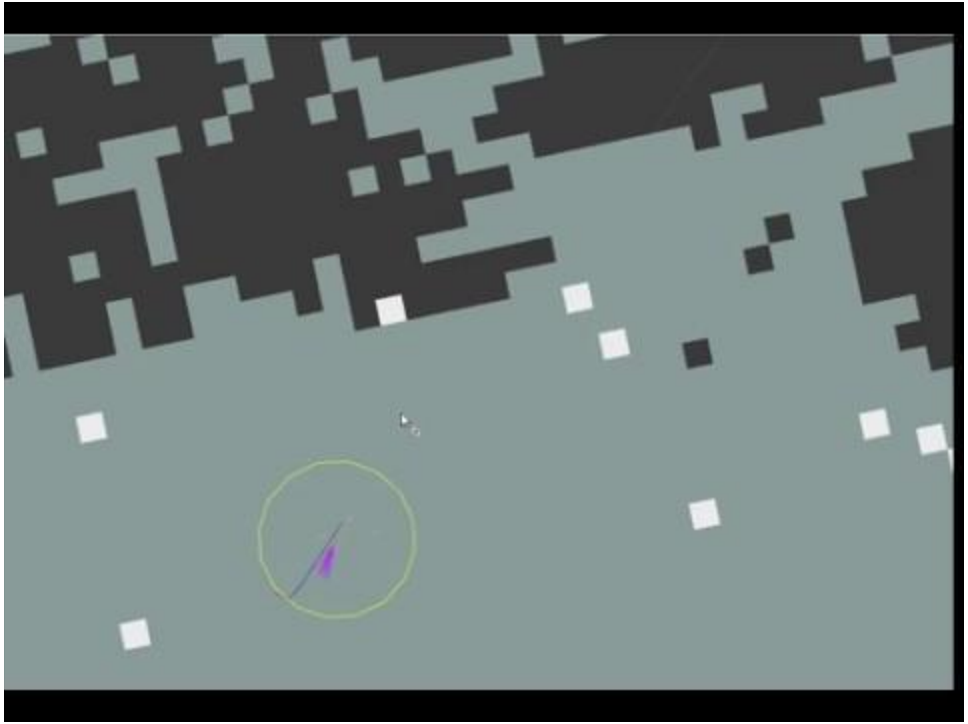
CONCLUSION

Performance

- Able to set and reach goals effectively in simulations (provided they are valid).
- Simulations were done in Gazebo 7.14.
- Hardware testing will be carried out later on.

QUESTIONS?

Demo



TF Tree Comparison

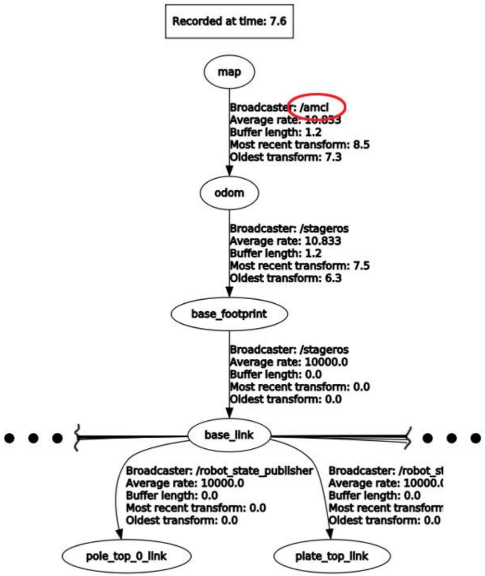


Fig. 9a: TF tree for turtlebot using amcl for navigation

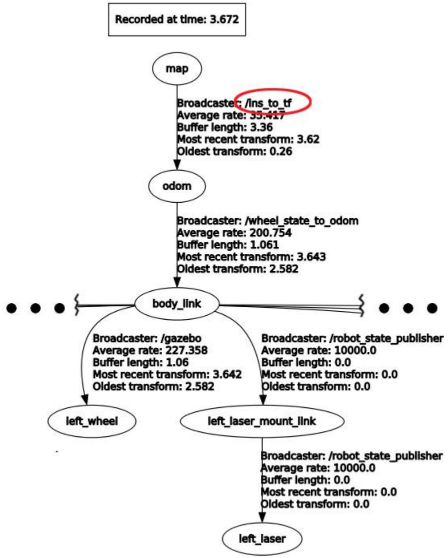


Fig. 9b:TF tree using INS data for navigation

INS data without odometry

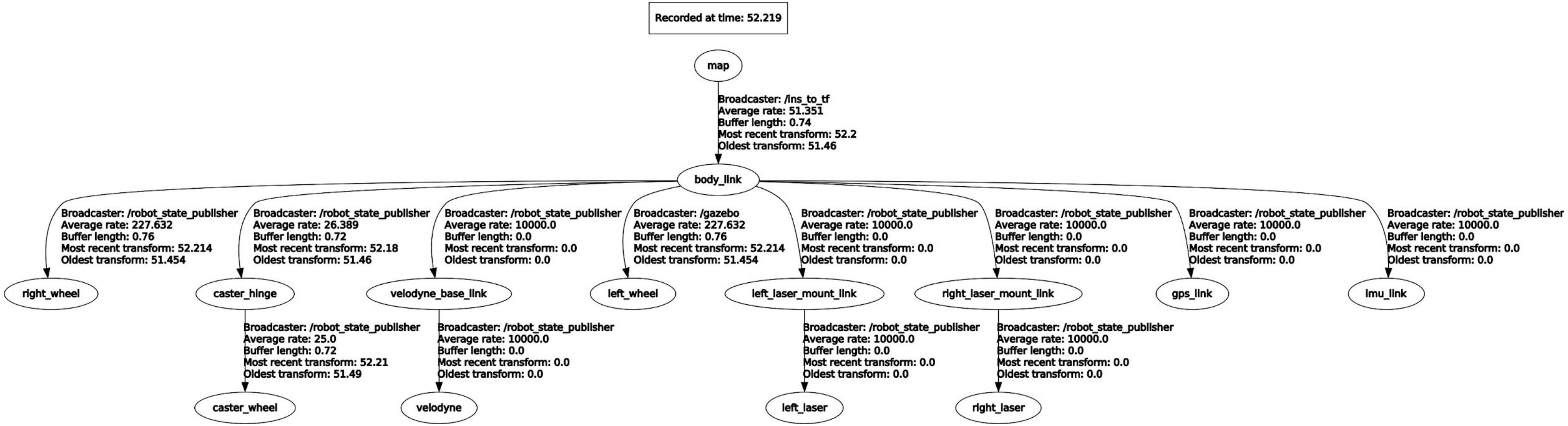


Fig. 10: TF tree when odometry is unavailable