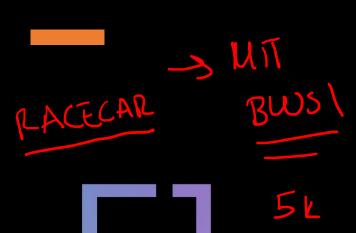
- Pos
- Subs
- Pul CADi

~ Tipos Dat ROS as a Development Platform
- 3°
- Compleio F1Tenth

Rolando Bautista Montesano

rolando.bautista@tec.mx

July 22, 2020



Lidar Circum Visition Wisting Oden

TENTH

models



Key Components



Traxxas 1/10 Ford Fiesta



• Hokuyo 10LX - lide

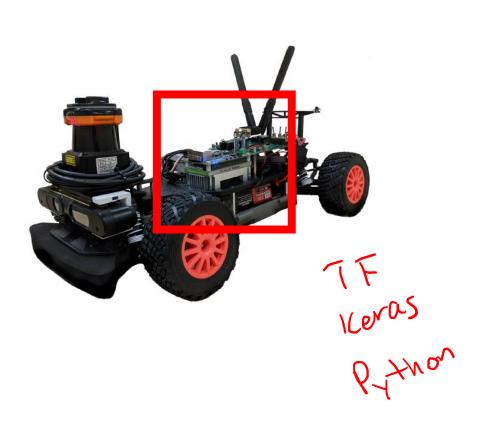


→ Nvidia Jetson TX2 _ 6PU



Nvidia Jetson TX2





- CPU Dual-Core NVIDIA Denver 2 64-Bit CPU
- RAM 8GB 128-bit LPDDR4 Memory
- **§•** ROM 32GB

1 1600 USD 2x 57 Hokuyo 10LX ~ 20 - 50 L/K





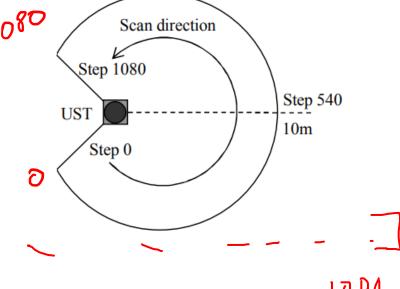


40172



- Measurement steps 1081
- Detection angle 270° Angular resolution 0.25°





Hokuyo Demo

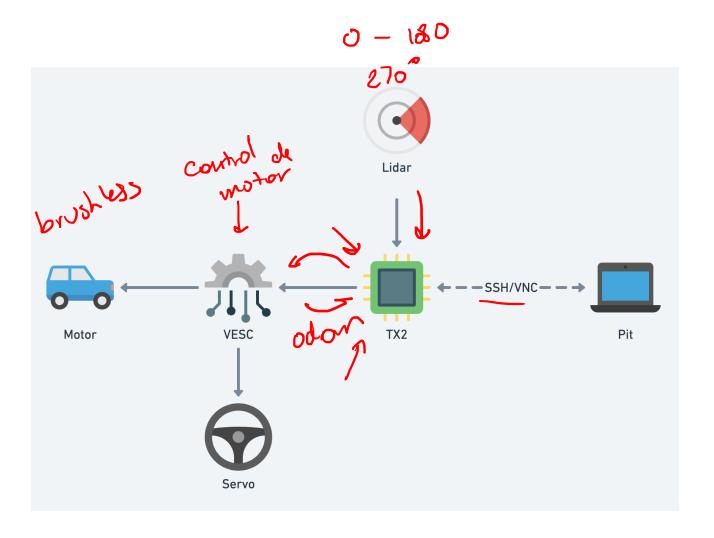
- Installation
- Topics
- RVIZ

General Architecture

Software

- Ubuntu 18.04 LTS
- ROS1 Melodic





How to install a ROS package

- sudo apt-get update
- sudo apt-get install ros-melodic-Ackermann-msgs

- → steering_angle (rad)
 - steering_angle_velocity (rad/s)
- → speed (m/s)
 - acceleration (m/s^2)
 - jerk (m/s^3)

Odometry

- Provides vehicle information
- Pose
- Twist

```
geometry_msgs/PoseWithCovariance pose
    geometry msgs/Pose pose
      geometry_msgs/Point position
        float64 x
        float64 y
        float64 z
      geometry msgs/Quaternion orientation
        float64 x
        float64 y
        float64 z
        float64 w
float64[36] covariance
 geometry msgs/TwistWithCovariance twist
    geometry_msgs/Twist twist
      geometry_msgs/Vector3 linear
        float64 x
        float64 y
        float64 z
      geometry_msgs/Vector3 angular
        float64 x
                      W
        float64 y
        float64 z
  _float64[36] covariance
```

How to run the containerized environment

Terminal 1

- cd catkin_ws/src/f1tenth_gym_ros
- sudo ./docker.sh

Terminal 2

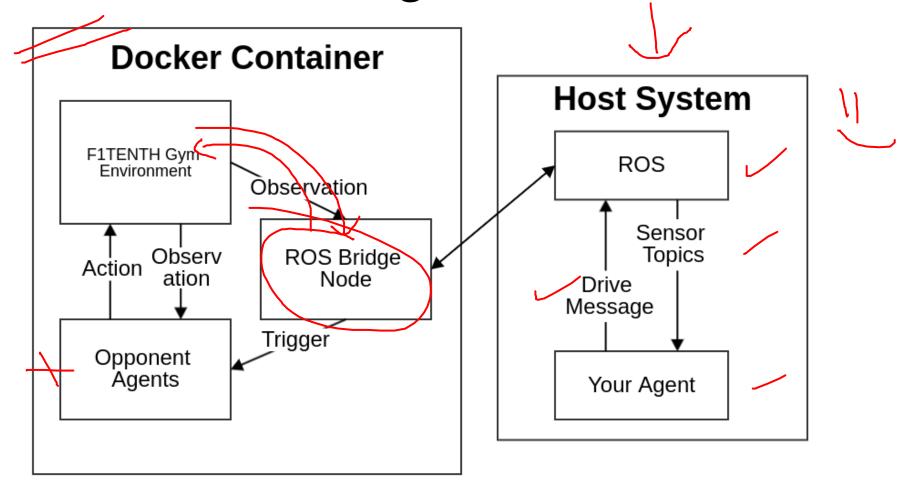
- source ~/catkin_ws/devel/setup.bash
- roslaunch f1tenth_gym_ros agent_template.launch



Challenge

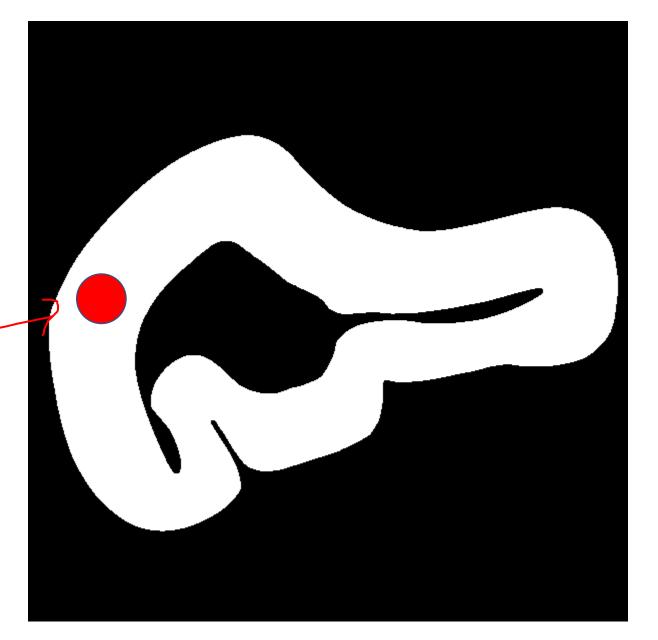


F1TENTH gym environment ROS communication bridge



The environment

- Length ~60m
- Width ~2m



CADi ROS as a Development Platform F1Tenth

Rolando Bautista Montesano

rolando.bautista@tec.mx

July 22, 2020

To do

 Implement a wall follower algorithm that successfully navigates through the track

 The vehicle should complete two laps to pass the assignment.



To do

- Implement a wall follower algorithm that successfully navigates through the track with obstacles.
- The vehicle should complete two laps to pass the assignment.
- Save your path in a csv file using the odometry data.
- Plot your path using matplotlib.



Track with obstacles



• Check all your LIDAR readings!



To do

- Implement a longitudinal control.
- Implement a pure pursuit control.
- Implement a Stanley control.
- The vehicle should complete two laps in the track with obstacles to pass the assignment.



Hints:

- Do you need the same speed during all the track?
- What is your max steering angle?
- Do you need waypoints? (from the previous challenge)