ROS - Robot Operating System

Introduction to ROS
XV SEMATRON

SEMATRON

EESC • USP

Hello world!

Marco Antonio Arruda (Msc.)

Mateus Valverde (Msc. Student)

Vitor Akihiro (PhD Candidate)

GRUPO DE MECATRÔNICA
LABORATÓRIO DE ROBÓTICA MÓVEL

Prof. Dr. Marcelo Becker

What is ROS?

“Flexible framework for writing robot software”

What is ROS?

“Flexible framework for writing robot software”

“Tools, libraries and conventions”

What is ROS?

“Flexible framework for writing robot software”

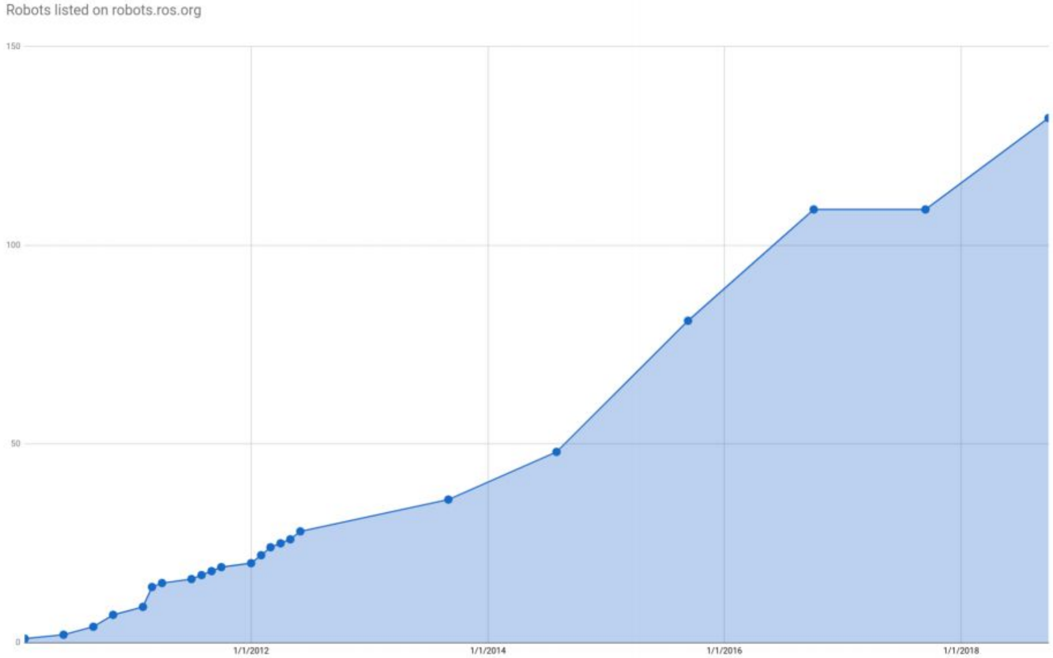
“Tools, libraries and conventions”

Provide faster way to solve complex robotics problems with various robotic platforms

Why ROS?

Trending robotics tools all around the world

Documented ROS Robots

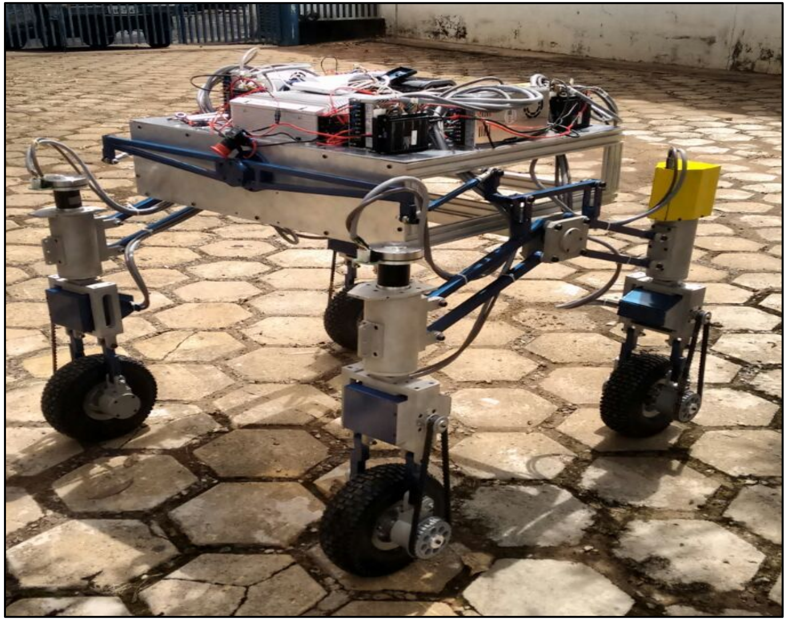


The number of different types of robots available to the community with ROS drivers

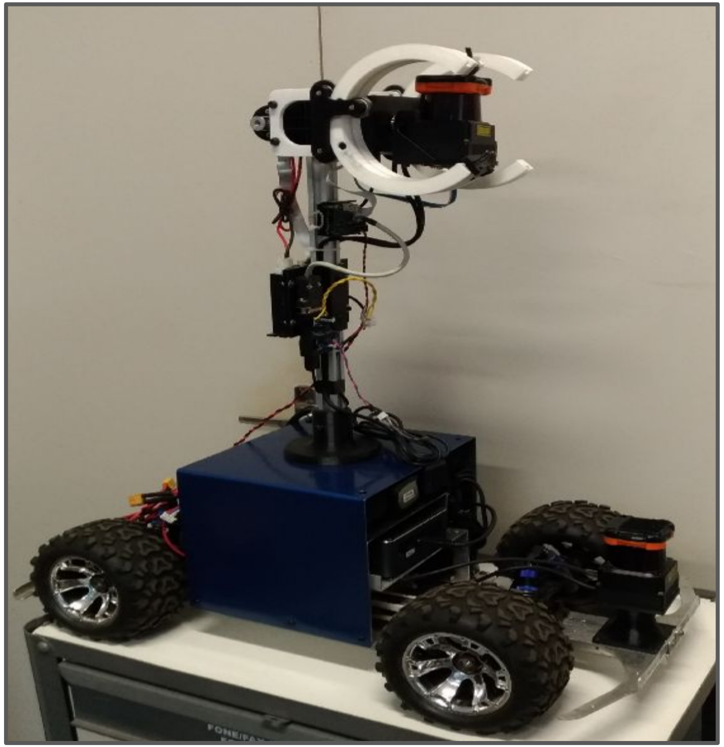
http://download.ros.org/downloads/metrics/metrics-report-2018-07.pdf

Source: Ken Conley, Tully Foote, wiki.ros.org/Robots, 2017 changed over to Robots.ros.org

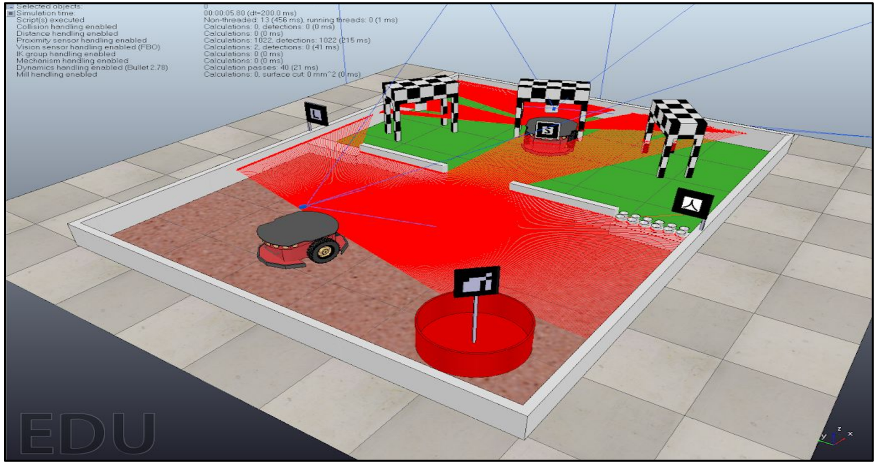
Mirã 2



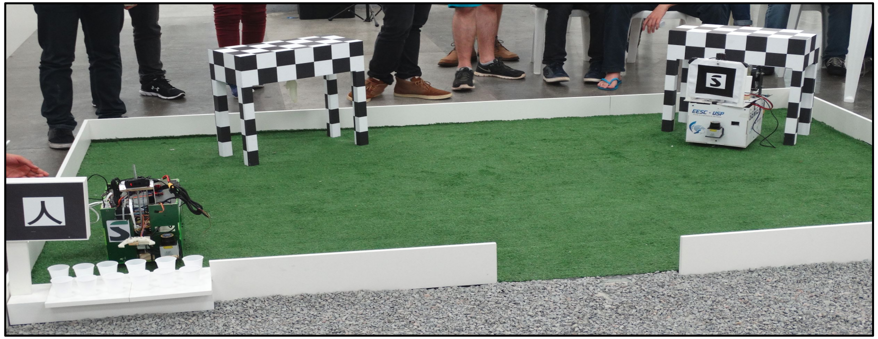
Helvis 4



SEMEAR - IEEE Open



```
Selected objects:                          0
Simulation time:                           00:00:05.80 (dt=200.0 ms)
Script(s) executed                         Non-threaded: 13 (456 ms), running threads: 0 (1 ms)
Collision handling enabled                 Calculations: 0, detections: 0 (0 ms)
Distance handling enabled                  Calculations: 0 (0 ms)
Proximity sensor handling enabled          Calculations: 1022, detections: 1022 (215 ms)
Vision sensor handling enabled (FBO)       Calculations: 2, detections: 0 (41 ms)
IK group handling enabled                  Calculations: 0 (0 ms)
Mechanism handling enabled                 Calculations: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78)    Calculation passes: 40 (21 ms)
Mill handling enabled                      Calculations: 0, surface cut: 0 mm^2 (0 ms)
```



ROS Environment

- Nodes
- Topics + Message
- Publishers, Subscribers
- Services and Parameters

ROS Environment - Nodes

What is a node?

Independent program running in a ROS device (Robot, computer, etc..) able to communicate with other ROS nodes through **topics, services and others**

It doesn’t have to be developer using same language of other nodes. Does NOT interfere in other nodes process

ROS Environment - Topics + Messages

A topic, in a ROS environment, is an open channel of communication (Like a streaming communication)

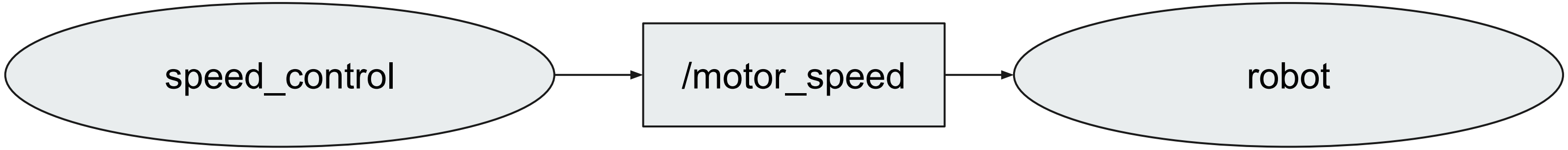
Nodes can send to and read messages from a topic

It’s used to send/receive messages in a given frequency

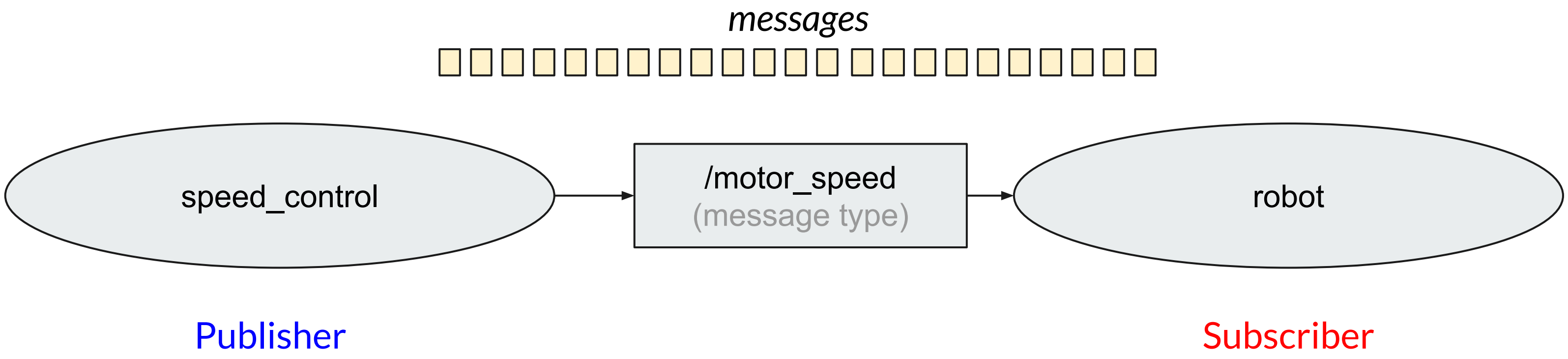
E.g:

1. Read an IMU sensor
2. Continuously motor speed control

ROS Environment - Nodes and Topics



ROS Environment - Publishers and Subscribers



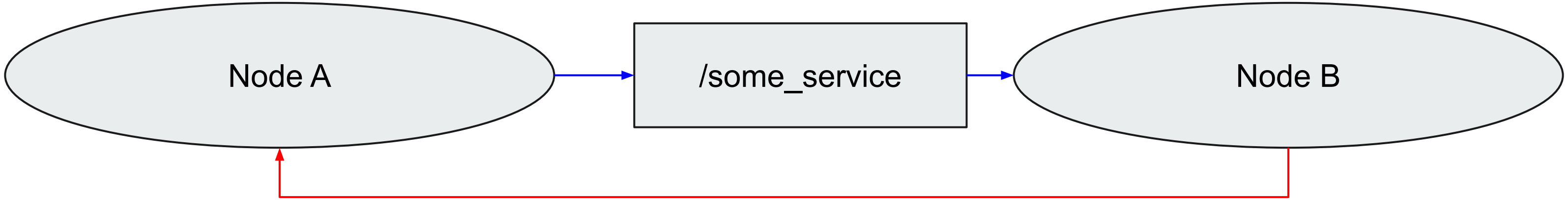
ROS Environment - Services

Request and response behavior (like a client/server architecture)

Node (A) requests a service provided by Node (B)

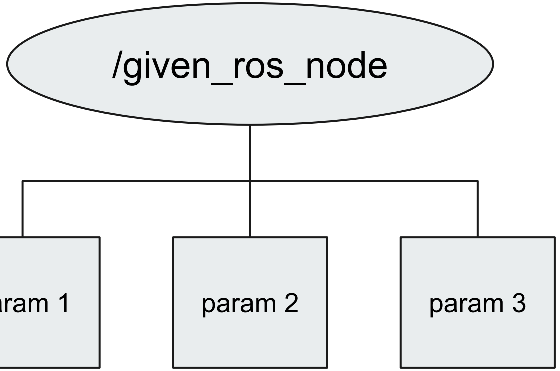
Node (B) responds Node (A)'s request

End of communication!



ROS Environment - Parameters

Environment variables that can be used to tune/configure the behavior of ROS scripts/programs



ROS - Robot Operating System - Let’s practice!!!

Introduction to ROS
XV SEMATRON

SEMATRON

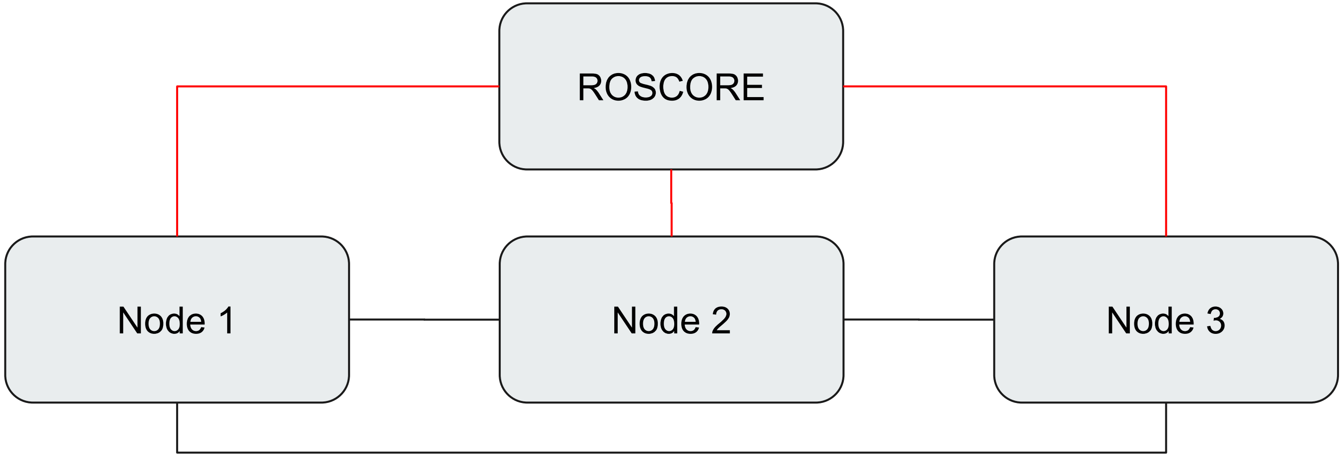
EESC • USP

1 - Meeting ROS workspace

```
cd /home/user (cd ~)

mkdir -p sematron_ws/src

cd sematron_ws/src

catkin_create_pkg sematron_pkg rospy std_msgs geometry_msgs sensor_msgs

cd ~/sematron_ws

catkin_make
```

2 - ROSCORE

Run *roscore* (The server in charge of synchronizing ROS nodes, topics, services, etc.)



3 - Creating a node

my_1st_node.py (-- open file and code line by line)

rosnode list

4 - Creating a publisher

publisher.py (-- open file and code line by line)

rostopic list

rostopic type

rostopic echo

rostopic hz

5 - Creating a subscriber

subscriber.py (-- open file and code line by line)

rostopic pub

rostopic info

6 - Launch file

How can I launch multiple nodes with a single command? Use a launch file!

-- Create launch file for publisher and subscriber --

ROS - Robot Operating System - Simulation!!!

Introduction to ROS
XV SEMATRON

SEMATRON

EESC • USP

Turtlebot 3 Burger/Waffle simulation

Run simulation

Shell 1

roslaunch sematron simulation_empty.launch

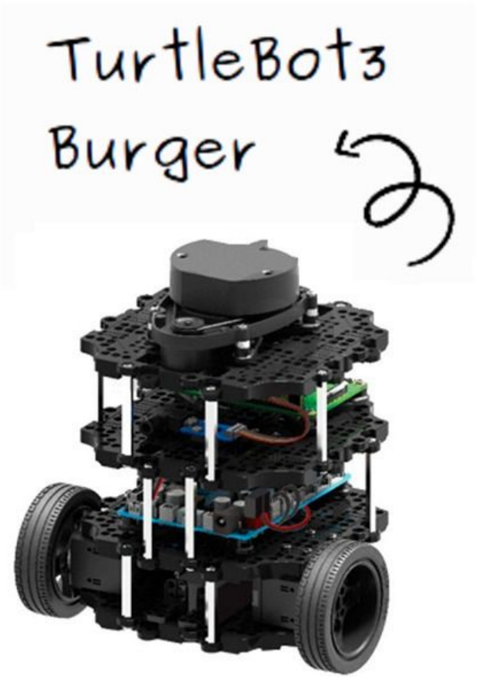
Shell 2

rviz

rostopic list

rosnode list

rqt_graph

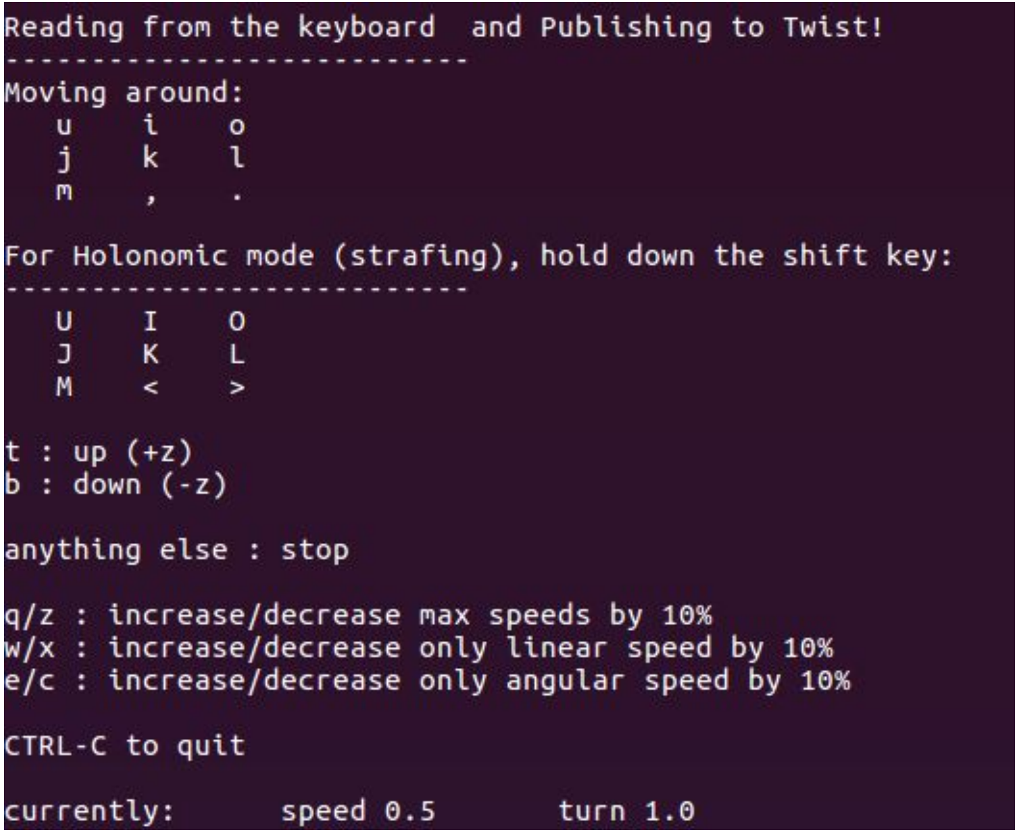


Laser Scan

Differential Driver

Moving the robot using the topic /cmd_vel

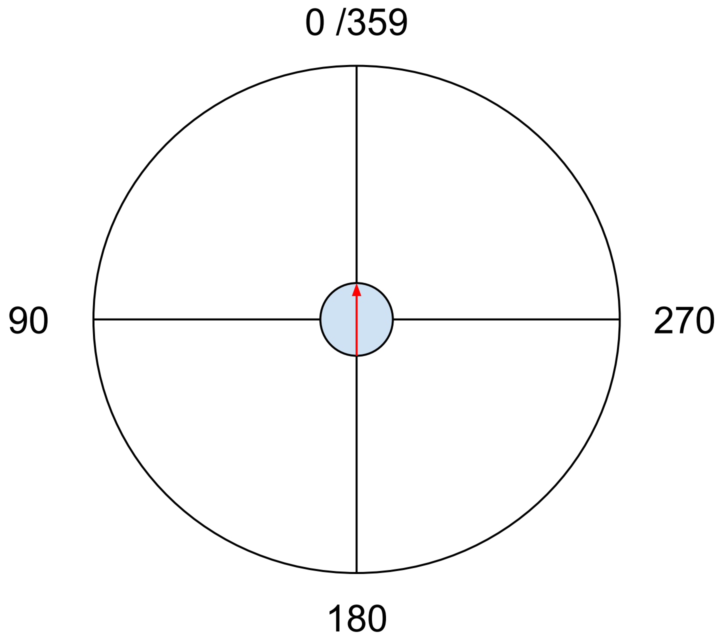
- teleop keyboard



```
Reading from the keyboard  and Publishing to Twist!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

For Holonomic mode (strafing), hold down the shift key:
---------------------------
   U    I    O
   J    K    L
   M    <    >

t : up (+z)
b : down (-z)

anything else : stop

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently:      speed 0.5       turn 1.0
```

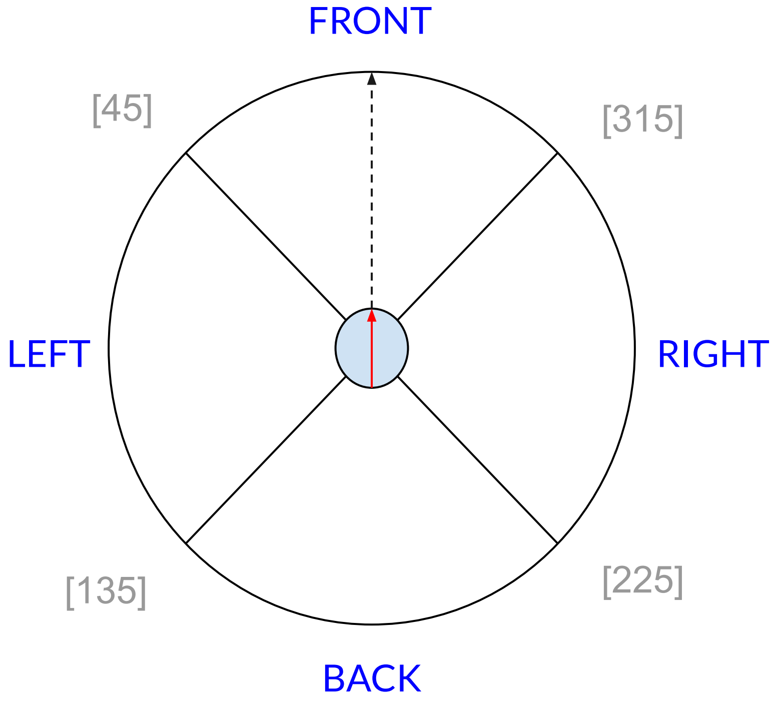
RViz - Laser Scan data

Add laser scan to RViz and configure to read from /scan topic



Read laser scan and organize data

Group laser data by robot orientation



ROS - Robot Operating System - Challenge!!!

Introduction to ROS
XV SEMATRON

SEMATRON EESC • USP

Obstacle Avoidance using ROS!

roslaunch sematron simulation_obstacles.launch

Create a new script in charge of navigation around the Turtlebot world, avoiding the obstacles.

Create a **node** with both, **publisher** and **subscriber**, objects

- Subscribe the laser topic (/scan)
- Publish to the drive topic (/cmd_vel)

References

ROS Official Website - http://wiki.ros.org

ROS Courses - http://wiki.ros.org/Courses

ROS Tutorials - http://wiki.ros.org/ROS/Tutorials

Thank you!

Marco Arruda - me@marcoarruda.com

Mateus Valverde - mateus.gasparino@usp.br

Vitor AKihiro - vitor.higuti@usp.br

Keep Learning!

Repository: https://github.com/marcoarruda/sematron_2019
(Code + Slides)

Turtlebot 3 simulation

https://github.com/robotis-git/turtlebot3

https://github.com/robotis-git/turtlebot3_msgs

https://github.com/robotis-git/turtlebot3_simulations