Practical Work

Local planner:
Nearness Diagram (ND)

Master in Robotics, Graphics and Computer Vision
Escuela de Ingeniería y Arquitectura de Zaragoza

Ignacio Herrera Seara
Juan García-Lechuz Sierra

Index

1. Introduction
2. Algorithm description
3. Implementation
4. Experimental results
5. Conclusions

1 - Introduction

GOAL: Complete tasks in different scenarios despite the complexity of the environment.

Reactive navigation strategies

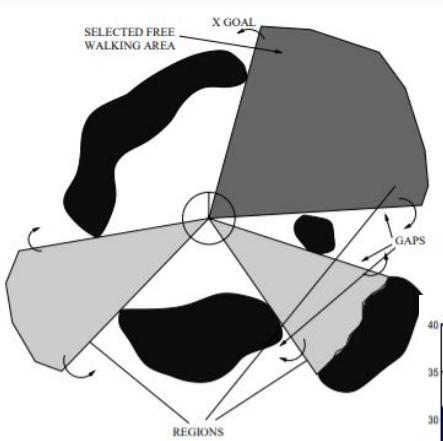
ND

- Robotics Group (University of Zaragoza).
- Improvements:
  - Interpreting the environment.
  - Actions computation.
- **Complex** situation ⟶ **simple** action

ROBOTICS
UNIVERSIDAD DE ZARAGOZA

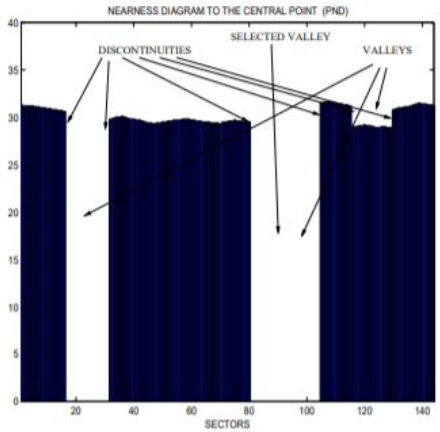
2 - Algorithm description

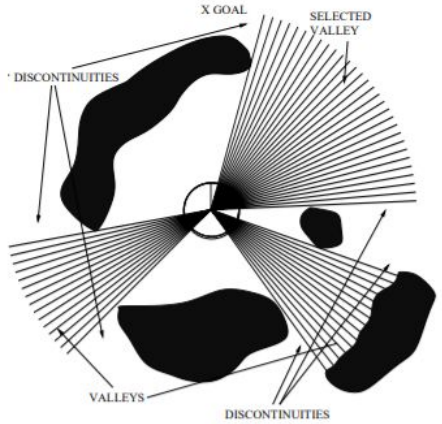
ENVIRONMENT ANALYSIS



PND

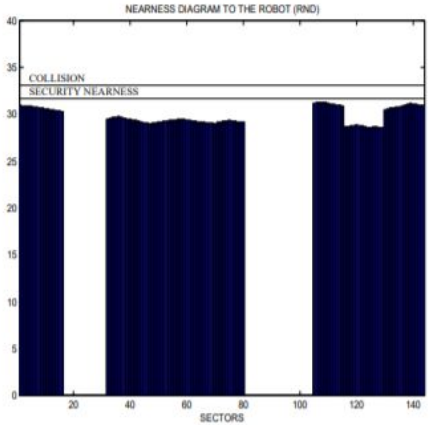
Nearness of the obstacles to the central point



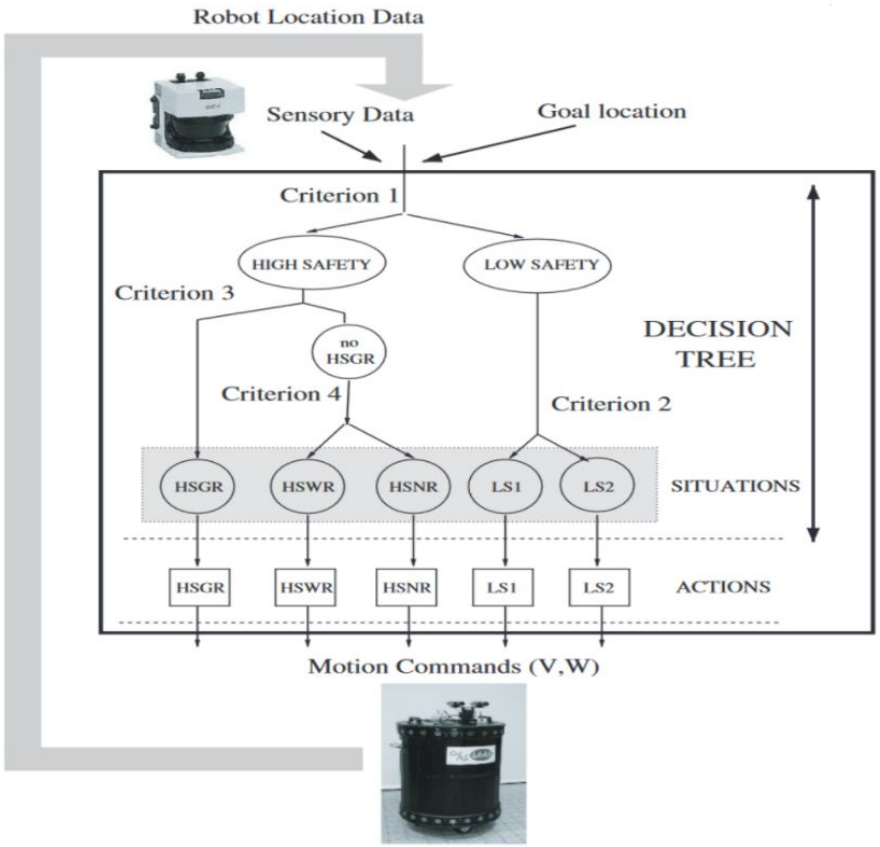


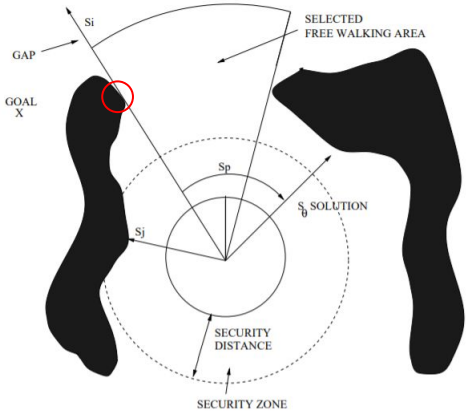
RND

Nearness of the obstacles to the robot bounds

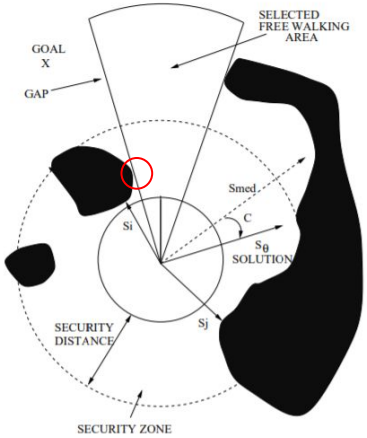


2 - Algorithm description





LS1



LS2

2 - Algorithm description

NAVIGATION STRATEGY

Translational velocity

Direction

$$\boldsymbol{\theta} = bisector(s^{\theta})$$

Module

HS $$\mathbf{v} = v_{max} * Abs(1 - \theta/\frac{\pi}{2})$$

LS $$\mathbf{v} = v_{max} * \frac{d_{obs}}{d_s} * Abs(1 - \theta/\frac{\pi}{2})$$

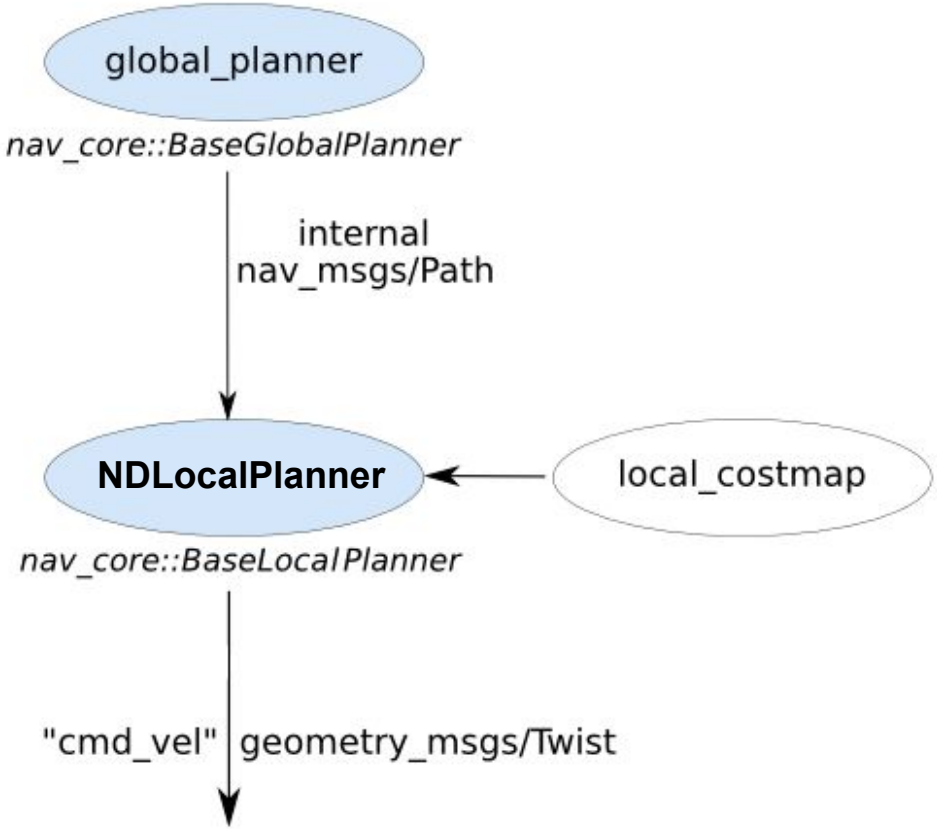
Rotational velocity

$$\mathbf{w} = w_{max} * \theta/\frac{\pi}{2}$$

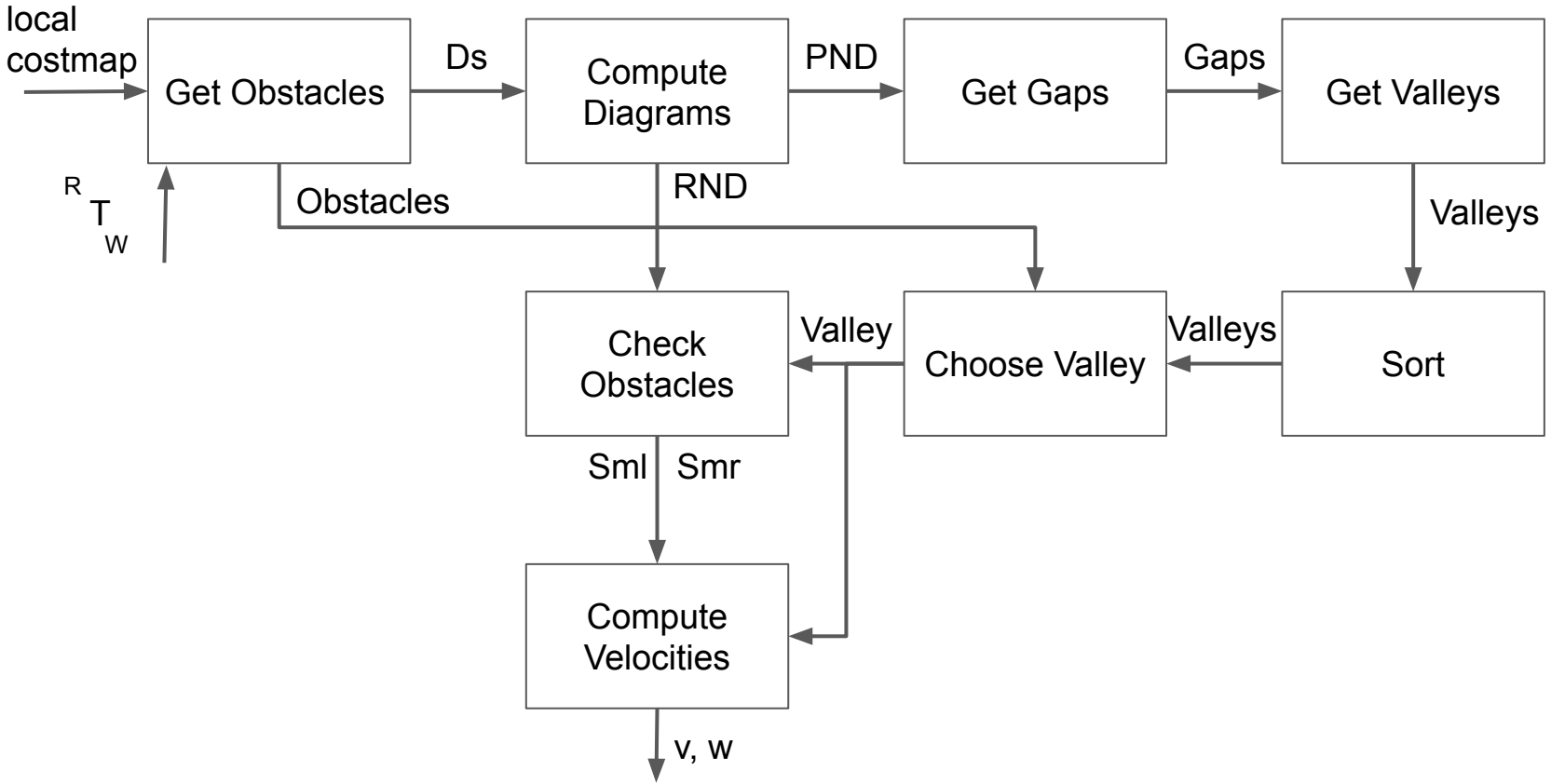
s_θ: solution sector
d_obs: distance of the closer obstacle to the robot bounds
d_s: security distance.
v_max: maximum translational velocity.
w_max: maximum rotational velocity.

3 - Implementation

ROS Melodic
Navigation Stack

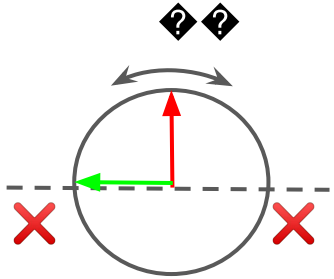


3 - Implementation



3 - Implementation

Compute Direction



LS1

$$\rho = 1 - \frac{|s_{rd} - s_{ml}|}{n/2}$$

$$s_\rho = \frac{s_{max}}{4} * \rho + \frac{s_{max}}{2}$$

$$s_\theta = s_{rd} + sign(s_{rd} - s_{ml}) * s_\rho$$

LS2

$$c = D * (1 - \frac{min(d_l, d_r)}{d_l + d_r})$$

$$s_\theta = \frac{s_{ml} + s_{mr}}{2} \pm c$$

HSGV

$$s_\theta = s_g$$

HSWV

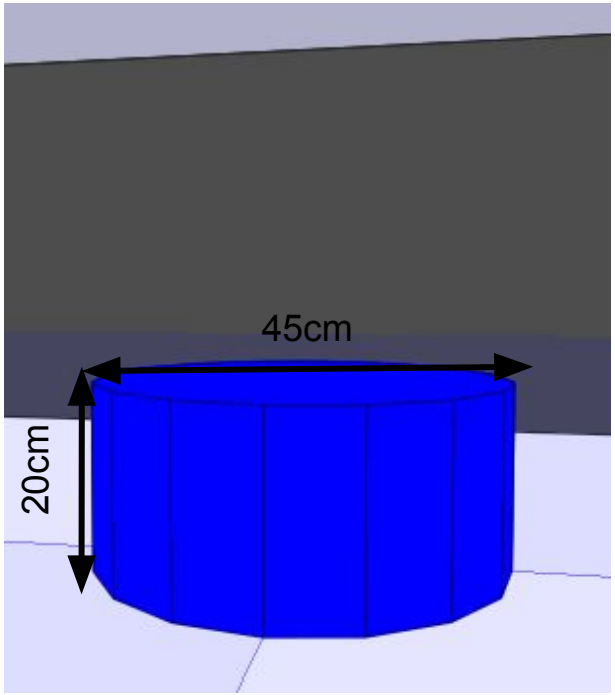
$$s_\theta = s_{rd} + \frac{s_{max}}{2}$$

HSNV

$$s_\theta = \frac{s_l + s_r}{2}$$

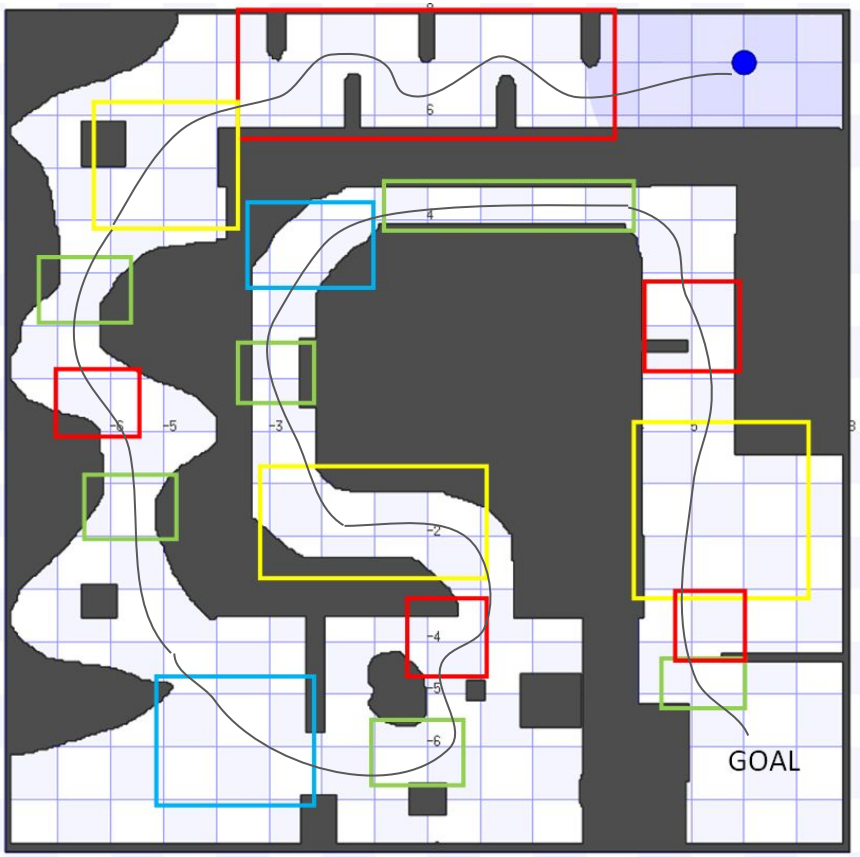
$$\theta = \text{bisec}(s) = \pi - \frac{2.\pi}{n} \cdot s$$

4 - Experimental results



Parameter	Value
vmax	0.5 m/s
wmax	1.57 rad/s
dmax	3 m
fov	360
ds	0.38m
n	144
smax	36
D	3

4 - Experimental results



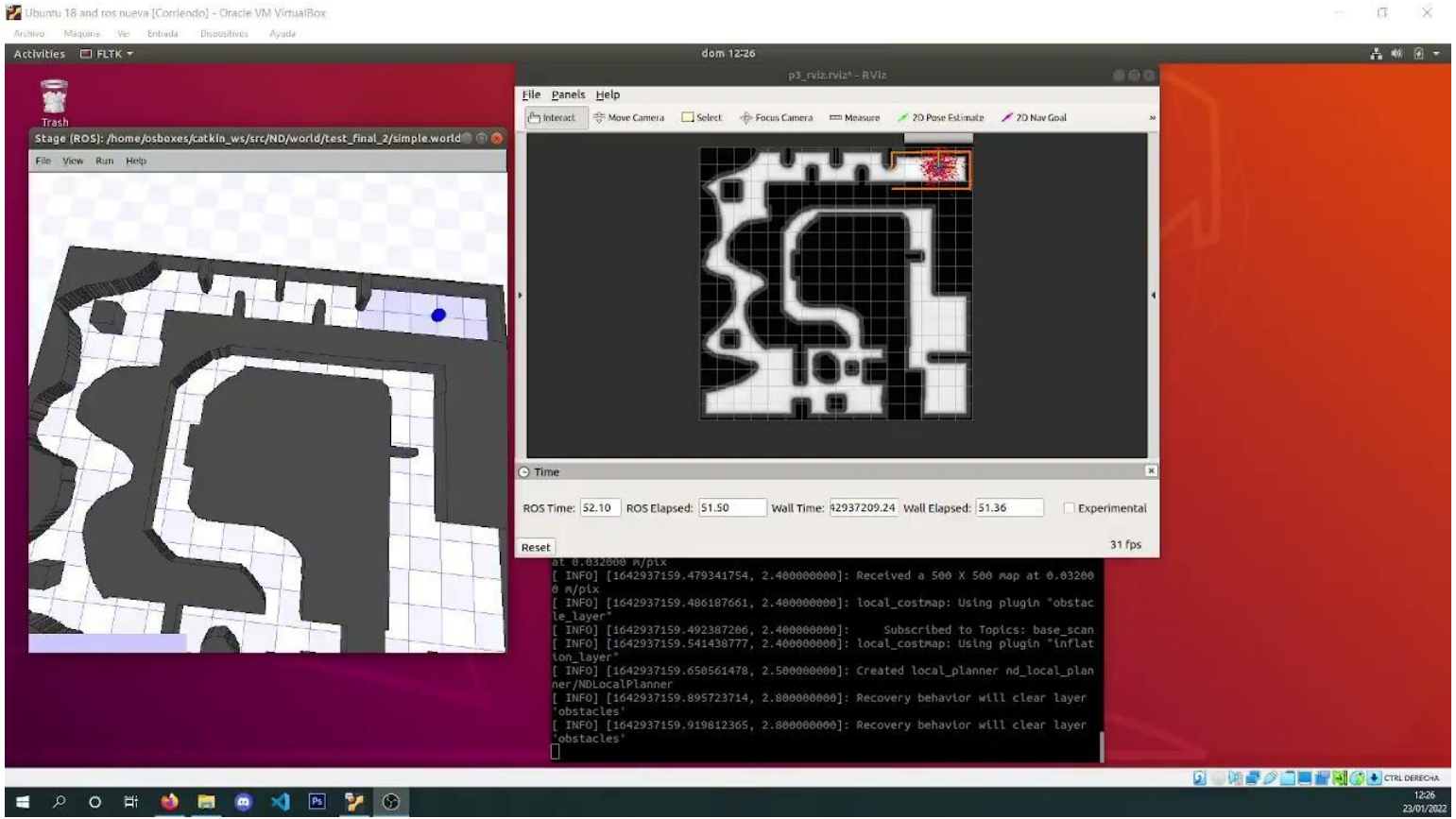
Final Environment

Nearness Diagram:

- Avg. computation time of the algorithm: *8.15ms*
- Avg. time to finish: *3min 24s*

DWA: fails on the map ❌

4 - Experimental results



5 - Conclusions

- The ND reduces the complexity of navigation to a series of motion commands, extracted from a specific analysis of the environment

- The ND allows the robot to navigate unknown, highly complex and dynamic environments where other approaches have difficulty navigating safely and fluidly.

- One of the advantages of this method is that it allows to improve the development of navigation by adding more situations to the generation of motion commands, as occurs in the ND+, where a new low-security situation is added.

- In this work ND has been developed for the case of a non-holonomic circular robot, but the planner can also be extended to work with robots of different shapes and constraints.