POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

DWA vs custom trajectory tracking controller: a comparison in ROS

Project for the Control of Mobile Robots course
Computer Science and Engineering

Giuseppe Chiari, 10576799
Leonardo Gargani, 10569221
Serena Salvi, 10607377

Supervisor:
Luca Bascetta

Academic year:
2022/2023

Abstract:
The Dynamic Window Approach (DWA) is an online collision avoidance strategy for mobile robots. It incorporates the dynamics of the robot by reducing the search space to only the velocities reachable within a short time interval.
In this work we first present a comparison between the DWA algorithm from the paper and its implementation in Robot Operating System (ROS).
Then, a further comparison is made between the implementation above and a custom trajectory tracking controller, which is composed of an inner linearisation law (based on the kinematic model) and an outer tracking law (based on a proportional integral controller with velocity feed-forward).

Table of Contents

1. Introduction

... (state that the experiment has been carried on ROS Melodic)

2. DWA overview

...

3. DWA in ROS

3.1. From ROS wiki

...

3.2. Comparison with the paper formulation

...

4. Setup of the experiment

4.1. The robot

In our experiment we chose to simulate a small differential drive robot.

In particular, it is characterized by two main dimensions (specified as YAML parameters in the code):

- `d` = 15 cm, which is the distance between the two motorized wheels;
- `r` = 3 cm, which is the radius of the two motorized wheels.

The precise footprint is a pentagon, just for convenience, so that when looking at it we are able to determine the orientation of the robot. However, this is not a decisive detail since it has o influence on the robot's behavior.

4.2. The map

Regarding the map, it is important to highlight the different setting we have with respect to the usual DWA use.
There are two main differences:

- in our setting there are no obstacles, neither fixed nor moving;
- the robot does not have any sensor.

As a consequence, we don't need both the local map and the global map, but only the local one.
Moreover, this map needs to be empty so it is generated starting from a totally white image.

4.3. The trajectory

The robot must follow a precise trajectory, which is used to perform all benchmarks.
In our case we chose an eight-shaped trajectory with a dimension of 2 x 1 meters.

In order to make DWA compute the velocities of the robot, we must feed it a goal.
This means that the complete trajectory has to be "discretized" in multiple points. Each one of these points is passed to DWA as the current goal, and once it is reached the next point is set as the new goal. You will find a detailed explanation in the following sections.

5. Implementation

5.1. Architecture overview

... (things in common between our controller and DWA: simulator, ...)

5.2. Simulator

...

5.3. Trajectory controller

...

5.4. DWA controller

...

6. Parameters tuning

...

7. Experimental Results

... (plots of the bags + plots of the comparison with the custom script)

8. Encountered problems

... (deprecated parameters name in the official doc: put screen of the doc + screen of the comments in the code of the library)

9. Usage of the code

...

10. Conclusions

...

11. Conclusionse

...