Universidad
Rey Juan Carlos

Mobile Robotics (III). Obstacle Avoidance (Reentrega).

Cristian Sánchez Rodríguez

47584641X

Course 2021/2022

Introduction.

In this project, we will solve the obstacle avoidance problem.

To do it, we implement the VFF algorithm. This works like a force compensation, where attractive forces are generated by our target, and repulsive forces by any obstacle detected.

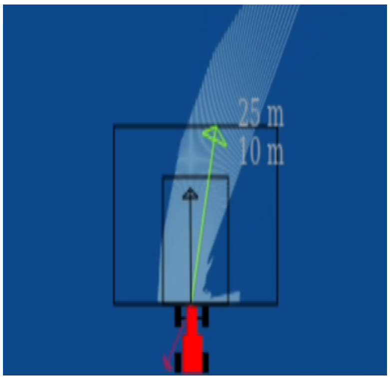

Fig 1. Force representation.

Attractive force

The first step is to obtain the force generated by our sub objective (target). To do it, we need to use the relative pose with respect to the car, in other words, we use the distance target-car in car coordinates.

Then, we have to limit the magnitude for long distances, it has to be affordable.

At this point, we can choose different scenarios, shown in the images below:

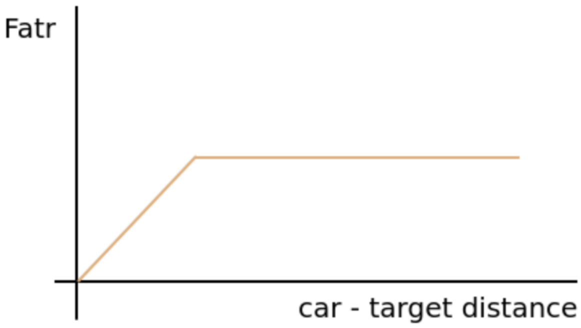

Fig 1.1 Normal approach.

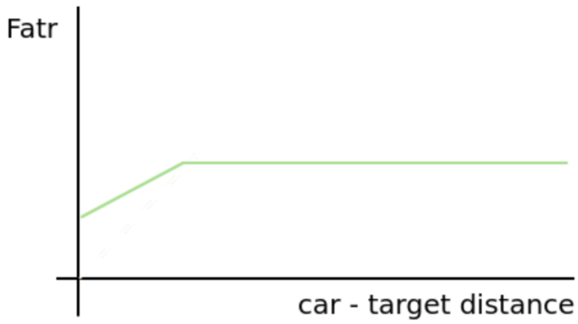

Fig 1.2 Mixed approach.

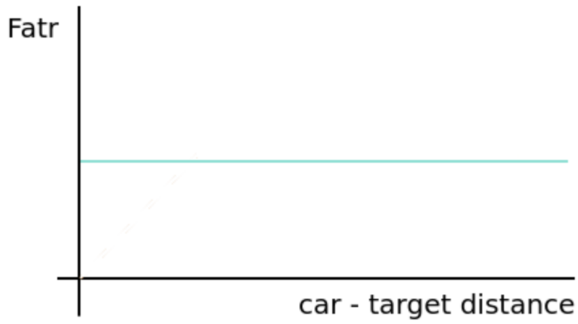

Fig 1.3 Constant approach.

I've decided to use the **normal approach**, because we have to complete with prudency, as fast as we can one lap in the circuit using a sequence of sub-objectives. This method provides us with a way to achieve sub-objectives by avoiding oscillations.

In normal navigation, the first two approaches are the best choice. This is because we hit the final position with more precision.

Repulsive force

In order to follow the VFF theory, we establish one value per laser measure.

To be consistent, we have to define an inverse proportionality dividing the distance car - obstacle by one. We want to get higher values when the distance is less.

Next, we get x, y components oriented properly, and we sum all together to obtain a repulsive resultant.

Resultant force

Last part, we have to join attractive and repulsive forces, adjusting two factors, alpha and beta:

$$F_{resultante} = \alpha F_{atractiva} + \beta F_{repulsiva}$$

Increasing alpha is equal to giving importance to the attractive forces, this is translated into risky behaviors where collisions can occur.

On the other hand, increasing beta value causes fearful behaviors, where goal is never reached (not path found or local minimum problems).

Conclusions and comments.

Once the factors previously mentioned are adjusted, our car can properly navigate the circuit avoiding other cars.

Lap times, using different alpha/beta values, oscillate between 2 min and more than 6 min, also depending on the implementation (more prudent is slower). (It’s noteworthy that times are slower if gazebo and the recording program are running).

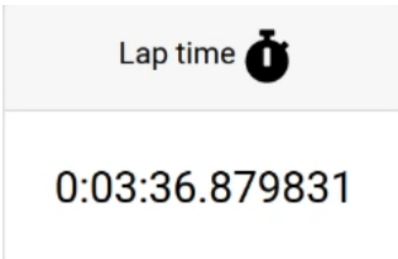

Fig 2.1 riskyVFF Lap Time.

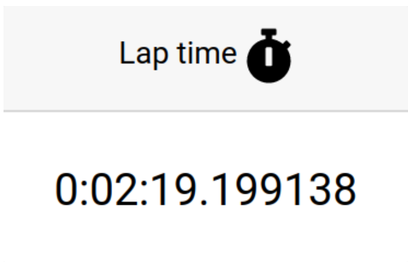

Fig 2.2 riskyVFF without recording and Gazebo running.