Velocity Obstacle

Rahul Sajnani
20171056
International Institute of Information Technology, Hyderabad, India
rahul.sajnani@research.iiit.ac.in

1 Introduction

Velocity Obstacle formulation is a planning algorithm that transforms dynamic obstacles to the robot's perspective and avoids them. In this report we obtain the velocity command that the robot should take in order to reach its goal and avoid obstacles.

2 Assumptions

For this report, we make the following assumptions:

1. Obstacles are circular and their **radii are known**.
2. Obstacle **velocities are known** at every time step.
3. Obstacle velocity are constant and no acceleration occurs.

3 Problem formulation and solution

Our problem is to estimate a velocity vector $\mathbf{v}$ that avoids obstacles $i = 1, 2, ..., n$. The i^{th} obstacle has radius r_i^o, velocity $\mathbf{v}_i^o$, and position $\mathbf{p}_i^o$. We begin by formulating the obstacle avoidance constraint for the i^{th} obstacle. Let the position of the ego robot be $\mathbf{p}$ and its radius be r, we transform the dynamic obstacle to the robot's reference frame.

$$\mathbf{p}_i^r = \mathbf{p}_i^o - \mathbf{p} \tag{1}$$
$$\mathbf{v}_i^r = \mathbf{v} - \mathbf{v}_i^o \tag{2}$$
$$r^r = r + r_i^o \tag{3}$$

Here, $\mathbf{p}_i^r$ and $\mathbf{v}_i^r$ are the relative position of the obstacles and relative velocity by which the robot is approaching the obstacle respectively.

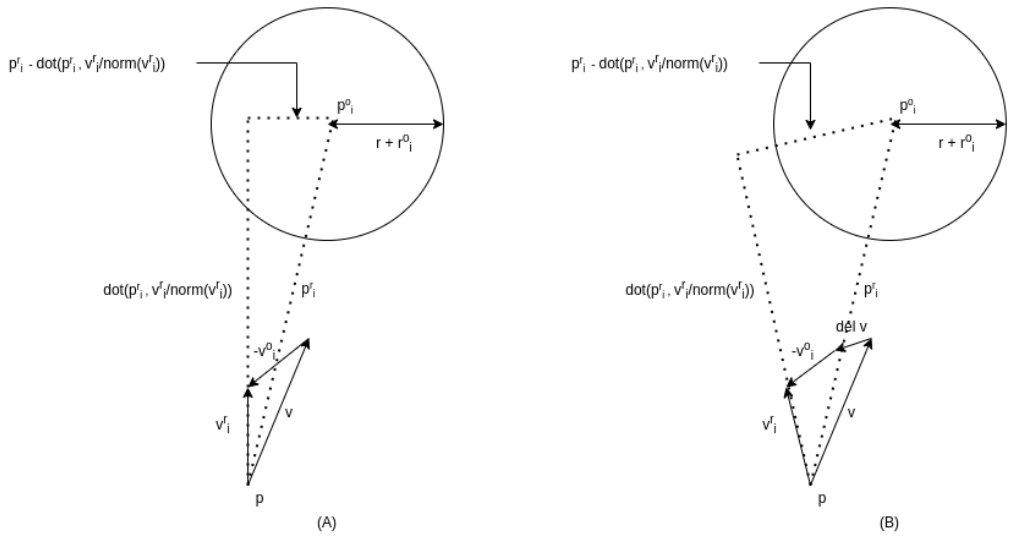


Figure 1: **Obstacle avoidance constraint**: (A) Collision (B) Avoidance. The circle represents the velocity obstacle and **p** represents the position of the robot in the world. . When the vector subtraction of the $\mathbf{p}_i^r$ with its projection on $\mathbf{v}_i^r$ we have avoided the obstacle.

To avoid the obstacle we project $\mathbf{p}_i^r$ in the direction of v_i^r. If the vector subtraction of $\mathbf{p}_i^r$ and the resultant project is smaller than the obstacle radius then the velocity direction collides with the obstacle. This can be visualized in figure 1 part (A).

$$\mathbf{p}_i^{proj} = \mathbf{p}_i^r \cdot \frac{\mathbf{v}_i^r}{||\mathbf{v}_i^r||_2} \tag{4}$$

$$||\mathbf{p}_i^r||^2 - ||\mathbf{p}_i^{proj}||^2 < (r + r_i^o)^2 \qquad (Collision \quad (A)) \tag{5}$$

Now we need to estimate $\Delta\mathbf{v}$ such that its vector addition with $\mathbf{v}$ avoids the obstacle as shown in figure 1 part (B). By adding $\Delta\mathbf{v}$, we are changing the direction of the robot such that it avoids **all obstacles** at **every time step**.

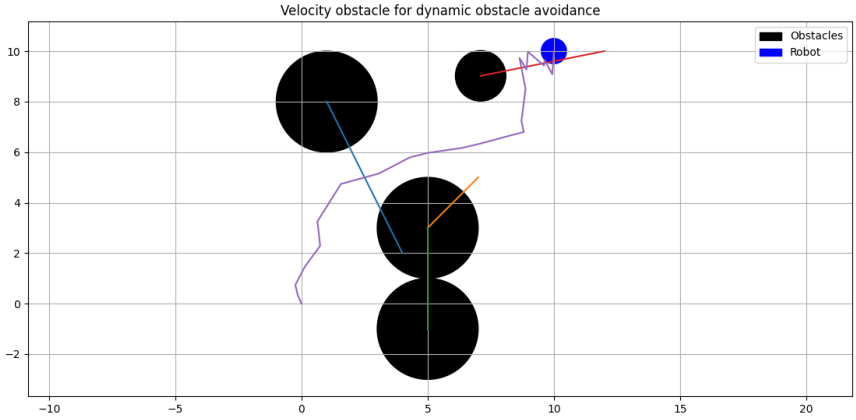
$$^{new}\mathbf{v}_i^r = \mathbf{v}_i^r + \Delta\mathbf{v} \tag{6}$$

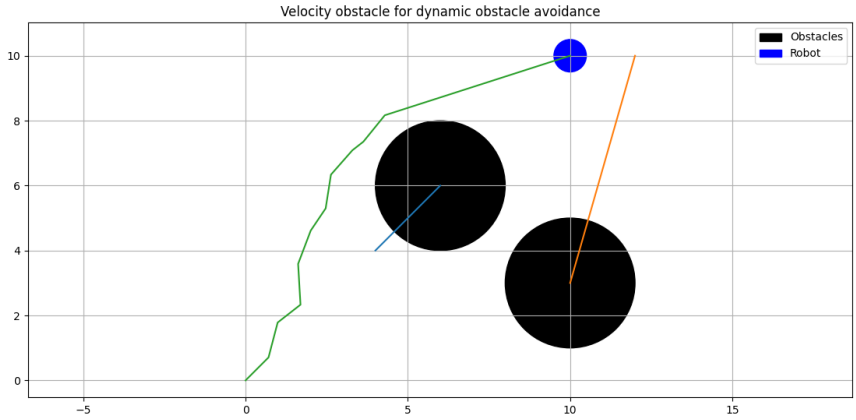
$$^{new}\mathbf{p}_i^{proj} = \mathbf{p}_i^r \cdot \frac{(\mathbf{v}_i^r + \Delta\ \mathbf{v})}{||\mathbf{v}_i^r + \Delta\ \mathbf{v}||_2} = \mathbf{p}_i^r \cdot \frac{^{new}\mathbf{v}_i^r}{||^{new}\mathbf{v}_i^r||_2} \tag{7}$$

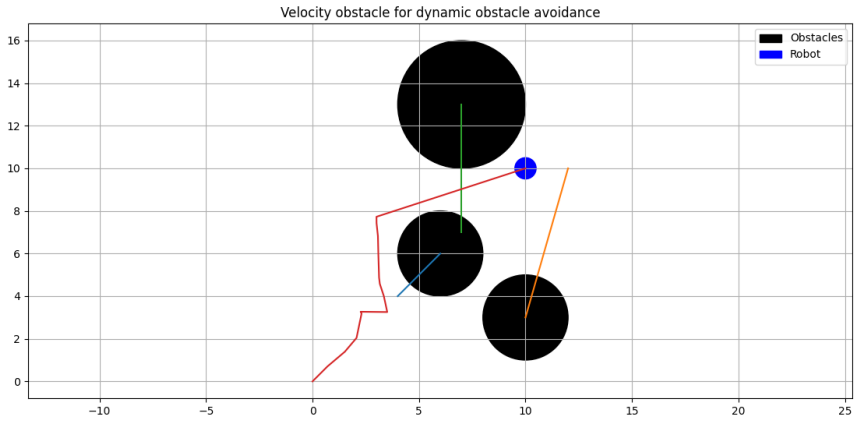
$$||\mathbf{p}_i^r||^2 - ||^{new}\mathbf{p}_i^{proj}||^2 > (r + r_i^o)^2 \qquad (Collision \quad avoided \quad (B)) \tag{8}$$

To obtain $\Delta\mathbf{v}$, we uniformly sample over the angles and velocity commands that can satisfy the obstacle avoidance constraints for all obstacles.

4 Results







5 Deliverables

The code performs the dynamic obstacle avoidance by using velocity obstacle. It also provides configuration file to change the number of simulation parameters in a simple manner.

5.1 Running the code

Change the configuration in *config/config.yaml* and run the code.

```
python simulator.py
```

5.2 Contributions

Collision cone, agent, simulator, and report - Rahul Sajnani