POLITECNICO
MILANO 1863

Autonomous Vehicles

Assignment III

Daniele Cianca 10764733
Lecturer: Stefano Arrigoni

07/11/2024

Politecnico di Milano

Aim of the problem

Starting from initial position of the robot:

- Define a “red sphere” (c83030) of 0.1m radius 6 m on the left
- Define a “purple sphere” (c80067) of 0.2m radius 6 m in front
- Define a feedback control in order to move the robot from initial position to collide with “red sphere”

First initialize the environment creating the two spheres:

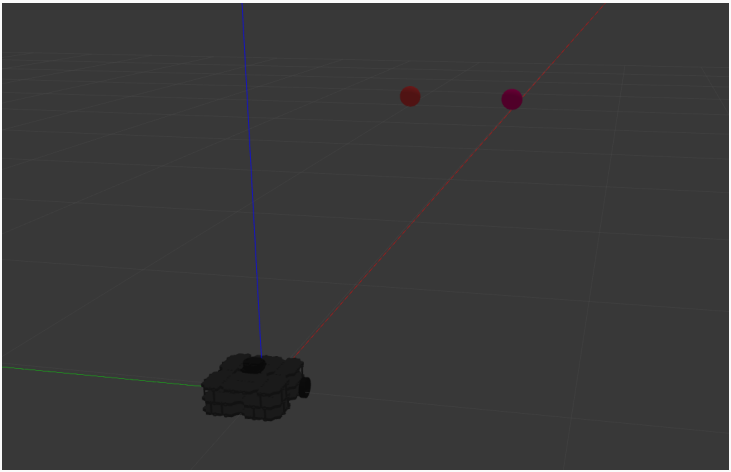

Figure 1: Gazebo environment

In order to detect the red sphere the following SIMULINK model is developed:

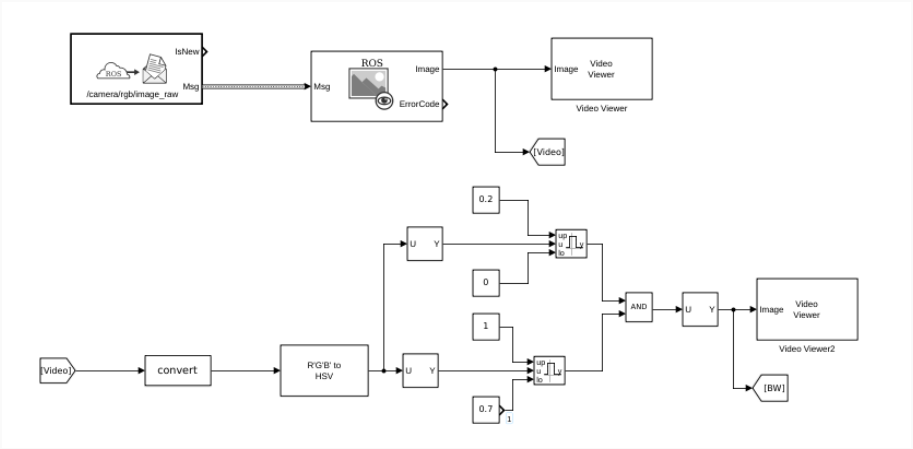

Figure 2: Simulink model for detection

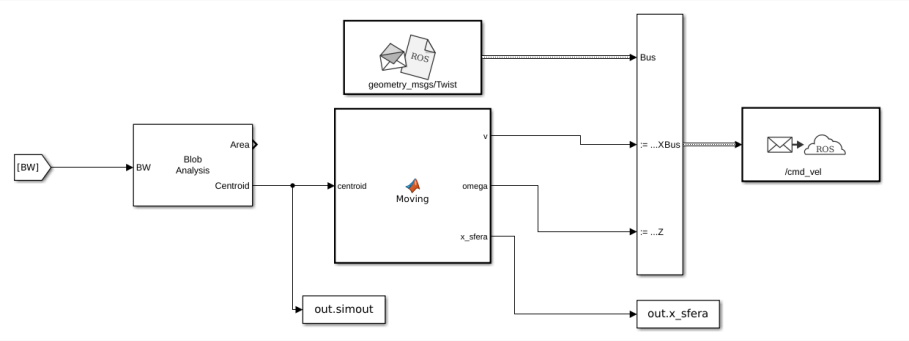

Figure 3: Simulink model for moving

Implementation

The image from the camera is converted from RGB to HSV. To detect the red sphere, upper and lower bounds are defined for the **Hue** and **Saturation**.

- Hue: 0 - 0.2
- Satuarion 0.7 - 1

The result is a black and white matrix of 0 and 1 where white pixels represent the red sphere.

Implementation

The position of the centroid is detected by the **Blob Analysis**:

1. **Input** $\rightarrow$ A binary image.
2. **Blob Detection** $\rightarrow$ Detects groups of connected white pixels.
3. **Property Analysis** $\rightarrow$ Measures characteristics such as:
   - Area
   - Centroid
   - Perimeter
   - Orientation

The centroid position is provided as input to the function *Moving* .

The function *Moving* controls the movement of the TurtleBot using a proportional controller, based on the position of the centroid detected by the camera.

- **Angular velocity** (ω): Determined proportionally to the error between the position of the centroid and the center of the image (x = 320.5).
- **Linear velocity** (v): If the centroid is centered, the robot moves at maximum speed. Otherwise, the linear velocity is reduced based on the error.

- The robot rotates (ω) and moves (v) proportionally to the error.
- If the error is **smaller than the dead zone**, it is set to zero and the robot does not rotate.
- If the centroid is not visible (e.g., if its coordinates are NaN), the robot stops ($v = 0, \omega = 0$).

The error is smoothed using a **filter** that calculates a weighted average between the current and previous error:

$$\text{error} = 0.7 \times \text{prev\_error} + 0.3 \times \text{error}$$

The filter reduces sudden changes, improving the stability and smoothness of the robot's movement.

Parameters

- $k_{p_\omega} = 0.002$, proportional gain for angular velocity.
- $k_{p_v} = 0.1$, proportional gain for linear velocity.
- angular dead zone $= 15$.
- $\omega_{\text{max}} = 0.3$.
- $v_{\text{max}} = 0.5$.

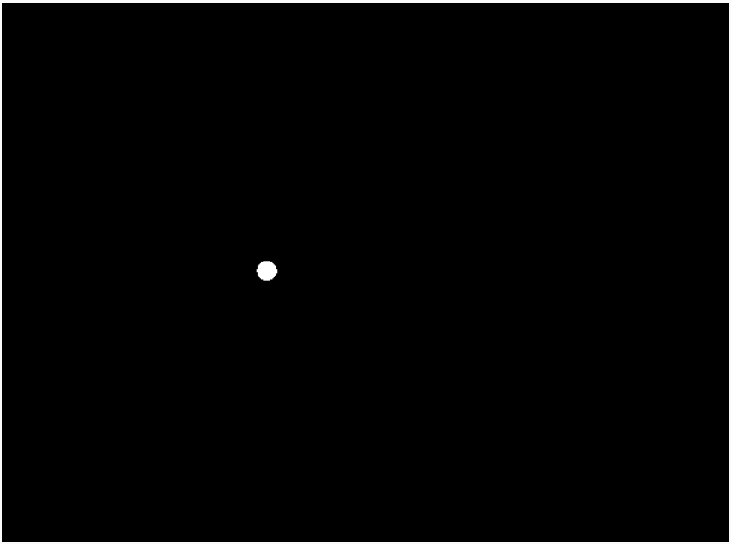

Figure 4: Red sphere detected

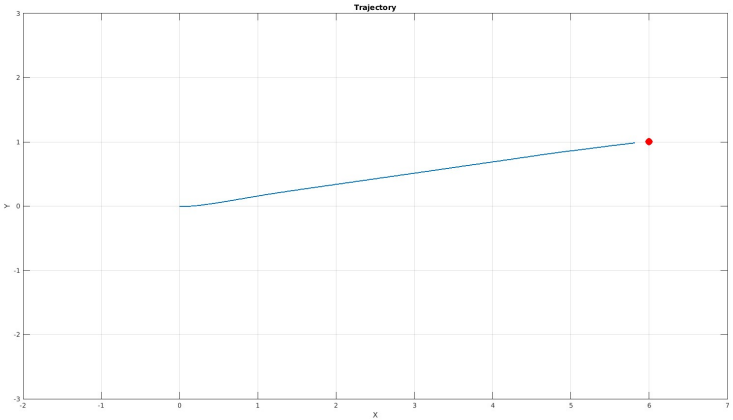

Figure 5: Trajectory

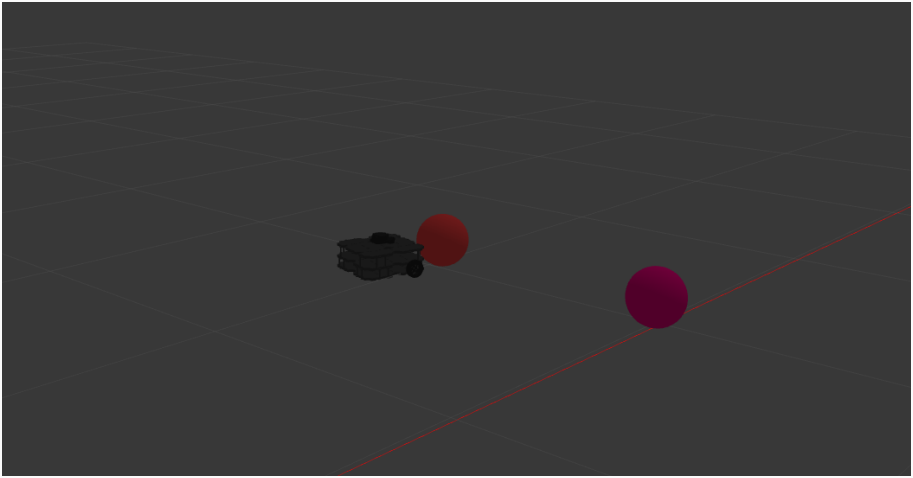

Figure 6: Red sphere reached

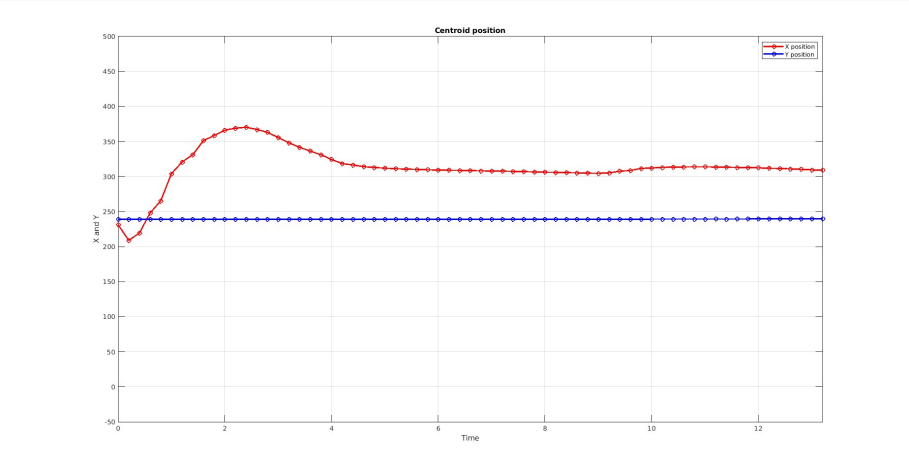

Figure 7: Centroidal position

Results

From the trajectory shown in Figure(5), the correct behavior of the TurtleBot is evident, as it detects the sphere and then continues at maximum speed until it touches it.
From the centroid position, the initial oscillations are visible, and then the centroid x remains at the center of the image (x = 320.5), while the y-coordinate is fixed at its central position (y = 240.5).

POLITECNICO
MILANO 1863