**Задача MPC**

1. **[In Python]** For the following kinematic model of mobile robot

write a code of implementing Acados numerical solver for finding the solution of the MPC problem for trajectory optimization of the robot with constraints on all input and output states and taking into account avoiding two static obstacles. The goal of the optimization trajectory is minimizing the difference between the trajectory of the robot and initial trajectory in addition to avoiding obstacles.

In progress use the following information:

* The cost function of minimizing the error between real and initial trajectory has the following general formula

where q and u are the output and input vectors of the trajectory of the robot, p and r are the output and input vector of the trajectory, , , are weights. The desired control inputs are

* For avoiding obstacles, use the following cost function:

where w is weight, and d is given as

where , , , are the coordinate of the robot and obstacles, and b = 0.4.

* The initial position of the robot is (x = 0, y=0 , = 0 )
* The obstacles are located at x\_obst1 = 3.5, y\_obst1 = 3.5, x\_obst2 = 1.1, y\_obst2 = 0.6
* The initial trajectory is between two points (0 , 0), (5 , 5)
* The desired linear velocity is 0.3 m/s
* The robot starts motion from velocity 0 m/sec and should stop motion at the final position.
* Length of prediction horizon N = 30.
* The orientation needed at the final position is 0.
* Use the following constraints on the output and input

-100 ≤ x ≤ 100

-100 ≤ y ≤ 100

0 ≤ v ≤ 1

-10 ≤ 𝜃 ≤ 10

-0.1 ≤ a ≤ 0.1

-0.3 ≤ w ≤ 0.3

The needed after implementation:

1- Understand your code

2- Figure for the initial path and the path of the robot with obstacles

3- Figures for the following input and output states of the robot (v , 𝜃 , a , w)

4- Time of finding solution

5- The value of the resulting cost

1. **[In C++]** Use the C-generated code of the previous task for writing ROS node which finds the same solution gotten in previous task with the same conditions

The needed after implementation:

1- Understand your code

2- Figure for the initial path and the path of the robot with obstacles

3- Figures for the following input and output states of the robot (v , 𝜃 , a , w)

4- Time of finding solution

5- The value of the resulting cost