Robotarium task 1

Tobias Hallingstad

September 19, 2020

1 Task undertaken

This task was about making a robot (in a simulation and in real life) move along a ploted corse. The corse was going to be like a infinity symbol.

2 Motion planning algorithm

Calculate the angle   To make the robot follow the given corse, i created the motion algorithm my self. I set out to create a "simple" way of finding the angle difrence from the robot to the next point of the corse. I get the angle by using this function in python:

```python
def calcAngle(rob, x2, y2):
    theta = math.atan2( (y2-rob[1]) , (x2-rob[0]) )

    turn = 3 * np.sin( rob[2] - theta )
    return -1*turn
```

Afther i have calculated the angle I then calculate a extra turning speed. This means that if the next point is straight forward the robot will do small corrections, and bigger corrections if the angle is large.

Calculate the speed   I also have a function to calculate the speed of the robot. This function is designed to increas the speed of the robot when the next point is far way and reduse the speed when the point gets close.

```python
def calcSpeed(rob, x2, y2):
    length = math.sqrt( math.pow(x2-rob[0], 2) + math.pow(y2-rob[1], 2) )

    return 5 * length if length > 0.2 else 0.2
```

This function is not perfect, the the robot will contantly change speed and it is no max speed defined. The function works fine in the simulation because of the short distances I am working with.

Extra functions I have writen some extra functions to make programming easyer. One function to set the speed and angle of the robot, one function to calculate if I am at the goal or not, and one to plot the corse.

3   Results simulation

4   Results at Robotarium