

# Robotarium task 1

Tobias Hallingstad

September 25, 2020

## 1 Task undertaken

This task was about making a robot (in a simulation and in real life) move along a plotted course. The course was going to be like a infinity symbol. I decided to write my own motion algorithm and write a function to draw the course from a few variables.

## 2 Motion planning algorithm

**Calculate the angle** To make the robot follow the given course, I created the motion algorithm myself. I set out to create a "simple" way of finding the angle difference from the robot to the next point of the course. I get the angle by using this function in python:

```
1 def calcAngle(rob, x2, y2):
2     theta = math.atan2( (y2-rob[1]) , (x2-rob[0]) )
3
4     turn = 3 * np.sin( rob[2] - theta )
5     return -1*turn
```

After I have calculated the angle, I then calculate an 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 increase the speed of the robot when the next point is far way and reduce the speed when the point gets close.

```
1 def calcSpeed(rob, x2, y2):
2     length = math.sqrt( math.pow(x2-rob[0], 2) +
3         math.pow( y2-rob[1], 2) )
4
5     return 5 * length if length > 0.2 else 0.2
```

This function is not perfect, the robot will constantly 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.

### 3 Results simulation

Running the simulation, the robot works as intended. When the angle between the robot and the goal is large, the robot turns fast, then slows down. When the robot arrives at the goal the robot drives to the next goal.

I did have some problems with my code before, I was talking with a friend and I implemented some code he used without thinking about if that code was going to work for me. This resulted in some time troubleshooting the code, but in the end my algorithm worked fine.

### 4 Results at Robotarium

At first, I had a while loop that looped to a count, I incremented the count when I hit the first goal. This way seems not to work. I then change the loop to a for loop and the robot started to move.

The robot is definitely not running perfect, it is overcorrecting some, and not always in the middle of the track, this is probably some of my code not being the totally best, and the robot not having the most accurate sensors. In general, the robot is doing what I have told it to do.

When we are going to work on a robot, we have then I think it is smart to test more and fine tune the variables or try to implement a more advanced algorithm.