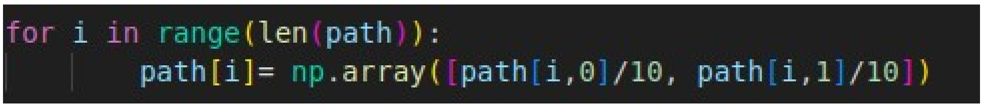
LAB 6 : Mobile Robot Navigation

Soulaïman Marsou

I didn’t achieve to make my program work. I give up this lab.

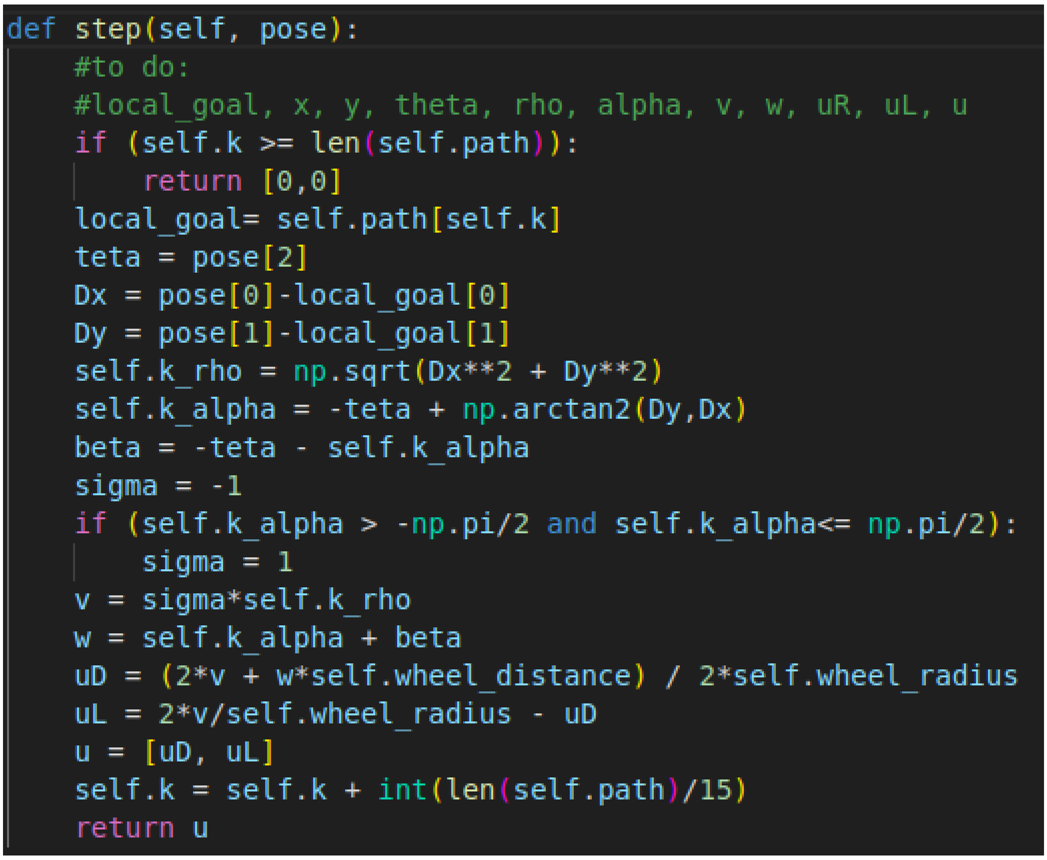
What I tried :

Firstly, I had to convert my path which was from a map of size 250x250.



```
for i in range(len(path)):
        path[i]= np.array([path[i,0]/10, path[i,1]/10])
```

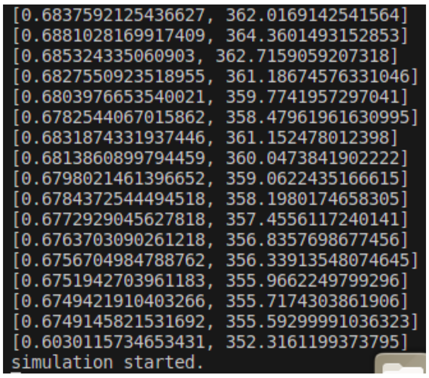
Secondly, I tried to implement the equations to obtain uD and uR.



```
def step(self, pose):
    #to do:
    #local_goal, x, y, theta, rho, alpha, v, w, uR, uL, u
    if (self.k >= len(self.path)):
        return [0,0]
    local_goal= self.path[self.k]
    teta = pose[2]
    Dx = pose[0]-local_goal[0]
    Dy = pose[1]-local_goal[1]
    self.k_rho = np.sqrt(Dx**2 + Dy**2)
    self.k_alpha = -teta + np.arctan2(Dy,Dx)
    beta = -teta - self.k_alpha
    sigma = -1
    if (self.k_alpha > -np.pi/2 and self.k_alpha<= np.pi/2):
        sigma = 1
    v = sigma*self.k_rho
    w = self.k_alpha + beta
    uD = (2*v + w*self.wheel_distance) / 2*self.wheel_radius
    uL = 2*v/self.wheel_radius - uD
    u = [uD, uL]
    self.k = self.k + int(len(self.path)/15)
    return u
```

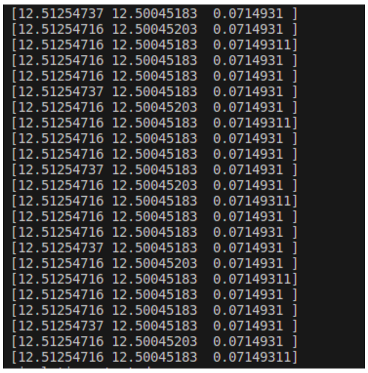
But as a result, my robot doesn’t follow the path, it doesn’t move at all.
Here is the output of u , and the output of the different pose during the animation.

U = [uD, uL]



```
[0.6837592125436627, 362.0169142541564]
[0.6881028169917409, 364.3601493152853]
[0.685324335060903, 362.7159059207318]
[0.6827550923518955, 361.18674576331046]
[0.6803976653540021, 359.7741957297041]
[0.6782544067015862, 358.47961961630995]
[0.6831874331937446, 361.152478012398]
[0.6813860899794459, 360.0473841902222]
[0.6798021461396652, 359.0622435166615]
[0.6784372544494518, 358.1980174658305]
[0.6772929045627818, 357.4556117240141]
[0.6763703090261218, 356.8357698677456]
[0.6756704984788762, 356.33913548074645]
[0.6751942703961183, 355.9662249799296]
[0.6749421910403266, 355.7174303861906]
[0.6749145821531692, 355.59299991036323]
[0.6030115734653431, 352.3161199373795]
simulation started.
```

Pose= [x , y, z]



```
[12.51254737 12.50045183  0.0714931 ]
[12.51254716 12.50045203  0.0714931 ]
[12.51254716 12.50045183  0.07149311]
[12.51254716 12.50045183  0.0714931 ]
[12.51254716 12.50045183  0.0714931 ]
[12.51254737 12.50045183  0.0714931 ]
[12.51254716 12.50045203  0.0714931 ]
[12.51254716 12.50045183  0.07149311]
[12.51254716 12.50045183  0.0714931 ]
[12.51254716 12.50045183  0.0714931 ]
[12.51254737 12.50045183  0.0714931 ]
[12.51254716 12.50045203  0.0714931 ]
[12.51254716 12.50045183  0.07149311]
[12.51254716 12.50045183  0.0714931 ]
[12.51254716 12.50045183  0.0714931 ]
[12.51254737 12.50045183  0.0714931 ]
[12.51254716 12.50045203  0.0714931 ]
[12.51254716 12.50045183  0.07149311]
[12.51254716 12.50045183  0.0714931 ]
[12.51254716 12.50045183  0.0714931 ]
[12.51254737 12.50045183  0.0714931 ]
[12.51254716 12.50045203  0.0714931 ]
[12.51254716 12.50045183  0.07149311]
```