CMP755 - Robotics - 2021

Homework #5

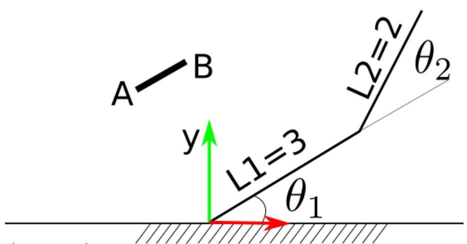
Due: 07.01.2022

You can submit your files as softcopy (scans, readable proper images, etc.) or bring as hardcopies to the class.

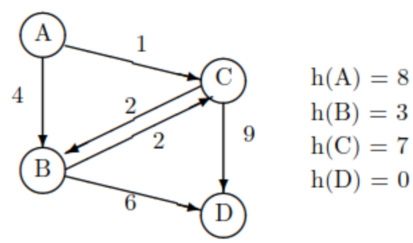
1. (20 pts) A 2-link manipulator is given in Figure. For this manipulator, $0 < \theta_1 < \pi$ and $0 < \theta_2 < 2\pi$. The line segment AB is an obstacle to be avoided. A=[-1,1]; B = [-0.8,1.2]. Draw the cspace of the manipulator. *Hint:* use the intersection function given below and find if the links collide with the obstacle at every angle (360x360 angles in total). One option: You can define cspace as a matrix and plot it with plt.imshow().

```python
def on_segment(p, q, r):
    #check if r lies on (p,q)
    if r[0] <= max(p[0], q[0]) and r[0] >= min(p[0], q[0]) and r[1] <= max(p[1], q[1]) and r[1] >= min(p[1], q[1]):
        return True
    return False

def orientation(p, q, r):
    #return 0/1/-1 for colinear/clockwise/counterclockwise
    val = ((q[1] - p[1]) * (r[0] - q[0])) - ((q[0] - p[0]) * (r[1] - q[1]))
    if val == 0 : return 0
    return 1 if val > 0 else -1


def intersects(seg1, seg2):
    #check if seg1 and seg2 intersect
    p1, q1 = seg1
    p2, q2 = seg2
    o1 = orientation(p1, q1, p2)
    #find all orientations
    o2 = orientation(p1, q1, q2)
    o3 = orientation(p2, q2, p1)
    o4 = orientation(p2, q2, q1)
    if o1 != o2 and o3 != o4:
        #check general case
        return True

    if o1 == 0 and on_segment(p1, q1, p2) : return True
```

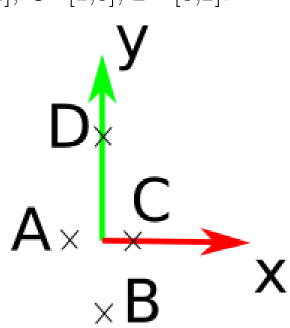
```python
    #check special cases
    if o2 == 0 and on_segment(p1, q1, q2) : return True
    if o3 == 0 and on_segment(p2, q2, p1) : return True
    if o4 == 0 and on_segment(p2, q2, q1) : return True
    return False


segment_one = ((-1, 0), (1, 0))
segment_two = ((0, 1), (0, -1))
print(intersects(segment_one, segment_two))
```



2. (20 pts) Trace the execution of the A* in the graph. Indicate the expanded path at each step, show the path from the start to the goal found by A*. A is the start and D is the goal node.

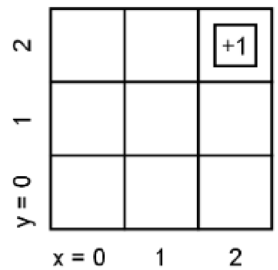


3. (20 pts) For the figure shown, draw the voronoi diagram for points A, B, C and D (they are point obstacles to be avoided). You can draw it by hand. A=[-1,0], B=[0,-2], C=[1,0], D=[0,2].



4. (20 pts) For the figure, consider it as an MDP problem. The transition rewards are 0 except for the exit reward. The reward for the exit transition

is +1. The discount factor is 0.5. Fill in the optimal values for the grid.



5. (20 pts)

```
t = [1,10,22,35,40,51,59,72,85,90,100]
y = [0.18,0.22,0.29,0.39,0.48,0.16,0.56,0.61,0.68,0.75,0.81]
```

t denotes the time and y denotes the measured position. We consider a constant velocity system.

Measurement error $r_n = 0.05$, initial estimate error $p_{0,0} = 0.3$. Initial estimate $x_{0,0} = 0.1$, and we take uncertainty noise $q = 0.05$. Use 1D Kalman filter to estimate the true values and draw the plot measured values vs Kalman filter values. Write an iterative script to compute. Also draw the Kalman gain values.