Parsa Yadollahi
COMP 417 - Assignment 1
260869949

1. By following the equation $x(t+1)$ from the equation assuming that:

R($\omega\delta t$) = $\begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$, x(t) = $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$.

We get the following result:

$$x(t+1) = \begin{bmatrix} xcos(wst) - ysin(wst)) \\ xsin(wst) + ycos(wst) \\ \theta + wst \end{bmatrix}$$

We can deduct that $R()$ is simply a *2x2* matrix and that the last column and row is to preserve the rotation rate. When the left and right wheel of the robot rotate with different speeds, the robot rotates with respect to some global frame of reference denoted ICC. We denote, as in the question, the rotation rate as ω radians/second and δt as time in seconds. Which implies that:

$$\phi' = \omega\delta t + \phi$$

Now consider a *2x2* matrix of rotation. Let *v* be a vector starting at the origin (x,y) and the new position (x',y'). This vector can computed using a 2D matrix around ϕ (ICC) to produce (x',y') with angular velocity ω for δt seconds. After rotation, we can deduct that

x_r= $\begin{bmatrix} xcos(\theta) \\ xsin(\theta) \end{bmatrix}$ and y_r = $\begin{bmatrix} -ysin(\theta) \\ ycos(\theta) \end{bmatrix}$

Meaning that the final vector has coords:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} xcos(\theta)) - ysin(\theta) \\ xsin(\theta) + ycos(\theta) \end{bmatrix} = \begin{bmatrix} cos(\theta)) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Thus R($\omega\delta t$) = $\begin{bmatrix} cos(\theta)) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$

2.
   a. <u>State space:</u> All possible permutations of the first and second link in the exception that the two arms do not intercept.

   $$X = [Gpx,\ Gpy,\ G\theta 1,\ G\theta 2]$$

   Where θ1ε[-π/2,π/2] and θ2ε[-π,π]

b. <u>Action space:</u> Rotating the two arms. I.e when the angle of the arm 1 and arm 2 change as well as the position of the end-effector (x,y). So when $U = [Cx, Cy, C\theta1, C\theta2]$ changes.

c. Let a1 be the length of the robots arm that is attached to the table and a2 be the lengths of the robots arms not attached to the tables

   i. Let x = x_1+x_2 where x_1 is the length of the horizontal length of the a1 and x_2 be the horizontal length of a2.
   Let y = y_1+y_2 where y_1 is the length of the vertical length of the a1 and y_2 be the vertical length of a2.
   Let $\Theta1$ be the angle from the horizontal axis to the first arm and $\Theta2$ be the angle between the first arm and the second arm.
   Let $\Theta3$ be the angle between the first arm and 3pi/4 and $\Theta4$ be the angle between the positive horizontal axis and the second arm.
   We get that:

$$x1 = a1cos(\Theta1)\ and\ y1 = a1sin(\Theta1)$$
$$X2 = a2cos(\Theta4)\ and\ y2 = a2sin(\Theta4)$$

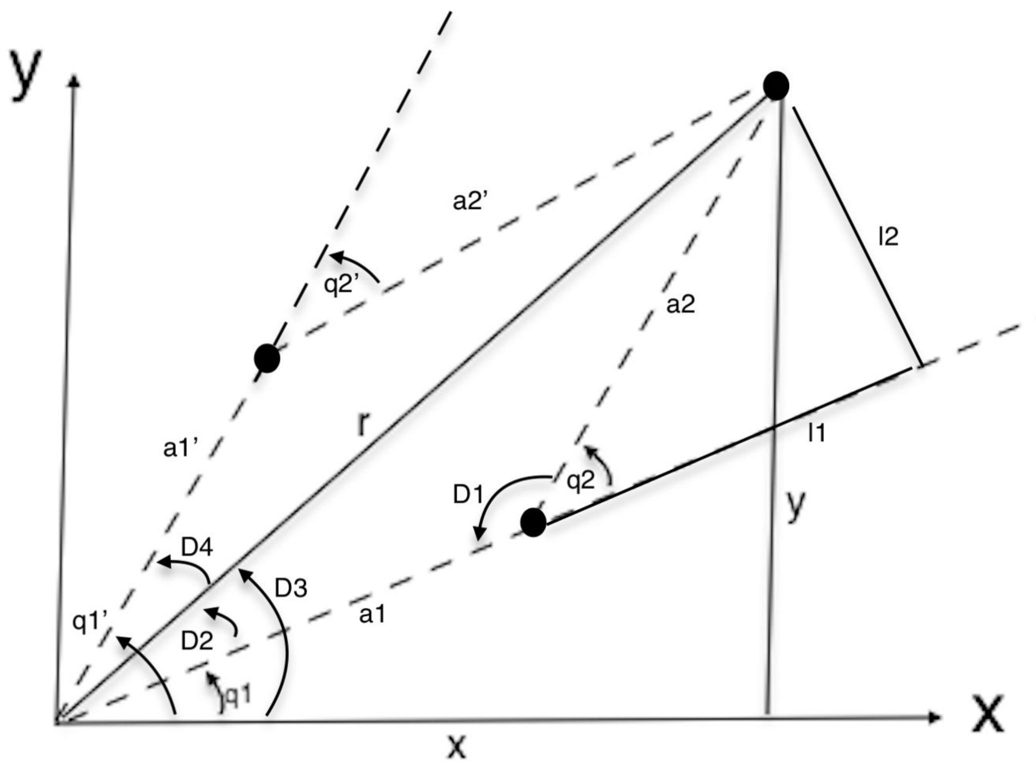
   We know that:

$$\Theta2 = \Theta3 + \Theta4 + pi/2\ and\ \Theta3 = arcsin(x1/a1)$$
$$\Rightarrow \Theta4 = \Theta2 - \Theta3 - pi/2 = \Theta2 - arcsin(x1/a1) - pi/2$$
$$\Rightarrow x2 = a2cos(\Theta2 - arcsin(x1/a1) - pi/2)$$

   and $y2 = a2sin(\Theta2 - arcsin(x1/a1) - pi/2)$

   forward kine($\Theta1$, $\Theta2$) = $((a1cos(\Theta1 + a2cos(\Theta2 - arcsin(x1/a1) - pi/2)),$
   $(a1sin(\Theta1) + a2sin(\Theta2 - arcsin(x1/a1) - pi/2)))$

d. There will be two solutions to the following problem. The first will be for when the first arm goes beneath 45 degrees and the second will be for when the first arm goes beyond 45 degrees. Ultimately, these two answers return the same (x, y) location. For simplification and ease of understanding, I have drawn a diagram with labels for each of the relevant angles and lines. Note that a1 and a2 are both of length 1 but for the sake of general solutions where they might not be 1, I included them in the equations.



i. Since we are given x and y coords, we can calculate r=sqrt(x^2+y^2). Now let's calculate D1 using the cosine rule. We get

$$D1 = (a1^2 + a2^2 - r^2) / 2(a1 * a2)$$

Next we are going to find q2 using the same cosine rule:

$$\cos(q2) = -\cos(D1)$$

Now we're going to find *l1* and *l2* which will create a right angle triangle with a1 and the yellow ball. We get that

$$l1 = a2cos(q2)$$
$$l2 = a2sin(q2)$$

By doing this we can calculate D2 since we have a right angle triangle where the base is *a1 + l1* (let's denote this as l3) and the side being *l2* we get:

$$D2 = arctan(l2 / l3)$$

Now we can calculate D3 and by doing so deduct D3 from D2 to get q1 which is the first angle of the joint.

$$D3 = arctan(y/x)$$
$$q1 = D3 - D2$$

To put it all together, we get the final equations for q1 and q2 being:

$$q1 = arctan(y/x) - arctan(l2 / l3)$$
$$q2 = arccos(x^2 + y^2 - r^2) / 2(a1 * a2)$$

ii. Now let's calculate this as if the first arm was above 45 degrees. We get similar solutions. Let's first calculate D4:

$$D4 = l2sin(q2') / (a1 + a2cos(q2'))$$

Notice here that this is the same equation we used above to find D2, so we can refactor this equation to:

$$D4 = l2\ /\ l3$$

We can calculate q1 with this information:

$$q1' = D3 + D4$$

To put it all together, we get the final equations for q1 and q2 being:

$$q1' = arctan(y/x) + arctan(l2/l3)$$

$$q2' = -arccos((r^2 - a1'^2 - a2'^2)/2(a1' * a2')$$

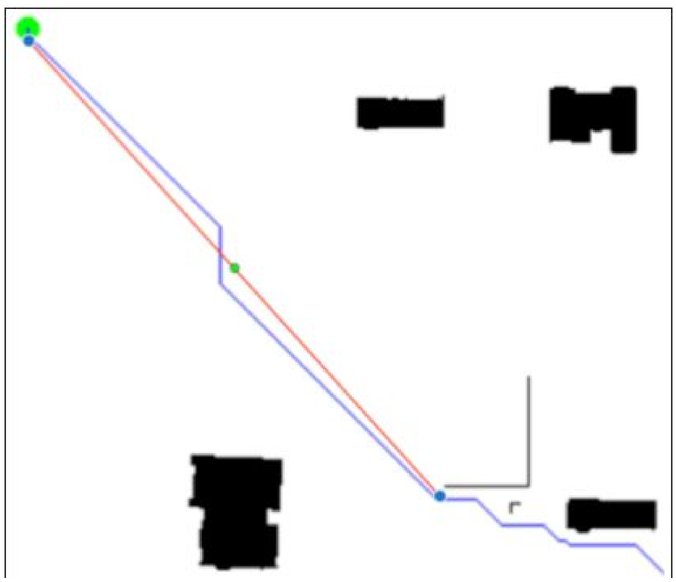
The pseudo code for the equations above is:

```
func lawOfCosines(a, b, c) {
      return arccos((a*a + b*b - c*c) / (2 * a * b))
}
func distance(x, y) {
      return Sqrt(x*x + y*y)
}
func q2(x,y,a1,a2) {
    top = distance(x,y)*distance(x,y) - a1*a1 - a2*a2
    bottom = 2*a1*a2
    return arccos(top/bottom)
}
func q1(x,y,a1,a2) {
    left = arctan(y/x)
    top = a2sin(q2)
    bottom = a1+a2cos(q2)
    right = arctan(top/bottom)
    total = left - right
    return total
}
funct q2_prime(x,y,a1_prime, a2_prime){
    top = distance(x,y)*distance(x,y) - a2_prime*a2_prime - a1_prime*a1_prime
    bottom = 2*a1*a2
    return -arccos(top/bottom)
}

func q1_prime(x,y,a1_prime, a2_prime) {
    left = arctan(y/x)
    top = a2_primesin(q2_prime)
    bottom = a1_prime+a2_prime*cos(q2_prime)
    right = arctan(top/bottom)
    total = left + right
    return total
}
```

3.

a. No, it cannot be the bug #1 algorithm since before the robot hit an obstacle, it changes directions.
b. No, it cannot be the bug #1 algorithm either for the same reason as a) since the robot changes directions before it hits an obstacle.
c. Yes, since Dijkstras is like A* meaning it is complete but is not always optimal. This path is not optimal but the robot has found the goal (the red dot). The only time the path would be optimal is if the graph had no weight, yet we cannot conclude that with the given information.
d. No, A* is not a possibility since A* returns an optimal path, given that the heuristic is admissible. We do not have any information about the heuristic so I will deduct that it is. If it is not admissible then we can conclude that the path is a possibility and that it would behave like dijkstra's. Yet this is not an optimal path. For example, take the following path:

   The path highlighted in red would be optimal (A* path) whereas the path in blue (in the question) is not. And by this, we can conclude that the robot did not take the A* algorithm.
e. Yes, RRT could be a possibility since it is randomized. Meaning that the robot could have chosen the path picked in the question out of all the possible path configurations to get to the goal.