SOLUTIONS: QUIZ 2

CONTROL OF MOBILE ROBOTS

1

Consider a unicycle robot with dynamics

$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\dot{\phi} = \omega$$

If the two inputs (v, ω) are constant what will the robot do?

SOLUTION

There are two ways of approaching this problem – one intuitive and one mathematical. The intuitive way is to picture yourself in a car, driving at a constant speed and with the steering wheel at a constant angle. You are obviously driving in a circle. The more you turn (bigger ω) the smaller the circle, i.e., the smaller the radius. Similarly, the faster you drive, the bigger the radius. So, based on these observations, the only answer that is possible is that the radius is v/ω.

The "mathy" way of approaching this is to assume that the unicycle is driving along a circle with mid-point (x_0, y_0) and radius r already, i.e., that

$$x(t) = x_0 + r\cos(\theta(t))$$
$$y(t) = y_0 + r\sin(\theta(t)),$$

where $\theta(t)$ tells us where on the circle the unicycle is at time t. Simple geometry immediately tells us that $\theta = \phi - \pi/2$, where ϕ is the heading of the unicycle. As such, we have (from a standard trigonometric rule) that

$$x = x_0 + r\cos(\phi - \pi/2) = x_0 + r\sin(\phi) \;\Rightarrow\; \dot{x} = r\dot{\phi}\cos(\phi) = r\omega\cos(\phi).$$

But, we already know that $\dot{x} = v\cos(\phi)$ and, as such,

$$v = r\omega \;\Rightarrow\; r = \frac{v}{\omega}.$$

So, the answer is: Circular arc with radius v/ω.

2

A differential-drive robot is equipped with a wheel encoder with 10 "ticks" per revolution, wheel radius of 0.1m, and the two wheels are 0.2m apart. The robot starts at the origin (position and orientation is 0) and, during a short time interval of 0.5s a total of 5 ticks were recorded for the right wheel and 3 ticks for the left. Where is the robot approximately located after 0.5s?

SOLUTION

We have seen that the pertinent equation is

$$D_{r,\ell} = 2\pi R \frac{\Delta \text{tick}_{r,\ell}}{N},$$

which gives that the distance traveled by the two wheels is

$$D_r = 2\pi 0.1 \frac{5}{10} = 0.3142, \quad D_\ell = 2\pi 0.1 \frac{3}{10} = 0.1885.$$

This, in turn, tells us that the central arc has traveled

$$D_c = \frac{D_\ell + D_r}{2} = 0.2513,$$

i.e.,

$$\begin{aligned} x_{new} &= x(0) + D_c \cos(\phi(0)) = 0 + 0.2513 \cos(0) = 0.2513 \approx 0.3 \\ y_{new} &= y(0) + D_c \sin(\phi(0)) = 0 + 0.2513 \sin(0) = 0 \\ \phi_{new} &= \phi(0) + \frac{D_r - D_\ell}{L} = 0 + \frac{0.3142 - 0.1885}{0.2} = 0.6283 \approx 0.6. \end{aligned}$$

As, such, we get the answer that $x \approx 0.3,\ y \approx 0,\ \phi \approx 0.6$.

3

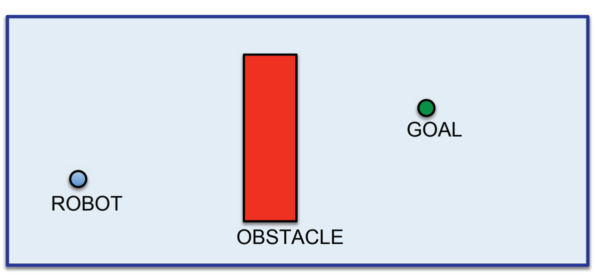
Same question as Question 2, with the small difference that the time interval is now 5 minutes and the right tick count is 40 and the left tick count is 65.

SOLUTION

The odometry equation assumes that the robot has traveled along a circular arc. This is a good approximation over small time intervals but there is absolutely no reason to believe that during a 5 minute interval the robot has somehow only driven along a particular circular arc. Moreover, the drift will without a doubt have been rather horrendous so we cannot trust the equations over such a long time interval. As such, the answer is: Impossible to say – not enough information provided.

4

Let the go-to-goal behavior drives a robot straight towards the goal location and the obstacle-avoidance behavior drives perpendicularly away from the closest obstacle point. Moreover, assume that the system blends the two behaviors in the following way: it uses only go-to-goal when the obstacle is far away and only obstacle-avoidance when the obstacle is really close. What would happen to this robot if it was to negotiate the environment shown below?



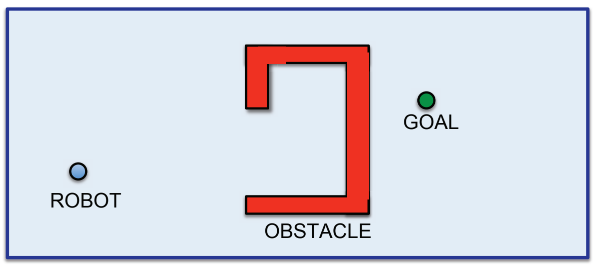
SOLUTION

The robot will approach the obstacle. Once it gets close enough, the obstacle-avoidance behavior will kick in, pushing the robot straight away from the obstacle. The combined, blended behavior of the robot will be to have the robot move “upwards” along the obstacle until the go-to-goal and the avoid-obstacle behaviors point in completely different directions, causing the robot to move neither up nor down, i.e., it effectively gets stuck.

As such, the answer is: The robot will get stuck and not be able to negotiate the obstacle.

5

Now, consider the labyrinth below. Assume the robot is indeed equipped with a well-designed go-to-goal behavior. Which of the following obstacle-avoidance behaviors would not be able to lead to a successful negotiation of the labyrinth.



SOLUTION

Here we have to go through the different choices. And, as nothing was specified on how/when the different behaviors become active, we have to focus on which behavior would have no possible way of solving the labyrinth problem.

Follow-Obstacle-Clockwise: This means that the robot always moves in such a way that it has the obstacle (or at least the closest point on the obstacle) to its right. By doing so, the robot can indeed make it out of the labyrinth by going up and actually getting around the obstacle. Assuming that we have a reasonable way of deciding when to “let go”, this behavior (if designed correctly) would work.

Follow-Obstacle-Counter-Clockwise: Same argument as before. Only difference is that the obstacle will be to the left of the robot, and it will initially move “downwards” rather than up when contact is made with the obstacle. Conclusion: his behavior (if designed correctly) would work.

Always-Keep-Obstacle-To-The-Left-Of-Robot: This is the same behavior as Follow-Obstacle-Counter-Clockwise. Conclusion: his behavior (if designed correctly) would work.

Follow-Obstacle-In-Direction-That-Moves-Robot-Closer-To-Goal: If we use this behavior, we would, after making contact with the obstacle, start moving up to a point when the obstacle surface is orthogonal to the direction from the robot to the goal at which point the robot would no longer move. In essence, it is the same problem as the problem with the behavior in Question **4**. Conclusion: This behavior will not successfully negotiate the labyrinth.

Answer: The behavior that will certainly fail is “Follow-Obstacle-In-Direction-That-Moves-Robot-Closer-To-Goal”.