CSCI 3302 - Introduction to Robotics

15 September 2017

Nikolaus Correll

Cora Schneck, Kylee Bennett, Max Messenger Bouricius, Anthony Hauger, Douglas Allen

---

Lab 3: Sparki Inverse Kinematics

[Github Link to Live Repo]

1. *Use the forward kinematic relationship for a differential wheel platform from the book (x_r' and θ_r' as a function of left and right wheel speed) to calculate its inverse, that is left and right wheel speeds given x_r' and θ_r'.*

Let a,b,c be the controller gains of the robot. In this case, $x_r' = (r * \varphi_l)/2 + (r * \varphi_r)/2$. Alternatively, $x_r' = a * \rho$.

θ_r' is equal to $(r * \varphi_r)/d - (r * \varphi_l)/2$, or $b * \alpha + c * \eta$. Setting the two x_r' equal to one another and substituting for the two θ_r', one can conclude that $\varphi_l = (2(x_r'/r) - (\theta * d))/2$ and $\varphi_r = (2(x_r'/r) - (\theta * d))/2$.

2. *What happens if you decrease your constants (here 0.1)? What happens if you increase them? What happens if they get too big? What is the role of the 0.01*(THETA-theta_r) term?*

These equations are as follows:

- $X_i = X_i + cos(\theta i) * X_r' * 0.1$
- $Y_i = Y_i + sin(\theta i) * X_r' * 0.1$
- $\theta i = \theta i + \theta r' * 0.1$

If one decreases the constants, the robot will turn more sharply. If the constants become too large, the robot will spiral around its goal multiple times before (if lucky) eventually reaching the desired location and orientation. The role of 0.01*(THETA-theta_r) is to guide the robot in the desired orientation as it progresses towards its goal.

3. Assume the robot is able to turn away from obstacles while executing your controller. What happens if you place a u-shaped obstacle between the robot and its desired location?

When one places a U-shaped obstacle between the robot and its desired location, Sparki will execute the following actions:

1. Detect the wall and back up, increasing its distance from the goal
2. Turn (in our code, to the right), increasing its alpha value
3. If poorly implemented, Sparki will become trapped in this U-shaped prison

The distance and alpha value will increase more based on where Sparki is in relation to the obstacle (i.e. if it were facing the arc of the u, it would have to rotate further to get around the object than if were the uprights of the u).

A more structured approach would be to implement multiple goal positions with ranked priorities, so one can momentarily override the original goal position in order to inject a new desired position (e.g. around the U-shaped obstacle) until Sparki is confident that it can return to progressing towards the original goal.