Introduction to Robotics

Homework 3
Due Date: Wednesday, 20 November 2024

Theory questions

1) In class we discussed the differential flatness method and the A* method for trajectory generation.
   a) Please discuss which method requires more computational resources and why?
   b) For obstacle avoidance which method would you use and why?
   c) For movement in obstacle free 3d space, which method would you use and why? What are the drawbacks of the method you didn't choose?

2) **Pose stabilization:** Consider Example 3.2.1 in the class notes. Show that $\dot{V} < 0$ for the chosen control inputs (Equation 3.11 in the notes). Show all the steps in the math.

3) Consider the following nonlinear system

$$\dot{x_1} = -x_1 + x_2^3$$
$$\dot{x_2} = -x_2 + u$$

Using the Lyapunov function $V = \frac{1}{2}x_1^2 + \frac{1}{4}x_2^4$, find the control u that will stabilize this system.

4) Describe step by step (in words) the A* grid-based search algorithm. What are cons of this method?

5) Describe step by step (in words) the Probabilistic Road Map (PRM) sampling-based motion planning method. What are the cons of this method?

6) Describe step by step (in words) the Rapidly-exploring Random Tree (RRT) sampling-based motion planning method. What are the cons of this method?

Computational problem

7) Trajectory tracking using a closed loop controller

Consider the extended unicycle model of HW 2 problem 2 and 3.

In HW 2 problem 3, we saw that in the presence of noise and disturbances the open loop controller breaks down. For our robot to correctly track the trajectory in presence of noise and disturbances we need to use a "closed loop controller". That is, the controller will compare the current position with the desired position and generate the input accordingly.

We can design a trajectory tracking closed loop controller in the following way. Differentiating the velocities $\dot{x}, \dot{y}$ we obtained the following equations,

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta\,(t) & -V(t)\sin\theta\,(t) \\ \sin\theta\,(t) & V(t)\cos\theta(t) \end{bmatrix} \begin{bmatrix} a(t) \\ \omega(t) \end{bmatrix}$$

A trajectory tracking controller can be defined as,

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \end{bmatrix} = \begin{bmatrix} \ddot{x}_d(t) - k_{px}(x - x_d) - k_{dx}(\dot{x} - \dot{x}_d) \\ \ddot{y}_d(t) - k_{py}(y - y_d) - k_{dy}(\dot{y} - \dot{y}_d) \end{bmatrix}$$

In class we showed the above equation is stable (i.e., e(t) -> 0 as t-> ∞, where $e(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix}$) for certain choices of K_p and K_d.

So, based on the two equations above, we can write,

$$\begin{bmatrix} \cos\theta_d\,(t) & -\dot{y}_d \\ \sin\theta_d\,(t) & \dot{x}_d \end{bmatrix} \begin{bmatrix} a(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \ddot{x}_d(t) - k_{px}(x - x_d) - k_{dx}(\dot{x} - \dot{x}_d) \\ \ddot{y}_d(t) - k_{py}(y - y_d) - k_{dy}(\dot{y} - \dot{y}_d) \end{bmatrix}$$

Then a(t) and ω(t) can be calculated by taking the matrix inverse in the equation above. Here, $x, y, \dot{x}, \dot{y}$ are obtained from integrating the unicycle equations (you can think of these as sensor measurements). $x_d, y_d, \theta_d, \dot{x}_d, \dot{y}_d, \ddot{x}_d, \ddot{y}_d$ are the desired values obtained from the differentially flat trajectory, k_{px}, k_{py} are proportional gains and k_{dx}, k_{dy} are derivative gains.

Refer to Example 3.1.1 in the course notes for more details.

a) Implement the closed loop controller defined above to obtain the controls a(t) and ω(t). Use these to integrate the unicycle equations. Set $K_p = 1$ and $k_d = 2$. Inject noise in the V and θ equations as described in HW 2 problem 3. The robot should be able to track the trajectory successfully. Submit your code and screenshots.

b) Repeat part (a) for $K_p = 4$ and $k_d = 4$.