# Theory questions 1) In class we discussed the diff...



## Question

(0)

**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?

## Expert Answer

This solution was written by a subject matter expert. It's designed to help students like you learn core concepts.

### Step-by-step

1st step

All steps

Answer only

Step 1/3

## Solution:

1 ) . In class, you discussed the Differential Flatness method and the A* method for trajectory generation.

## Step-1:

### a) Computational Resources:

- Differential Flatness Method: Typically requires fewer computational resources because it focuses on finding a trajectory that satisfies system dynamics, leading to smoother and more efficient paths. It doesn't involve an extensive search process.

- A* Method: Generally requires more computational resources, especially in complex environments, as it searches a discrete grid or graph representation of the environment for an optimal path.

### b) Obstacle Avoidance:

For obstacle avoidance, the A* method is preferred because it is specifically designed to find collision-free paths in environments with obstacles. It considers discrete obstacle representations and efficiently avoids collisions.

### c) Movement in Obstacle-Free 3D Space:

For movement in an obstacle-free 3D space, the Differential Flatness method is more appropriate. It generates smooth and feasible trajectories that satisfy system constraints without the need for complex pathfinding algorithms.

### Drawbacks of not choosing each method:

- Drawbacks of using A* in obstacle-free spaces: A* may introduce unnecessary complexity and computational overhead when obstacles are absent.

- Drawbacks of using Differential Flatness in obstacle-filled spaces: Differential Flatness doesn't handle obstacles directly, so it may lead to collisions or require additional collision avoidance strategies.

Explanation:

A* may introduce unnecessary complexity and computational overhead when obstacles are absent.

Step 2/3

## Step-2:

### 2. Pose Stabilization:

To show that V < 0 for the chosen control inputs ( u1andu2 ) in Equation 3.11, you would need to compute the time derivative of the Lyapunov function V and demonstrate that it is negative.

### 3. Nonlinear System Stabilization:

- To stabilize the system $x_1 = -x_1 + x_2^3$ and $x_2 = -x_2 + u$, you can use the Lyapunov function $V = \frac{1}{2}x_1^2 + \frac{1}{4}x_2^4$.
- Find the control u that stabilizes the system by computing V and setting it to be negative:

  Compute $\dot{V} = dt / dV = (\partial V / \partial x_1) x_1 + (\partial V / \partial x_2) x_2$
  Chosen u such that V < 0 for all $(x_1, x_2)$ in the state space.

### 4. A* Grid-Based Search Algorithm:

- A* is a popular pathfinding algorithm for discrete grids or graphs.
- Initializing opening and closing sets.
- Add the starting node to the open set.
- While the set is not empty:
- Select the node with the lowest cost in the open set.
- Expand it by considering its neighbors.
- Calculate costs (e.g., g-cost, h-cost) for each neighbor.
- Update the open set with the neighbors.
- when the goal node is reached or no path is found, terminate.

**Cons:** A* can be computationally expensive for large grids, especially in high-dimensional spaces. It doesn't handle dynamic environments well and requires a good heuristic for efficient performance.

Step 3/3

## Step-3:

### 5. Probabilistic Road Map (PRM) Sampling-Based Motion Planning:

- Sample random configurations (nodes) in the configuration space.
- Connect nodes to form a graph based on collision checking.
- Find a path by searching the graph using algorithms like Dijkstra's or A*.

**Cons:** PRM's quality depends on the sampling density, and it may not guarantee the optimal path. It can be computationally expensive when the environment is cluttered.

### 6. Rapidly-Exploring Random Tree (RRT) Sampling-Based Motion Planning:

- Initialize a tree with the start configuration.
- Sample a random configuration and find the nearest node in the tree.
- Extend the tree towards the sampled configuration.
- repeat steps 2 and 3 until reaching the goal or a specified time limit.

**Cons:** RRT may not find the optimal path. They can be inefficient in cluttered environments and might require tuning to balance exploration and exploitation.



## Final Answer:

PRM's quality depends on the sampling density, and it may not guarantee the optimal path. It can be computationally expensive when the environment is cluttered.

## My Textbook Solutions

+ Add textbook