Robot Design Lab WiSe 2020 Shiyao Zhang
Group 7

Worksheet 4

Solutions:
Due Date: 11.01.2021

Exercise 1.

Compile a list of 3 path-planning algorithms and their pros and cons.

Solution

A*-algorithms:
Pros:
Could generate the shortest path.
Relatively fast and simple.
Cons:
Cannot search in every angle.
Every cost and node should be fixed. This algorithm cannnot adjust the dynanime area.
A new bounds or nodes cannot be calculated.

Dijkstra-algorithms:
Pros:
Could generate the shortest path.
Relatively simple.
Cons:
Spend too much memory to record all relative nodes.
Not efficient.

Bestfirst-algorithms:
Pros:
Efficient.
Cons:
It's possible to generate a nonoptimal path, when there are obstacles between the start and the end.
It's possible that it can't find a solution, when the algorithms get into a loop.

Create a map with a lot of nodes for the robot to plan a very close path around obstacles, assuming we use the A*-algorithm for path planning. Why might this be a problem, especially in a dynamically changing environment?

Solution

When we us the A*-algorithm, we need to calculate the costs. The one is the distance from starting node, the other one is distance from end node. We have to find the minimum cost from the starting to the end. Every step we calculate all nodes surround the current node. When the environment is dynamic, we couldn't predicate which node would be a obstacle in next time. Sometimes it causes the robot move too closely to the obstacles. All new bounds like obstacles or nodes could not be calculated. The default is there is no path found.

Exercise 2.

Give some example of when you would use Forward Kinematics and when Inverse Kinematics algorithms.

Solution

From the insight of the input and output:
When you have a set of joint angles as input und would determine the postion and orientation of robotic also the workspace of it, you would use the Forward Kinematics.
In the other hand, when you have the position and orientation of end-effector, you would use the Inverse Kinematics to determine all possible set of joint angles.
From the insight of possible solutions:
For the Inverse Kinematic we could have multiple solutions. You may have not a unique configuration, especially in redundancy. So first of all you should define a prefered configuration, so that it could make the robot to move in a desired way. Otherweise forward Kinematics could give a unique workspace for a robot like a robotics arm. We could know the final coordinate of the gripper of the robot, suppose we have get all joint's angle.

Which of the two(Inverse or Forward Kinematics) is usually more difficult to compute for a serial robot (composed of an open serial chain of joints)?

A serial robot has complicated geometry, inverse kinematics is more difficult to compute. Because you have to get the information of workspace, but for a serial robot the workspace is not unique. Therefore it's hard to solve how to get all possible workspaec.

And for a parallel robot (closed chain of joints)?

A parallel robos are usually based on kinematic constraints to simple serial mechanisms. Therefore, forward kinematics is more difficult, becuase it needs to take into account of constraints und all joints of it. It's hard to get all information about it.

Feedback

How much time did you spendon doing this sheet per person

10 hours.

Was is too easy, easy, ok, hard, too hard?

Compare the different algorithms is hard.
Find the examples about inverse and forward kinematics is easier than others.

What additional resources(blogs, papers, books, tutorials, etc) did you use?

`http://web.eecs.utk.edu/~leparker/Courses/CS594-fall08/Lectures/Oct-7-Path-Planning-I`
`pdf`
`https://qiao.github.io/PathFinding.js/visual/`
`https://www.youtube.com/watch?v=-L-WgKMFuhE`
`http://cats-fs.rpi.edu/~wenj/ECSE448F08/lec23.pdf`
`https://www.mathworks.com/help/symbolic/derive-and-apply-inverse-kinematics-to-robot-`
`html`
`https://www.researchgate.net/post/What-is-the-difference-between-forward-kinematics-a`

Any other issue?