Hybrid A Star

Alex Eagles

Contents

Table of Figures

Introduction

Hybrid A Star is an algorithm like the traditional **A Star** that uses the **shortest path** from the starting point to the current point and **heuristics function** that sometimes be **Manhattan** distance or **Euclidean** distance. Still, **Hybrid A star** rises when we want to simulate real-time autonomous cars or planes, and we want to consider the **kinematics** of these vehicles like if they can take this path or not because of their current angle or they're too big or all these proprieties which make the traditional made using **A star** is unreachable. **A Star algorithm** works well in a grid domain with many applications including computer games. However, this guided search on a grid has a limitation when we try to apply for nonholonomic systems such as a car. Although the output of A* can be globally optimal in the grid, a resultant path can be inefficient when it is tracked by the dynamics.

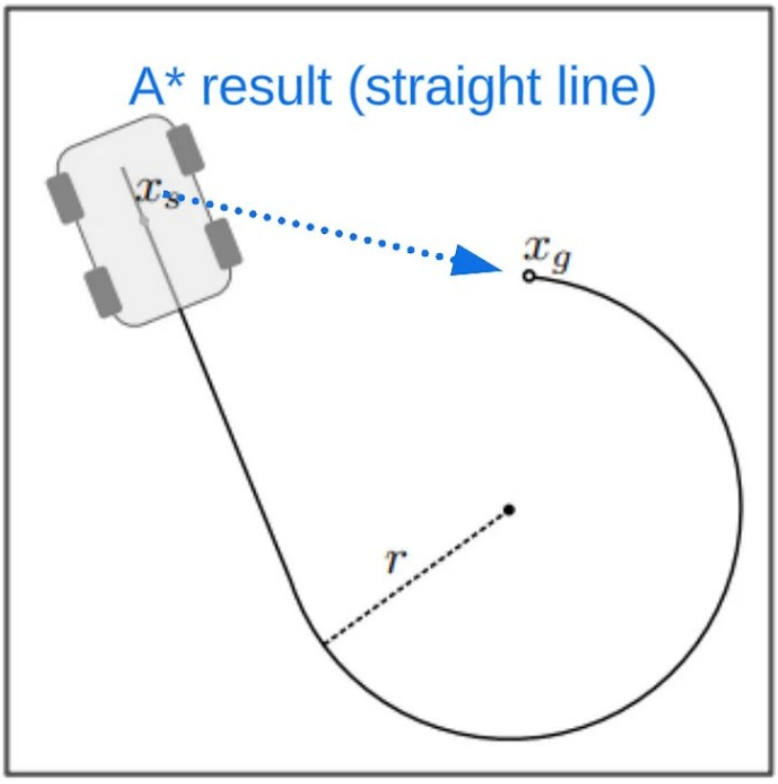


Figure 1 A Star Drawing Straight Line

Main Characteristics of Hybrid A Star

Objective

In the normal **A Star searching** algorithm in the grid, we're only trying to find the shortest path, so we're not considering the angle of rotation, which will lead us to an unreachable path sometimes, In **Hybrid A Star**, we'll expand the domain a little bit and we'll introduce the angle with us, but including this, we have to change the way we're calculating the distance, and we can use **Dubins Path**.

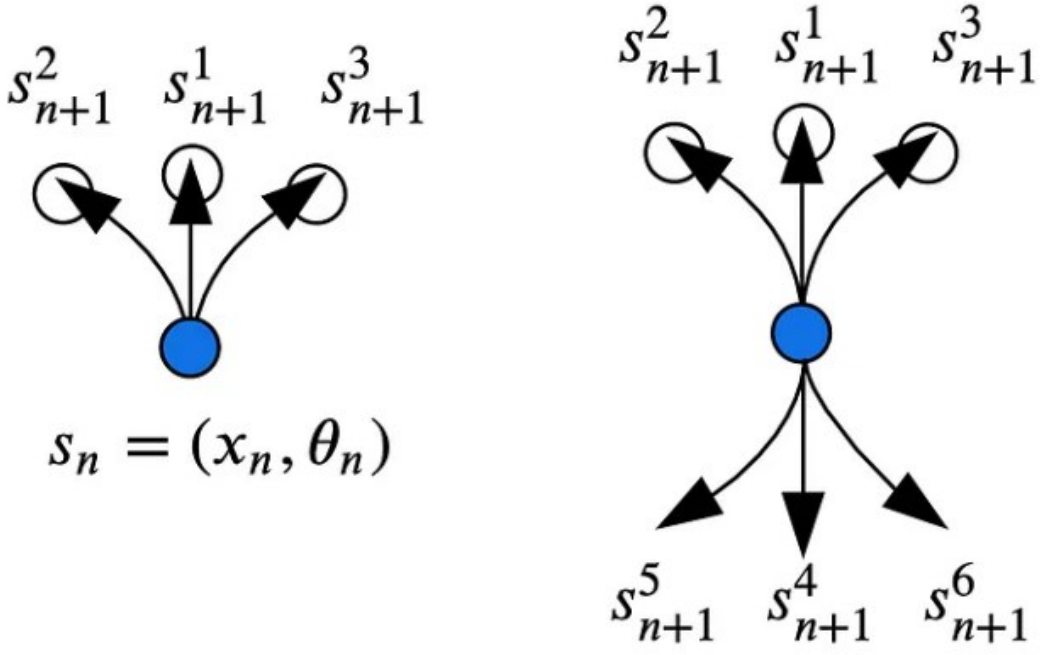


Figure 2 Hybrid A Star Considering Angles

Expansion

In the traditional **A-star**, when we determine the path and we want to select the next cell on the grid, we consider its **center**, but this is illogical in the real world because cars and planes can't turn this way, so with the expansion, we're determining the points on the grid logically depending on the angle of rotation and the initial position before moving. Also, we're

putting more **penalties** on positions, which leads to more distance traveled for changing the angle of rotation. There is also something called **pruning**, and this is used to remove a point from the considered points in the path if it reaches the position from another point in the angle but with **more** movements.

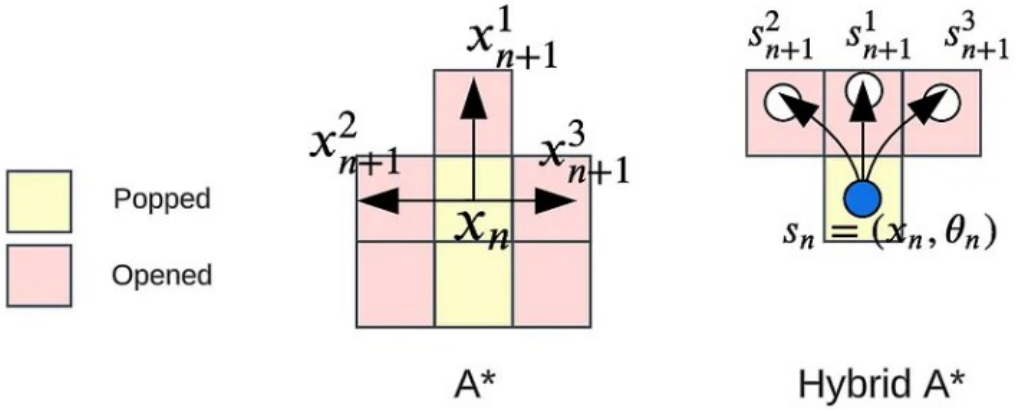


Figure 3 Expansion in Hybrid A Star

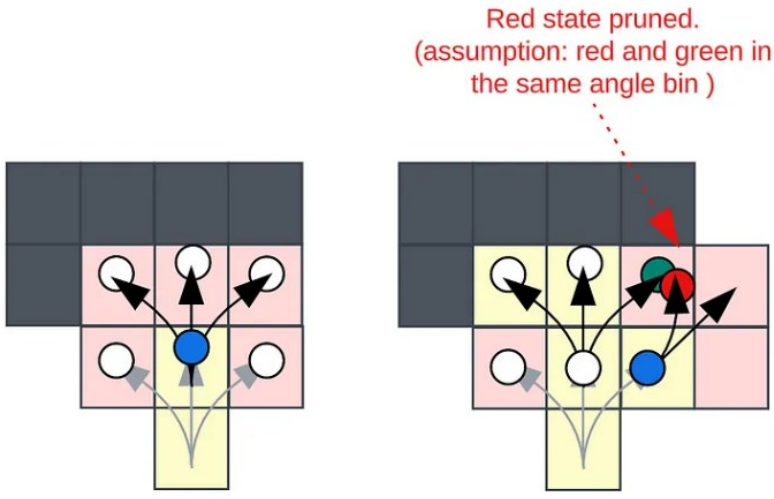
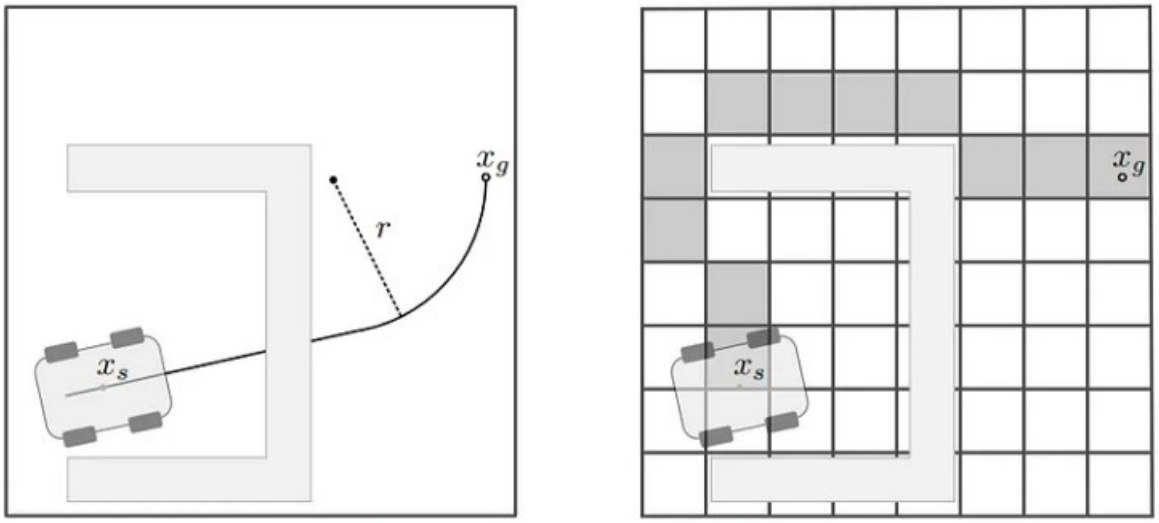


Figure 4 Pruning in Hybrid A Star

Heuristic

In the traditional **A Star algorithm** the **heuristic** is **optimistic** which means it doesn't **overestimate** the cost but sometimes in **A Star** it can be **overoptimistic** because it ignores the obstacles that between the goal and the node which **underestimate** the actual cost, in **Hybrid A Star** we can use **constrained heuristic** and **unconstrained heuristic**, the **unconstrained** one it calculates the cost between the goal and the end considering the orientation of the car and also the position, and in constrained heuristic we consider the **obstacles** which can gives us better results and **lowering the penalizing** that can happen if we took points that are closer to obstacles.



(c) The constrained heuristic heavily underestimating the cost of the path, due to ignoring obstacles.

(d) The unconstrained heuristic accounting for obstacles and dead ends.

Figure 5 Constrained and Unconstrained Heuristic

Analytical Expansion

This concept is used more in **Hybrid A Star** than traditional **A Star** and it means we take the best candidate node currently which means it is the last popped node from the **priority queue**, and we try to connect it to the goal using a **non-colliding line**, the line should be made in **Hybrid A**

Star using **Dubins Path** because we still want to consider the orientation not only the position, anyway, if the line reaches the goal this means we found the **optimal path** and the search will **stop**, we don't do this a lot when using traditional A Star because of the **possibility** of collision and even in **Hybrid A Star**, this analytical expansion will give better results in **non-dense** areas to minimize collisions.