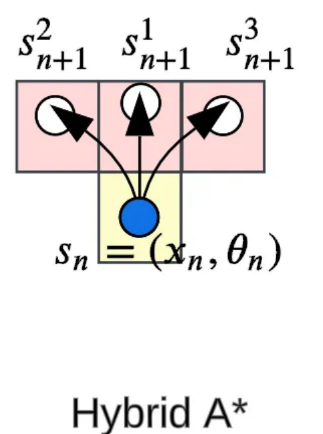
A diagram of a mathematical equation

Description automatically generated**Hybrid A\* Algorithm**

The Hybrid A\* algorithm is a pathfinding algorithm based on A\* but modified to account for physical constraints. While A\* works well in grid-based systems with many applications including [computer games](https://www.youtube.com/watch?v=uiUQ14C1O9Q), autonomous vehicles, and robotics (where objects can move freely in any direction), it doesn't account for real-world vehicle limitations. For instance, when a car moves forward and begins to turn right, it doesn't rotate in place but follows a curved path due to its non-holonomic constraints, preventing abrupt turns. Hybrid A\* adapts the A\* algorithm for scenarios where these kinematic constraints must be considered.

In A\* the state is represented in 2D ( x, y) and can only move/ expands it nodes along the grid directions( up, down, right, and left), while Hybrid A\* is represented by ( x, y, ϑ) reflecting the body’s heading and direction simulating its kinematics, it expands / moves by simulating feasible vehicle motions. It generates new states based on steering inputs, turning angles, and forward/reverse movement. This results in a continuous, curved path rather than discrete movements between grid points.

For the cost-so-far g(s), hybrid A\* has a higher cost than A\* as it considers the length of the arc between each state The cost-so-far g(s) in Hybrid A\* reflects the realistic, physical travel cost of a vehicle following a curved, non-holonomic path. It accounts for the actual distance traveled, penalizes turns and reverse motions, and prunes suboptimal states to maintain computational efficiency. Therefore it has a higher cost than A\*’s g(x) which is the sum of the Manhattan or the Euclidean distance.

For the heuristic h(s), A\* typically uses the Euclidean or Manhattan distance as a heuristic for estimating the cost to the goal while hybrid A\* uses a more complex heuristic that accounts for the vehicle's motion constraints. It may employ methods like Dubins or Reeds-Shepp paths to calculate the minimum distance while respecting the vehicle's turning limitations.