**2.1 Background**

**Flying cars**

A flying car is a type of personal air vehicle or roadable aircraft that provides door-to-door transportation by both ground and air. In the 20th century when airplanes were invented the idea of flying cars came to many scientists. They started building many prototypes, but all their attempts ended in failure. Flying cars were just a science fiction where you can only see them in movies and cartoons. Scientists didn’t stop thinking about a way to make them real and available for everyone and finally in the present many prototypes passed the tests successfully and we will see flying cars in commercial in the next few years it’s no more a science fiction.

**Dynamic path planning**

An Algorithm for planning an optimal path for self-driving vehicles from their starting point to a target point which satisfies the constraints of the vehicles performance and environmental constrains. The generated path grantees that the car will arrive safely with the lowest cost and shortest time all together. The path is continually recomputed as more information becomes available

**2.2 Related works**

**Dijkstra’s shortest path algorithm:** Is an algorithm to find the shortest path between nodes in a graph. The algorithm finds the shortest path from a node to all other nodes in the graph provided that the nodes are reachable from the starting node. In the case of finding the shortest path between two nodes the algorithm can be stopped when reaching the goal node. From the starting node Dijkstra’s algorithm searches all the neighbor nodes and put the costs in a table in ascending order then visite the node with the smallest cost, Once we moved to the node with the smallest cost we started to search all its neighbor nodes and calculate their costs by summing the cost of the edges that lead to the node we are checking from the previous node then update the table again and visit the node with the smallest cost. We keep doing this until we discover all the graph at this point we know the shortest path from any node in the graph to any other node no need to run the algorithm again except if anything in the graph was changed.

**A\* search algorithm:** Is an algorithm used in path finding between two nodes in the graph. The difference between A\* algorithm and Dijkstra’s algorithm are that in Dijkstra’s we calculate the cost between two nodes with only one function which is the actual cost between the two nodes but in A\* we use two function one which is like the cost function in Dijkstra’s algorithm and the other one is the approximate cost from this node to the goal node. The total cost of each node in A\* algorithm is calculated by the summation of the two functions. The real difference is that A\* search only expands a node if it seems promising. It only focuses to reach the goal node from the current node, not to reach every other node. It is optimal, if the heuristic function is admissible. So if the heuristic function is good to approximate the future cost, than you will need to explore a lot less nodes than Dijkstra.