Path Planning Project

I have used **Spline Technique** to generate the path. Firstly, I have declared some anchor points for the Spline corve to fit in. These anchor points are the ones which help in generating trajctories. A Spline header file is included in the main code with the help of which we can fit in the spline that passes through these anchor points and thus providing a smooth trajectory.

The simulator provides the waypoints which are not used up from the previous trajectory. So, to ensure a smooth transition from previous path plan to the current path plan and to avoid jerk during this transition, the path has been continued from the left-over points from previous path. To implement this, the last two points from the left-over path list has been utilised as anchor points which will be used as starting point of the spline. Then, we need some more anchor points which will append the future path plan to the trajectory. For that, if our car is right behind another car and there is a possibility to change the lanes, we will select these points accordingly. If we don't have any possibility, we will maintain the same lane with decelerating speed. As advised in the walkthrough, to get smoother trasition these final anchor points are chosen sufficiently long enough(30, 60 and 90 m) in the desired path direction. Finally, two initial points from the previous trajectory(to maintain smooth transition) and three final points for the desired lane and direction will be chosen to generate a spline. Later, this spline is used to find the future waypoints.

As advised in the walkthrough, to have a reasonable time horizon of 1 second, we need to provide 50 way points to the simulator. The constraints for this project were : a maximum velocity of 50 mph and acceleration and jerk limits which were of 10 units. So, an acceleration of 5 m/sec^2 is set for an incremental step in velocity. This ensures that our accelaration and jerk always remains below 10 units.

Then, to provide waypoints to the simulator, the anchor points have been fit into the spline. By playing with math, we can find that

#points = distance_to_cover / (0.02 * velocity)

which was represented in the walkthrough video of the project. With this mathematical representation, we can find the number of equidistant points on the spline for a given distance_to_cover and velocity. The max velocity is 50 mph which turns out to around 25 m/sec. And we have considered a time horizon of 1 second and so a distance horizon of around 25m would be better. As suggested in the walkthrough, I have chosen 30m which has given me good results. So, with this, for a given velocity, we are able to find equidistant points over the spline for the distance horizon. But, we will have left-over points from the previous trajectory and to maintain smoothness we will continue using them along with the new waypoints which, on a total, adds up to 50 points into future.

To process the shifting of lanes, we predicted the motion of other cars with the sensor fusion data. We will compare the results with our car's state at the same point in future. If results shows collision, then we will perform the lane change first. But, if it is not possible, we will slow down our car with the maximum accelaration limit we have defined(5 m/s^2).