Behaviour and Path Planning Project Write Up

This project was the most challenging one so far in SDCND program. It is hard for me to say that I have done the optimal planner but only the safe planner. There were couple of points that puzzled me with the requirements like acceleration and jerk. It is said that speed, acceleration and jerk is determined by averaging over 1 second yet the values I calculated over one second was far from what the simulator reported. Eventually I have used the maximum value of 0.224 mph increments to increase/decrease the speed which is, with my calculations, comes down to maximum acceleration of ~5m/s^2 if speed is increased/decreased consecutively with every point given to simulator. I have also spent quite some time to implement this planner by using optimizer used in MPC project instead of finite state machine (FSM) approach but I was not successful.

Eventually I have implemented FSM and selected my states so that in each state ego car's safety is ensured with enough space from the leading car depending on ego and leading car's speed. Following are the states ego car can be in; KeepLane, GoRight, GoLeft. GoRight and GoLeft states are transitional states in which ego car changes lane. When ego car is in GoRight and GoLeft states, it cannot go KeepLane state until after lane change completed flag is set.

Speed and lane change decisions are handled by checking forward and backward distances between ego and the closest vehicles to it in each lane. Vehicles farther than 350 metre are not considered in minimum space calculation. Minimum required distance is the twice the maximum speed plus distance required to close the speed difference between ego and front/back vehicles. E.g. if front vehicle is slower than ego then time/distance to reduce ego's speed to front vehicle's speed is calculated and distance is added to the minimum distance requirement.

If forward distance is larger than required minimum, ego can accelerate until it reaches maximum allowed speed. If forward distance is below the required minimum, ego checks other lanes to see if desired minimum forward space is available. To change lanes, in addition to minimum forward space requirement, there is also minimum backward space requirement so that ego doesn't cut other vehicles' way while changing lanes. If other lanes are not available, then ego decelerates until minimum desired space is obtained.

Maximum safety is ensured in each state by checking traffic conditions as following;
To stay in KeepLane: Increase speed to maximum when there is enough space. If not, check for availability of left/right lane. If neither left nor right is available decrease speed to obtain desired space.
To transition GoRight/GoLeft: If there is enough front/back space, it is safe to change lane. Stay in change lane until it is complete and then go to KeepLane state.

Better/optimal implementation would be to utilize cost functions and predict feature positions of ego and other vehicles for different scenarios instead of checking traffic conditions for maximum safety.