Path Generation

Here's the brief description about how the path is generated in my implementation.

Spline Fitting
Basically, I used the Spline module to generate the curve that the vehicle follows. To use Spline, I provide 4 points totally. For the first two points, I use the 9th and 10th points from a previous path cycle. The reason to use the 9th and 10th points is that I always reuse the first 10 points from previous cycle when constructing the path points for the current cycle since the vehicle is still running when the calculation for current cycle happens. Reuse 10 points can provide enough continuity and smoothness while the vehicle still reacts quickly to new situations. 10 is a trade-off number. With too few points from the previous cycle, there's possibility of high acceleration or jerk, and with too many, the responsiveness would be affected. The third point has the Frenet coordinate of (s, d), where s is the vehicle's current location plus 30, and d is the target lane center. The target lane may be the same lane where the vehicle is or a new lane based on the lane-changing algorithm. The fourth point is calculated similarly to the third one, just with a bigger s (current s plus 60) value. Following the Spline path, everything works well.

Path Points Generation
Once Spline fitting is done, path points can then be sampled from it. I provide 50 points totally in each cycle, with the first 10 from the previous cycle. For the remaining 40 points, I use a fixed-acceleration model, i.e. make the vehicle's acceleration a constant during this period. The fixed acceleration can be calculated by:

$$Vend = Vcurrent + a * t$$

while *Vend* is the reference speed, and *Vcurrent* can be calculated approximately by "*dist10-9 / 0.02*", *dist10-9* is the distance between the 10th and 9th points from previous path. Once a is determined, each point's X value can be approximately calculated easily by a physical formula:

$$X = Xcurrent + Vcurrent * t + 0.5 * a * t^2$$

I can use X value here to do the approximation because all X values are in the vehicle's coordinate system, which are in an almost vertical line in front of the vehicle. Y values can then be got from Spline.

Calculation above can be applied to normal situations where there are no other vehicles in front and the reference speed can be used as *Vend*. But when there are slower vehicles ahead, *Vend* and acceleration must be adjusted to avoid collision. The way to adjust this is to get *Send* first, i.e. the future (after 50 * 0.02s) S value, which can be calculated by:

$$Send = Sfront - safe\_gap$$

Where *Sfront* is the future (after 50 * 0.02s) S value of the vehicle just in front, assuming its speed remains unchanged. Safe_gap is the gap to keep to avoid collision. I use 30 here. Again, I use X as approximation, so I have *Xend*. I can get the adjusted speed *Vend-adj* and acceleration by following:

$$Xend = (Vcurrent + Vend\text{-}adj) / 2 * t$$
$$Vend\text{-}adj = Vcurrent + a\text{-}adj * t$$

Once I get the adjusted acceleration, (X, Y) can be calculated similarly as normal situations.

One more note is that, I put an limit to the acceleration value calculated above so that the it won't exceed the allowed maximum value.

Target Lane Determination

In the calculation above, when fitting Spline, two points in the target lane are needed. Target lane can be the current lane or a new lane. In my implementation, I seek to change lane when *Vend-adj* calculated above is less than 45MPH. To evaluate which lane to go, I use two simple cost functions for each lane. One is speed related. The faster the vehicle just in front of us in that lane, the lower the speed cost of that lane. The other is distance related. The farther the vehicle just in front of us in that lane, the lower the distance cost of that lane. Also, I define a hard-failure situation, i.e. if there are vehicles too close either in front of, or behind us in a lane, I bump its cost to a very high value, so that lane won't be considered for lane changing. To avoid oscillation, i.e. vehicle goes into one lane and comes back soon because two lanes have similar cost, I reduce the cost for current lane a little bit. Based on the cost value, new lane can be selected.