我在B站上学习了LQR的原理，他的大致框架为基于空间状态方程输出的误差用最优控制率输出控制指令（自行车模型中即为前轮转角）。其中矩阵是通过求解二次型代价函数

的最小值的解得到（需要用到Riccati方程）。而在matlab中可以直接调用lqr函数实现矩阵的计算。而我设计的路径是一条直线路径，为了简化模型，我将模型速度设定为恒定的10m/s。再利用visualization.m可视化修正过程（初始化状态为[0.5;0;0.1;0],目标为[0;0;0;0]）。

关于中的和：从直观上理解中的各项数据的大小反映出对于各种误差的惩罚程度（例如若想要使得横向误差最小则把中对应的值设得更大些，使其模型偏向于快速消除横向误差）；而则用于惩罚输出的控制量，在自行车模型中控制量只有一个，则实际上用于惩罚转向输入，越大，相对来说模型输出的转向角会更小。

LQR的优点是能综合模型输出的所有输出状态来输出控制量。从直观上看，对于的控制更有预见性（避免画龙）。