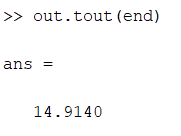
最终结果，生成轨迹（放大后发现正好没出界）

**总体思路**：利用MPCC寻路（路径规划），将寻得路径交给LQR控制器进行控制，获得最终的仿真结果。

**关键文件**：init\_mpcc.m，mpcc\_sim.m，init\_car\_dynamic.m，car\_sim.slx

MPCC的总体思路是向前看，即获取即将经过路径的边界，以此作为路径搜寻的依据。这样做可以提前开始转弯，减少对于控制器的要求。

具体的实现上，我将轨迹按照1：10的比例缩小。这实属无奈之举，因为不进行这样的处理MPCC会发散。相应的，寻得路径也经1：10缩小。并且为了防止MPCC在处理不闭合轨迹在最后会出现的观测缺失，我将终点按照其方向进行延长，以完成MPCC的模拟。为了验证寻得路径的正确性，我将寻得路径等比例放大，接入LQR控制器进行验证。

init\_mpcc.m：该文件提供了MPCC参数初始化工作，包括延长轨道以求解靠近终点处的轨迹，通过裁剪边界获得更加安全的轨道，在起始点处预先求解五次使得起始点的轨迹更为合理。（该处的vx=0.2较小（由于是在缩小后的轨道上，等效vx=2），这样可以缩小初始状态下轨道的曲率）

mpcc\_sim.m：该文件是MPCC求解的主要过程，通过设定迭代阻尼防止轨迹振动，之后求解转向角，获得下一步的初始位姿。在一次迭代的最后我们判断更新后的位姿是否达到终点线来结束循环。

init\_car\_dynamic.m：与源文件的区别就是参数Q和R，以及导入的轨迹是traj\_mpcc.mat而不是赛道的中线（该处的vx=20较大，这样可以充分在直道上发挥性能）

car\_sim.slx：采用LQR控制器，因为LQI控制器在大部分情况下会慢0.3秒且更不稳定。其余部分将Gain1改为1，并且与Sum2模块进行连接。