

## 第二章作业巡航系统

第二章作业是实现横纵向的 PID 控制，详细的任务说明参考作业 PDF 文档

在进行作业之前，要使得仿真器 SVL 和 ROS 进行正常通讯，SVL 和 ROS 通讯参考之前的文档，不再赘述

编写 PID 控制器按照课程里面的说明进行编写即可，只不过要**考虑抗积分饱和**，这里使用了限制积分上下界的方案，也可以参考课件选用其他的方案

PID 参数可以参考代码里面的参数，也可以按照需求调整参数，但是也不比过于追求较小的超调，超调和响应时间是一个平衡的过程

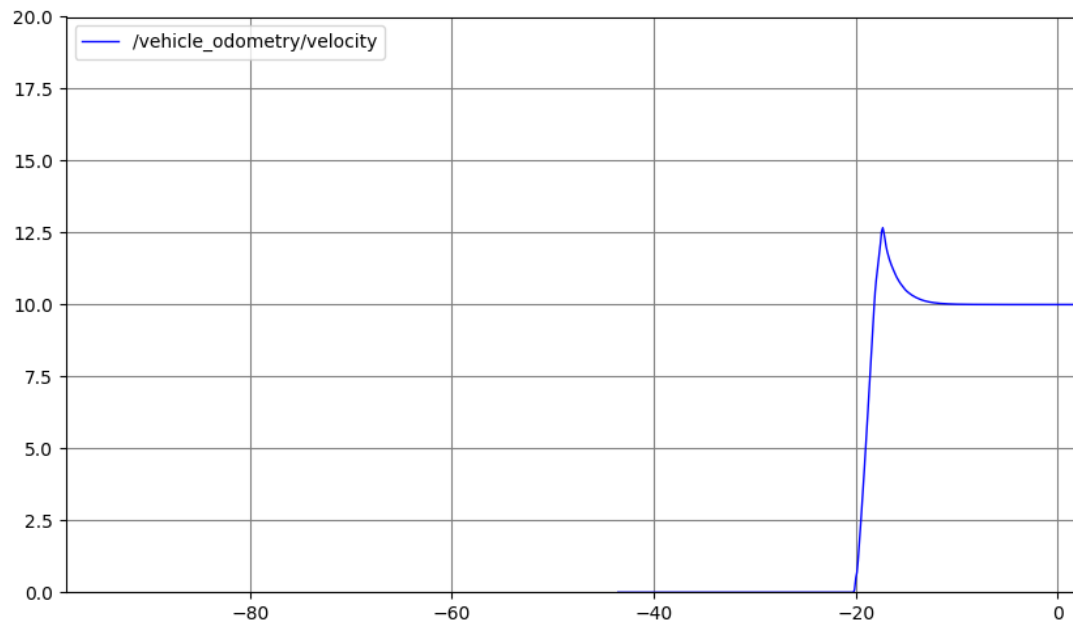
编写完成代码之后，使用 ROS 的 catkin\_make 编译，编译完成没有错误之后，使用下列命令运行

```
source devel/setup.bash
```

```
roslaunch vehicle_control vehicle_control
```

注意要在代码的工作空间目录运行

运行完成之后车辆速度曲线如下图所示，虽然有一定的超调，但是还在合理范围内，最终的静态无常也很小，在 ROS 运行的窗口也可以观察输出的控制命令，车速误差，角度误差等参数，具体的车辆表现可以参考文件夹里面的视频



```
acceleration_cmd: 0.03583
yaw_cmd: 1.13711e-06
start_heading: -1.5708
heading: 1.5708
v_err: 9.53674e-07
yaw_err: -8.25511e-08
control_cmd.target_wheel_angle: 1.13216e-06
acceleration_cmd: 0.0358305
yaw_cmd: 1.13216e-06
start_heading: -1.5708
heading: 1.5708
v_err: 9.53674e-07
yaw_err: -8.25511e-08
control_cmd.target_wheel_angle: 1.1272e-06
acceleration_cmd: 0.0358305
yaw_cmd: 1.1272e-06
start_heading: -1.5708
heading: 1.5708
v_err: 9.53674e-07
yaw_err: -8.25528e-08
control_cmd.target_wheel_angle: 1.12225e-06
acceleration_cmd: 0.0358306
yaw_cmd: 1.12225e-06
```