

智能控制报告

大作业.基于模糊控制及神经网络控制的小车转弯算法

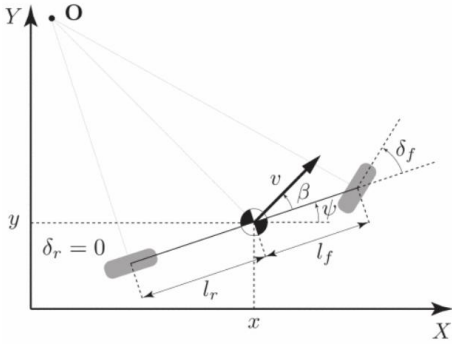
3220101111 洪晨辉

$$\alpha_r = \beta - \frac{L_r}{\dot{x}}$$

一、问题分析

车辆如何转弯是自动驾驶算法的重要研究内容。传统 PID 尽管已经足以解决简单情况下的转弯问题，但是在响应速度和稳定性方面不及更为先进的控制算法。因此，本报告力图通过模糊控制和神经网络控制的方法，与传统的 PID 算法进行对比，以展现此种方法的威力。

在转向问题中，汽车通常被简化为两自由度的两轮车模型。如图所示：



在本报告中，因为汽车可能在转过大角度时漂移，因此采用更为复杂的动力学而非运动学模型，以便更好地模拟现实情况。其动力学方程如下：

$$\begin{aligned}\ddot{x} &= \dot{\psi}\dot{y} + \alpha_x \\ \ddot{y} &= -\dot{\psi}\dot{x} + \frac{2}{m}(F_{c,f}\cos\delta_f + F_{c,r}) \\ \ddot{\psi} &= \frac{2}{I_z}(l_f F_{c,f} - l_r F_{c,r}) \\ \dot{X} &= \dot{x}\cos\psi - \dot{y}\sin\psi \\ \dot{Y} &= \dot{x}\sin\psi + \dot{y}\cos\psi,\end{aligned}$$

其中， \dot{x} ， \dot{y} 为汽车在自身质点坐标下的沿两轴速度， X, Y 为车辆在世界坐标系下的位置坐标， ψ 为汽车自身的横摆角， δ_f 为前轮转过角度， l_f 和 l_r 为前后轮和质心的距离， m 和 I_z 表示车辆自身的质量和转动惯量， α_x 为车辆轴向加速度。比较复杂的是 $F_{c,f}$ ， $F_{c,r}$ ，他们分别代表前后轮胎侧向受力大小，由以下近似公式决定：

$$\begin{aligned}F_{c,i} &= C_i \alpha_i \\ \alpha_f &= \beta + \frac{L_f}{\dot{x}} - \delta_f\end{aligned}$$

β 为车辆侧偏角。本例中采用的汽车参数如下：

| | |
|----------------------------|--------|
| m (kg) | 2045 |
| I_z (kg·m ²) | 5428 |
| l_f (m) | 1.488 |
| l_r (m) | 1.712 |
| C_f (N/rad) | -38925 |
| C_r (N/rad) | -39255 |

现为模拟车辆的转弯，设置一个简单的圆弧，该圆弧半径为 15m，角度为 90 度，符合正常转弯任务的半径。为控制变量，令车辆入弯速度为 30km/h，轴向加速度为 0.3m/s²，控制车辆 δ_f 完成转弯任务，出圆弧即视为转弯任务完成。

二、算法设计

对于该系统的控制成功度，有两衡量标准。其一为车辆偏移圆弧车道的距离：

$$e_d = \sqrt{(X - X_{round})^2 + (Y - Y_{round})^2} - R$$

其二为车辆在此点与车道最近点的切向方向的偏离角度：

$$e_a = \text{atan2}(X - X_{round}, Y - Y_{round})$$

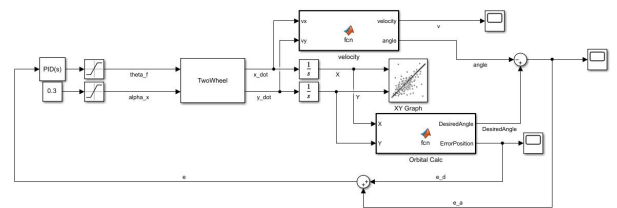
因此，控制器表现为一个 MISO 系统，力图综合两者表现，设计控制。对于普通 PID 和神经网络，本报告采用的是非常不精确的两者直接相加的方式，这种不可理喻的方法意外地可以成功。

$$e = e_d + e_f$$

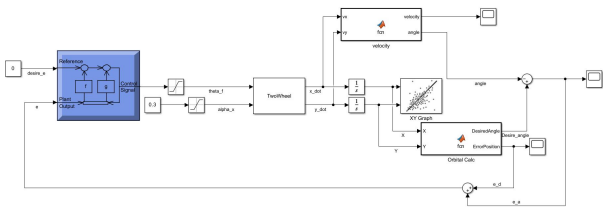
而对于模糊控制，本报告的控制分立，分别考虑两者的表现，通过模糊逻辑输出控制角度 $\delta_{f,d}$ ， $\delta_{f,a}$ ，再将两者加权叠合：

$$\delta_f = \omega \delta_{f,a} + (1 - \omega) \delta_{f,d}$$

普通 PID 控制如图：

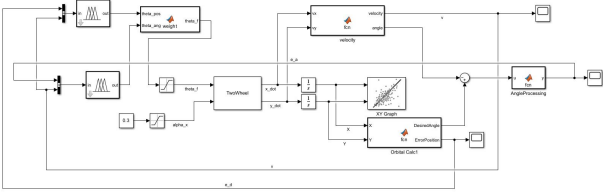


可以看到 PID 的输入为两误差的线性叠合。同样的，本报告的神经网络控制也采取一样的方式：

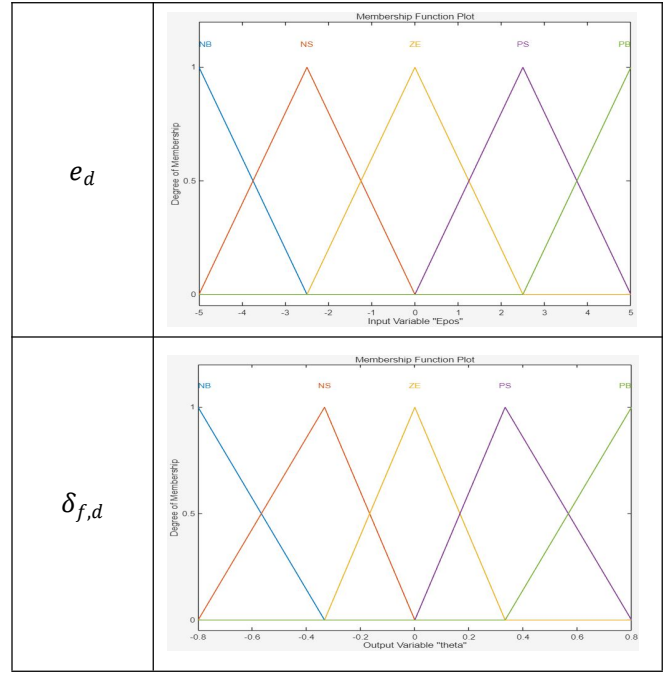
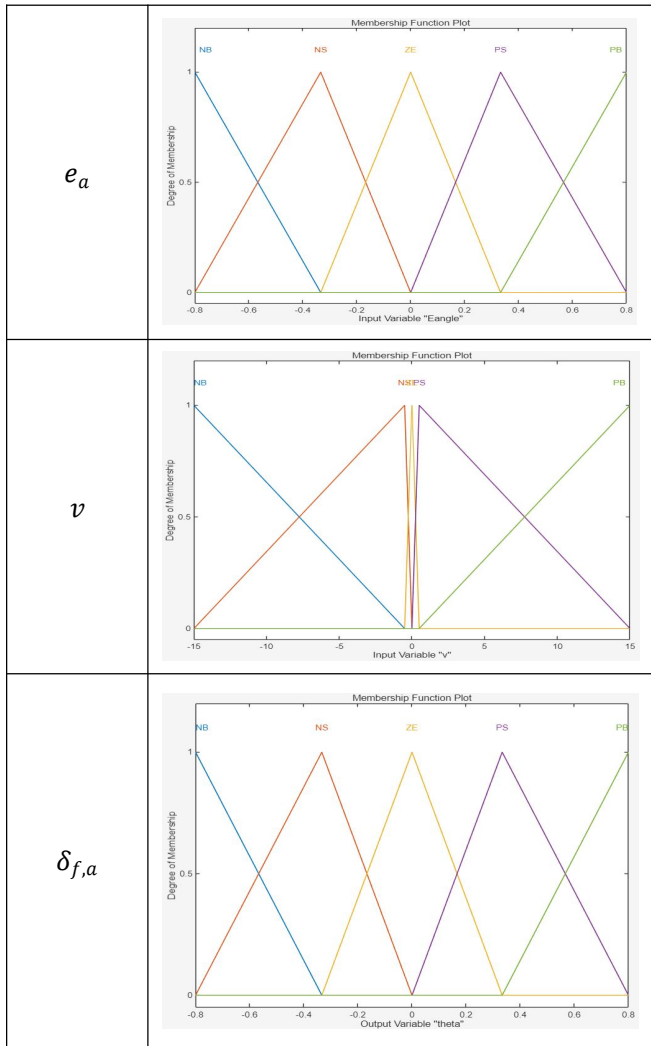


另：在本报告中，为保证训练速度，采用 **NARMA-L2** 训练神经网络。该算法的核心与普通的模型参考控制相仿，区别在于，**NARMA-L2** 使用近似的线性模型来代表系统，相对较为简单。

而模糊控制对两误差分别处理：



模型的模糊逻辑为：



其模糊规则表参照了一篇论文当中给定的值。此表的 θ_{pos} 对应 $\delta_{f,d}$, θ_{ang} 对应 $\delta_{f,a}$, E_{pos} 对应 e_d , E_{ang} 对应 e_a 。

| θ_{ang} | | E_{ang} | | | | |
|----------------|----|-----------|----|----|----|----|
| | | NB | NS | ZE | PS | PB |
| v | NB | PS | ZE | ZE | ZE | NS |
| | NS | PB | PS | ZE | NS | NB |
| | ZE | ZE | ZE | ZE | ZE | ZE |
| | PS | NB | NS | ZE | PS | PB |
| | PB | NS | ZE | ZE | ZE | PS |

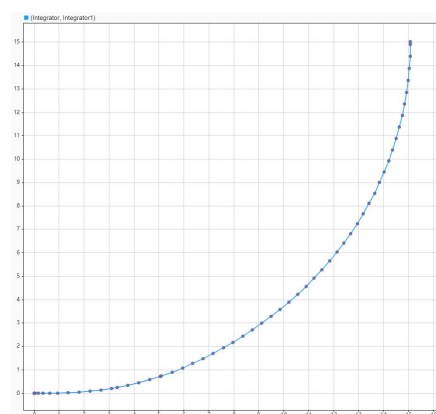
| θ_{pos} | | E_{pos} | | | | |
|----------------|----|-----------|----|----|----|----|
| | | NB | NS | ZE | PS | PB |
| v | NB | NS | ZE | ZE | ZE | PS |
| | NS | NB | NS | ZE | PS | PB |
| | ZE | NB | NS | ZE | PS | PB |
| | PS | NB | NS | ZE | PS | PB |
| | PB | NS | ZE | ZE | ZE | PS |

三、结果表现

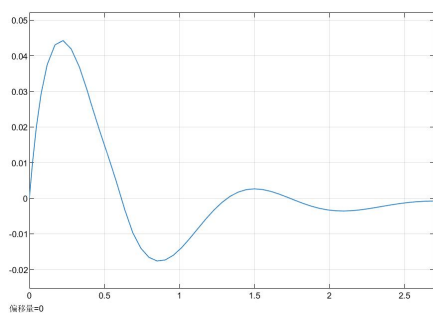
经过测试，所有的算法表现都不错，在 3 秒内基本实现了较为完美的转弯轨迹。其差别在于三者的响应时间和出弯后误差。

1. 普通 PID

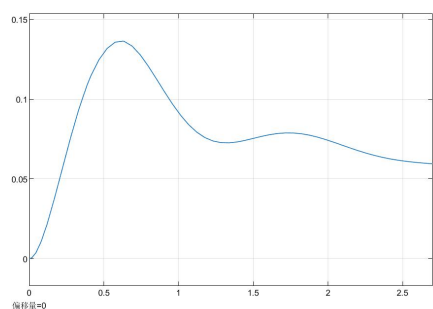
普通 PID 的参数为 $K_p = 2$, $K_i = 0.5$ 。并没有采用 K_d ，其原因之一在于加入 K_d 后，系统变得不大稳定。其控制得到轨迹如下：



而角度与距离的误差如图：



e_d

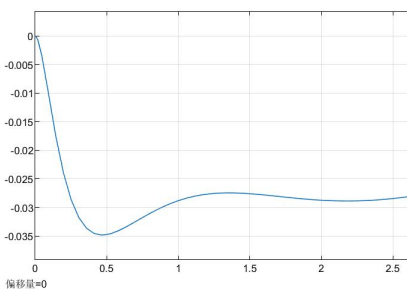
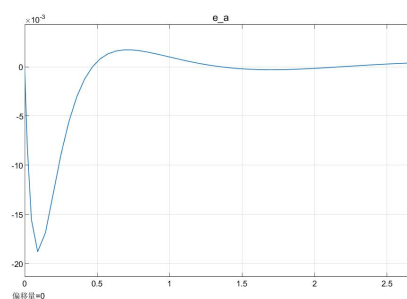
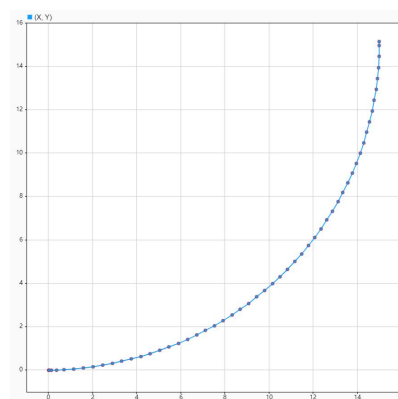


e_a

可以看到，两者的响应时间相对较长，且 e_d 的残差没有被完全消除。当然，将时间拉长，可以看到事实上在逐渐减小——但是在本问题中，超过这一时间，汽车就已经出弯，因此研究该部分没有太大意义。

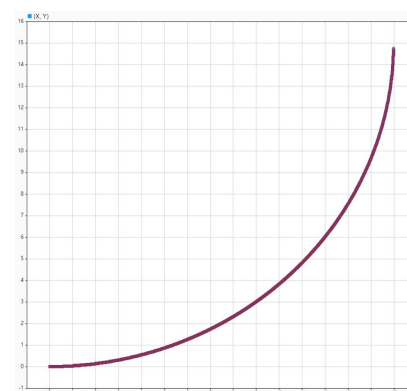
2. 模糊控制

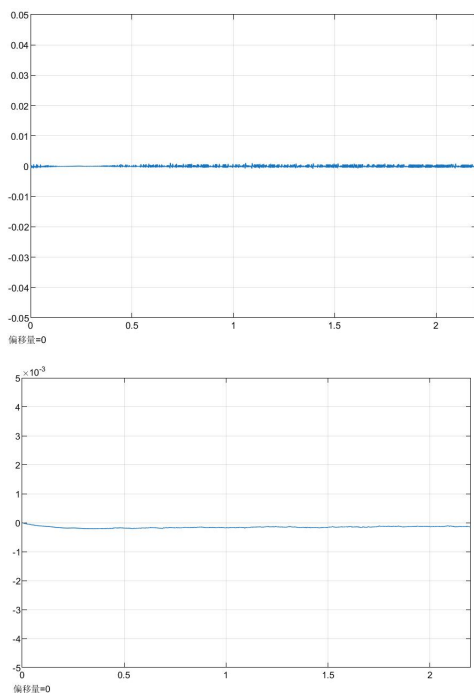
那么，换成模糊控制有没有效果呢？有的。模糊控制得到的轨迹及角度、距离误差如下：



角度的响应时间大大减小——几乎减小了一倍。当然， e_d 的残差仍然存在，但是大幅减小，且和PID控制一样，会随时间拉长而逐渐减小。

3. 神经网络控制





相较于前两种算法，NARMA-L2 的控制更为优异，令人惊讶地几乎没有残差，将误差一直控制在 0 左右。

四、后记

1. 神经网络的问题

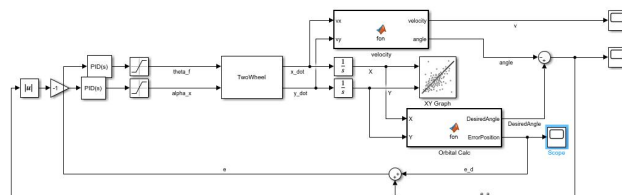
……然而，由于神经网络对于系统辨识的组数实在太少（只采用了 100 个数据），系统的输出不可避免地出现了较多的毛刺。这明显是由于输入的数据的组数不够，导致系统不足以更好地拟合导致的。

同时，如此训练的神经网络仅对于该车辆入弯速度 30km/h，轴向加速度为 0.3m/s^2 的模型有效。如果改换模型，譬如让车辆轴向加速度增大一倍，该模型的动力学辨识就会彻底失效。此时，则需要重新训练模型。因此，比较好的方法事实上是把踏板、入弯速度等因素加入神经网络加以考量，或者如果速度足够，在工况变化期间进行动态训练适应（而这对于现在的算力而言，显然不现实）。

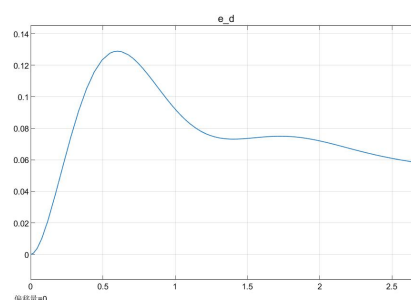
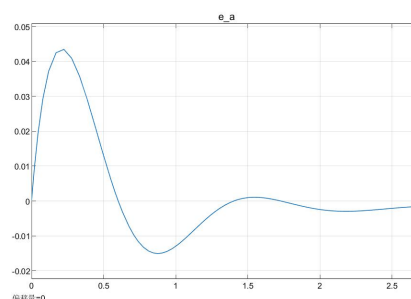
2. 踏板——是否可以改进？

这激发了一个问题——是否可以自动控制汽车的油门和刹车踏板，使得汽车轴向加速度 α_x 变化，以便更好地控制汽车轨迹？限于时间原因，本报告仅在 PID 算法中讨论此想法。答案是，在拥有最大及最小限制行程的时候，踏板对于整体角度和距离误差控制的影响微乎其微，主要影响的是速度。

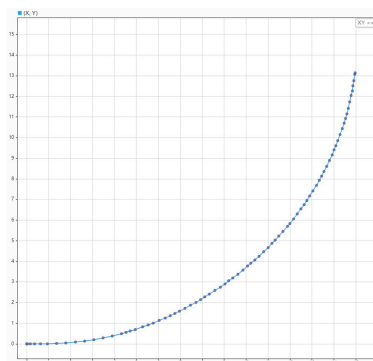
对 PID 算法增加踏板的 PID 控制环。如图，试图使得角度偏离过大时，踏板减速，以便回到正常位置，而回到正常位置时加速。



然而，观察时发现，这对于 e_d 和 e_a 的影响几乎可以忽略不计。



且出弯速度显著变慢。



(同样的时间内尚未出弯)

在模糊控制的实践中也碰到了类似的情况。当试图加入相似的模糊逻辑器时，系统的控制稳定性并未发生显著变化，甚至在一些情况下大幅降低。

综上，在本任务中，踏板最好维持定值，一直加速并不会影响控制性能。