

组会汇报

7.22组会工作总结

1. 学习唐老师发的新的4篇传统控制方向的论文，针对其中一篇有随机时延建模的论文学习了其中的数学建模
2. 撰写考虑状态预测不确定性的延迟强化学习代码

论文阅读

上次给唐老师讲过自己的idea以后，是说因为是结合了两篇强化学习的论文，创新点够不上顶会，让我要跨大领域去做交叉，就给我发了4篇**传统控制领域**过去师兄的论文让我学习。

这些论文大部分是聚焦在网络控制系统（NCSs）中同时存在的随机通信延迟和数据包丢失问题，然后设计一个状态估计器

虽然说都是状态估计，但我看的强化学习延迟论文估计状态都是使用神经网络做预测，这里都是基于数学推导的传统预测方案。

阅读后的启发

学习**卡尔曼滤波+强化学习**这样的结合传统预测和强化学习的论文

[KalMamba: Towards Efficient Probabilistic State Space Models for RL under Uncertainty](#)

这是在状态信息不完全的情况SAC算法结合卡尔曼滤波来提高性能

遇到的问题：

1. 在非强化学习领域的论文阅读**理解不深**，而且不知道怎么**套用到**时延强化学习上来
2. 简单的强化学习方向的A+B又会导致**创新不强**

代码编写

这一块框架已经完全搭好了，所以还是想试试idea是否有效，发一篇较小的论文。

梳理了baseline代码的框架

确定主要更改 `/src/runner.py`

需要增加的模块

`mc-dropout`

`Q uncertainty 预测`

`uncertainty actor`

`网络更新`

- 写好了mc-dropout状态预测mlp,rnn,transformer,用以得到 $[S_1 - S_N]$

```
def mc_mlp_rnn_Net()
def TransformerNet()
```

- *critic*利用 $[S_1 - S_N]$ 得到 N 个 $Q(S_i, a)$ 以及*uncertainty*的估计算法函数

```
def def get_critic_N_Q()
def get_N_Q_uncertainty()
```

下周的工作

1.继续阅读，尝试跨大领域建模时延强化学习

2.把剩余代码编写完成，同时考虑利用已经写好的模块，提前实验，有关 $mc - dropout$ 预测 $Q_{uncertainty}$ 是否可以反应出预测状态和真实值状态的差距，实验实现细节待思考

3.nips, rebuttal