

# First Step

---

- 首先实现一个 flocking 并使用 PID 控制, 运动方式由路线点给出的 简单 demo.
- 所有的无人机都由 reynold 控制即可.

# Second Step

---

- 此处需要实现关于 无人机偏航角的 实现.

## drone states

---

```
state = np.hstack([
    self.pos[nth_drone, :], self.quat[nth_drone, :], # 3,4
    self.rpy[nth_drone, :], self.vel[nth_drone, :], # 3,3
    self.ang_v[nth_drone, :],
    self.last_clipped_action[nth_drone, :]
])
```

## utils

---

- ☒ 性能分析 [https://blog.csdn.net/weixin\\_40583722/article/details/121659851](https://blog.csdn.net/weixin_40583722/article/details/121659851)

```
python -m cProfile -o flame_of_flocking.prof flocking.py
flameprof flame_of_flocking.prof > flame_of_flocking.svg
```

PROF

- ☒ 使用tensorboard

```
tensorboard --logdir=/home/lih/fromgit/gym-pybullet-
drones/gym_pybullet_drones/src/results/
```

## TODO LIST

---

- ☒ 需要在 flocking\_aviary 中将 ctrl\_freq 和 decision\_freq 分离开来, 因为 BaseAviary 中 step() 是按照 ctrl\_freq 调用的
- ☒ 计算 gaussian\_process 需要的计算压力太大了, 需要提高运算速度, 考虑使用 GPyTorch
  - 使用了 GPyTorch, 将运行速度提升了 5 倍
- ☒ 考虑使用 方位测量 和 距离测量进行替代, 这样的 reward 似乎难以设计

- ☒ 使 control\_by\_RL\_mask 的无人机无法接受迁移控制指令
- ☒ 考虑如何设计 reward function, 是否把对 angular speed 的幅度限制加入 `_computeTerminated()`, 已加入
- ☒ 为 gptorch 添加数据的归一化与反归一化, 现在 gpytorch 的行为与 sklearn 差异过大 已解决

## 几点初步设计思路

- ☒ 当前 control by RL 的无人机无法获取迁移指令? 这合理吗, 容易在初期丢失目标造成 terminated (这是在 eval 时观察的情况)
  - 并不是 eval 时没有获得迁移指令, 而是 deterministic 模式下无法运动
  - eval 时, 没有将 fov
- ☒ reward 完全无法得到收敛, 动作空间的设计方式是否有问题!
- ☐ 考虑应当模仿图的离散化, 重新设计 action\_space 更小,
  - ☒ 考虑将 action\_space 设计为基于 diff 的形式, 这一解法之前有误, 现已解决。目前控制赶不上规划
  - ☒ 使用 speed 模式时, 无法产生真实 yaw action
  - ☐ 考虑重新设计 reward, 我这个持续监控的模式, 可能不适合 reward
- ☒ 将 num\_uav 和 control\_by\_RL\_mask 设置为随机
- ☒ 减少 action space, 提前 truncated RL

## STAMP 的新思路

- ☐ 探讨了各种建模对于环境的影响, 但无论如何, 我们需要对 action space 进行图的离散化。
  - ☐ 这里需要考虑的是如何评估 Node\_feature, 这里需要将 GP 建模在 action space 之中
  - ☐ 先调通在图的离散化中基本的操作
- ☒ 使用 IPP 思路进行建模, action type 应当为 yaw 直接控制的形式

## 先考虑简化 OBS 与网络的设计,

- ☐ budget 在 IPP 中用于判断下一个节点是否可达, 我似乎并不需要这样一个 budget
  - ☒ 这个 budget 可以用于现在在 knn 中的连接, 因为我们需要限制过大的 yaw action
- ☐ 那么现在输入为增广图, 已执行过的路径, 这是一个简化
- ☒ 按照我的理解, 我的 action\_net 与 value\_net 已经包含在 attention\_net 之中
- ☐ 和每一个 node 连接的节点数量需要多一点, 以能够再次发现目标
- ☐ 先减少训练时, 无人机的数量,
- ☐ 还是需要考虑 FOV 对不确定性的影响
- ☐ 如何建模 reward function ?

## Install

- 需要自定义 pythonpath 避免 gym 使用已经注册并移动至 sitepackages 目录的环境:

```
# set python path
export PYTHONPATH=/home/lih/fromgit/gym-pybullet-drones:$PYTHONPATH
echo set python path for flocking aviary done!
```