MPC迅速简介

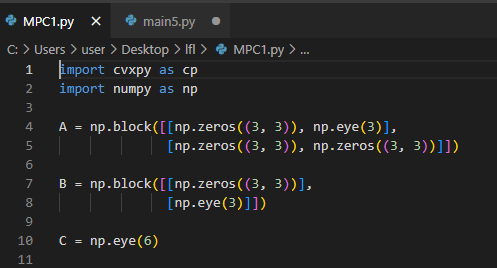
这个部分就可以完成MPC模型的控制

MPC又称为模型预测控制 我将分开为您简介

MPC是一种控制器，可以将输入模型的状态，作为控制器的输入，使用控制器的目标函数(公式)来对模型进行约束

模型=无人机= =这个数学式，我们会将无人机变成一个线性方程，称为模型。

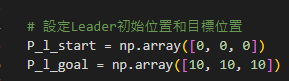
其中我们可以自行定义A B u等(我已经在MPC1.py中完成模型建立也确认可行，此部分只需要复制MPC1.py



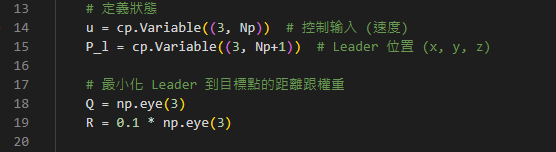
说完模型接着说预测

MPC控制器会有两个时间 控制区间Nc(控制器要运算的时间)和预测区间Ns(实际整个控制器运行的时间)

可以看到目标函数中有对应的Nc和Np

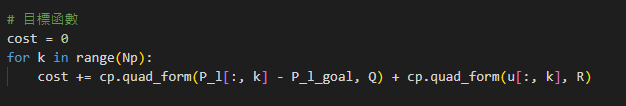
在定义MPC控制器的初期就要将会用到的参数加入其中(在main中定义如P\_l\_start和P\_l\_goal)以及定义该MPC控制器的Nc和Np

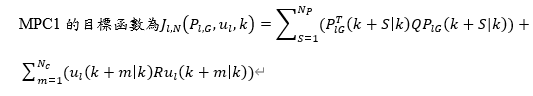
飞机的状态因为是不断更新的，所以要在main中定义，再于mpc控制器中呼叫做控制



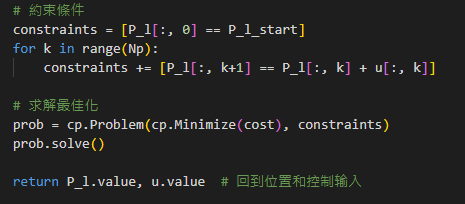
这部分一样是可以照抄的。

接着才是MPC不一样且重要的地方:目标函数

只需要将我给的数学式用程式语言表示即可



這個部分每一個MPC控制器不一樣，所以要根據我給的數學式下去編寫



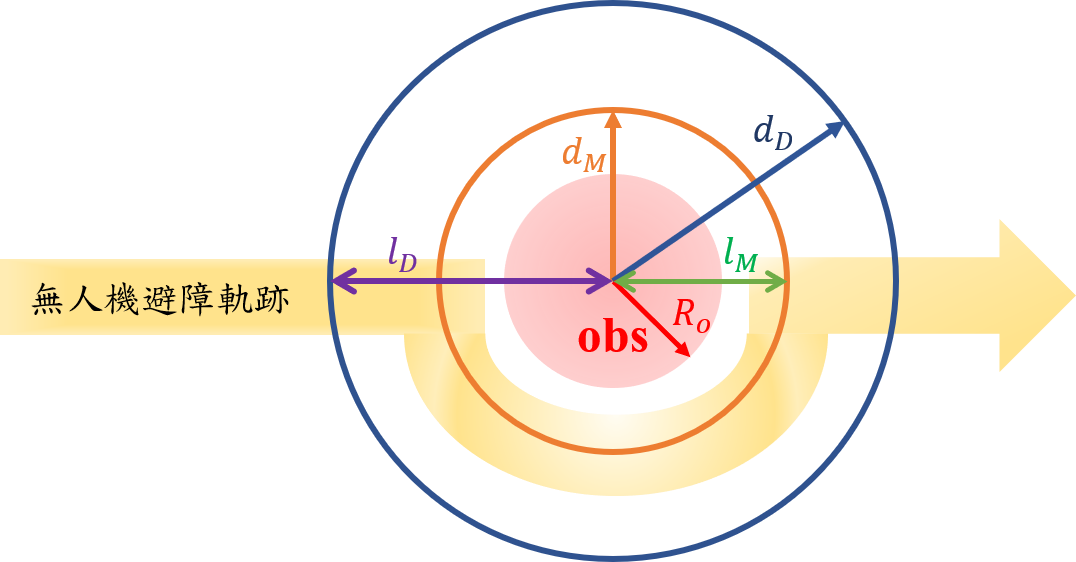
约束条件依据每一个控制器要做的事情下去写

像我第一个控制器的功能是让Leader的起始点作为目标点的起始点，在模型内PL每一次更新状态都会回传后再次求解。

以爬楼梯做为比喻，每爬一格他就会回报他在第几格(X,Y,Z)三维座标

以上部分是我搭配程式对MPC的简介，希望对您有帮助。

MPC3 Leader和障礙物(obs)避障



以下為對於無人機進行障礙物避障時的條件設定

UAV示意為Leader無人機

：障碍物安全半径(危险区域)

：无人机避障理想轨迹半径

：障碍物的安全半径的长度

：障碍物半径

设：障碍物的座标为

Leader无人机的座标为

目標函數

：Leader的座標和obs的相對距離-障礙物半徑

1. 障碍物的避障范围没有UAV(Leader)
2. 障碍物的避障范围出现UAV

功能˙当障碍物范围内出现UAV，目标函数最小化Leader和障碍物间的距离，但保持一个l\_M的关系Leader避障MPC3的解释

一开始你要在main中定义一个三维的球体作为障碍物()，这个障碍物本身的长度就是Ro，然后他的x座标是要随机生成的(一开始可以先自行设定，确定避障功能后再用随机生成他的xyz座标)，预期成果是至少有两个障碍物(可以先生成一个在leader必经的路线(n,n,n),n=3~7(因最短路径必为直线，所以生成在直线路径中一定需要进行避障)，后续避障功能顺利再随机生成障碍物位置即可(main中定义后要在MPC3定义时呼叫障碍物)

MPC3的目标函数里面写(2) a可以自行选择 如0.5-1.5就可以，在main中要做的事情如下

定义Leader位置(已经完成)

定义障碍物位置(尚未完成)

Main中的判断式

判断leader和障碍物之间的相对距离式

的情况:leader跟障碍物的距离大于dD=ld，數字相同但定義不同，在main(自行定义数字，如3 则main正常运行

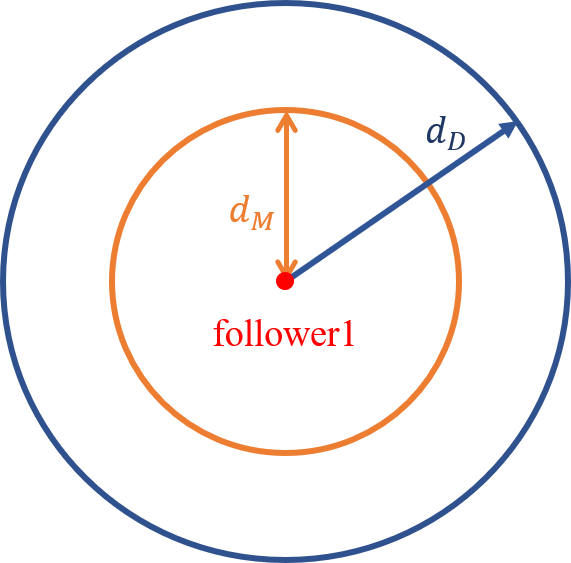
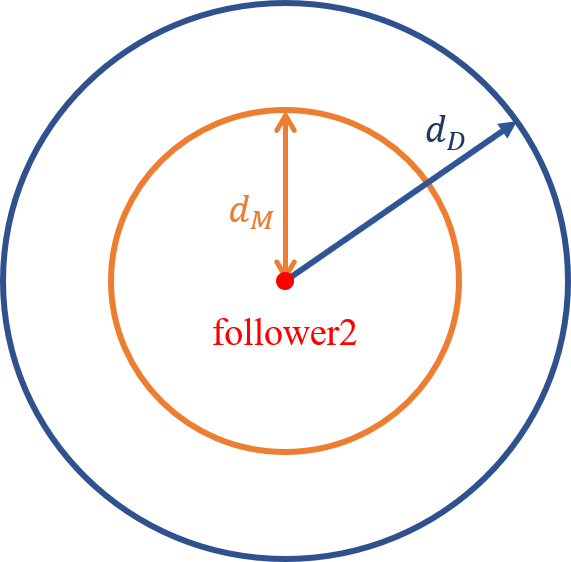
的情况leader跟障碍物的距离小于Dd，则要启动MPC3，公式中的lm=dM可定义为2，本身的定义也就照着

定義寫就好了，Ro就是障礙物的半徑，上面在定義障礙物就會用到

這些東西定義好後在定義MPC3時呼叫就可以。

MPC3的約束條件參考MPC1，用到的是Leader 大致上應該是這樣。

MPC4 follower1和follower2之间的避障

障碍物的安全半径也可以视为无人机(follower)的安全半径(防碰撞距离). 

：無人機安全距離

以表示follower1，中心點座標為

以表示follower2，中心點座標為

如果follower1和follower2之间的距离小于，则两者间需要避障

目標函數

1. follower1和follower2的距离大于安全半径，不需避障
2. follower1和follower2的距离小于安全半径，需要避障

功能follower1和follower2之间的距离最少要保持两个d\_M的距离，以免碰撞

MPC4的介紹

这里是为了避免follower在避障过程中会有产生碰撞的机会，所以要有MPC4

(MPC4在main中的权重要比MPC2大)(因为我的MPC2是以向量跟随leader，有可能leader避障成功但follower相撞或有相关问题)，可将mpc2中的约束条件更改为软性约束(先完成避障后若有发生运行不了的问题再更动此部分)

MPC4的功能是防止碰撞

目标函数一样是输入(2)，b一样为自行定义

其余dD和dM在main中阶已经定义过，引用即可

在mian中要定义d12，后于MPC4定义时呼叫即可

Main中若f1和f2的相对距离小于2个dM则需要启动MPC4

MPC4中的约束条件则参考MPC2，约束Follower

以上希望能帮助到您 谢谢