

# MATLAB仿真

启动程序

test\_trajectory.m

得到位置、速度、角度的实际量和控制量曲线→

算法实现:

controller.m

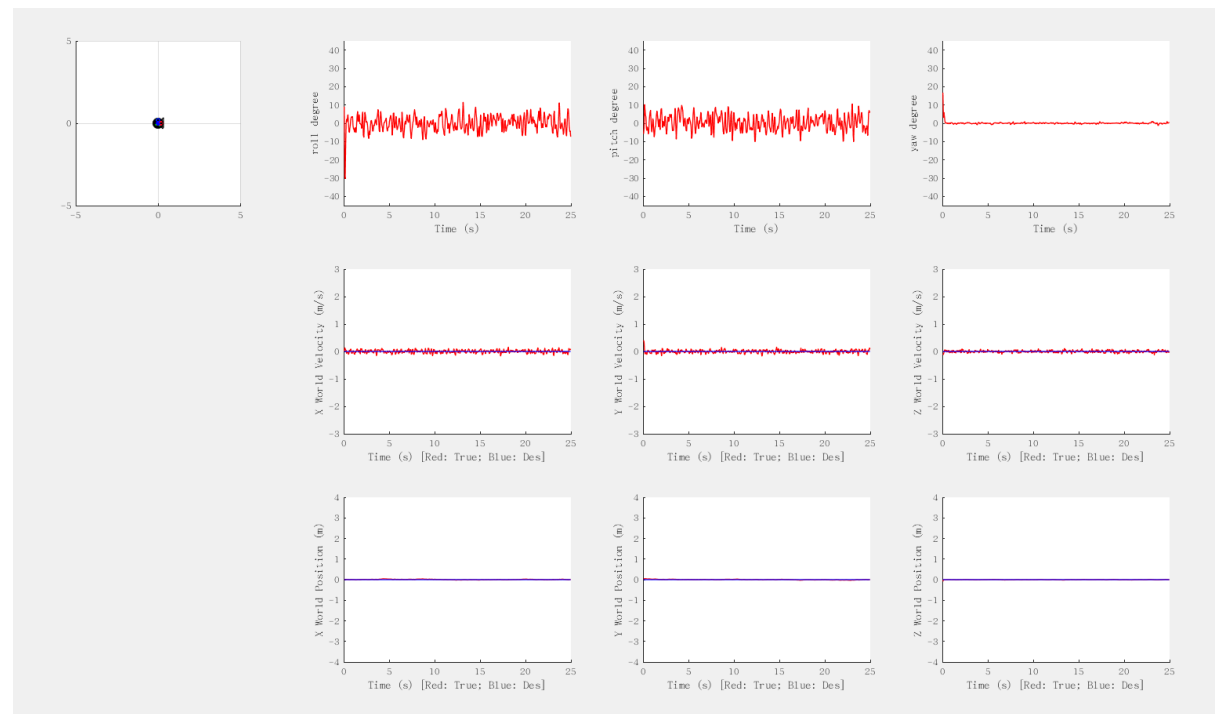
输入: 当前状态 $s$ , 期望状态 $s_{des}$

已知量: 质量 $m$ , 重力加速度 $g$ , 转动惯量 $I$

输出: 力 $F$ 、力矩 $M$

详细可查看文件夹中的"README.txt"

\*linear\_controller.m作为线性控制器的参考答案



# MATLAB仿真

名称	修改日期	类型	大小
 readonly	2021/8/11 0:33	文件夹	
 utils	2021/8/11 0:33	文件夹	
 circle_trajectory.m	2021/3/16 20:51	M 文件	1 KB
 controller.m	2021/3/16 20:51	M 文件	1 KB
 diamond_trajectory.m	2021/3/16 20:51	M 文件	2 KB
 hover_trajectory.m	2021/3/16 20:51	M 文件	1 KB
 test_trajectory.m	2021/4/9 20:14	M 文件	1 KB

# MATLAB仿真

使用“test\_trajectory.m”启动程序

名称

readonly

utils

circle\_trajectory.m

controller.m

diamond\_trajectory.m

hover\_trajectory.m

test\_trajectory.m

test\_trajectory.m

```
close all;
clear all;
clc;
addpath('./utils','./readonly');
figure(1)
h1 = subplot(3,4,1);
h2 = subplot(3,4,2);
h3 = subplot(3,4,3);
h4 = subplot(3,4,4);
h5 = subplot(3,4,6);
h6 = subplot(3,4,7);
h7 = subplot(3,4,8);
h8 = subplot(3,4,10);
h9 = subplot(3,4,11);
h10 = subplot(3,4,12);
set(gcf, 'Renderer', 'painters');
```

% Run Trajectory three trajectories, test one by one

```
run_trajectory_readonly(h1, h2, h3, h4, h5, h6, h7, h8, h9, h10, @hover_trajectory);
```

# MATLAB仿真

controller.m: 需要编写代码的部分。  
给定当前和期望的状态向量，计算力和力矩。

名称

readonly

utils

circle\_trajectory.m

controller.m

diamond\_trajectory.m

hover\_trajectory.m

test\_trajectory.m

controller.m

```
function [F, M] = controller(t, s, s_des)
```

```
global params
```

```
m = params.mass;
```

```
g = params.grav;
```

```
I = params.I;
```

控制器算法部分

```
F = 1.0; M = [0.0, 0.0, 0.0]'; % You should calculate the output F and M
```

```
end
```


# MATLAB仿真


文件夹“readonly”:


quadModel\_readonly.m: 一架500g四旋翼的物理参数

quadEOM\_readonly.p: 四旋翼动力学模型


run\_trajectory\_readonly.m: 迭代求解动力方程, 运行控制器代码, 并进行可视化。


 quadEOM\_readonly.m


 quadModel\_readonly.m


 run\_trajectory\_readonly.m


名称


 readonly


 utils

 circle\_trajectory.m

 controller.m

 diamond\_trajectory.m

 hover\_trajectory.m

 test\_trajectory.m

# MATLAB仿真

文件夹“utils”：一些有用的函数，例如四元数转换。

名称

readonly

utils

circle\_trajectory.m

controller.m

diamond\_trajectory.m

hover\_trajectory.m

test\_trajectory.m

quaternion\_to\_R.m

QuatToRot.m

R\_to\_quaternion.m

RotToRPY\_ZXY.m

ypr\_to\_R.m

# MATLAB仿真

\*\_trajectory.m: 根据路径设计四旋翼的轨迹。根据当前的状态向量和时间计算期望的状态向量。

“hover\_trajectory.m”悬停轨迹

“diamond\_trajectory.m”和“circle\_trajectory.m”，分别是钻石形和圆形轨迹

名称

readonly

utils

circle\_trajectory.m

controller.m

diamond\_trajectory.m

hover\_trajectory.m

test\_trajectory.m

circle\_trajectory.m

```
function s_des = circle_trajectory(t, true_s)

    s_des = zeros(13,1);

    omega=25;
    x_des=4*cos(t*omega/180*pi);
    y_des=4*sin(t*omega/180*pi);
    z_des=3/25*t;

    x_vdes=-omega/180*pi*4*sin(t*omega/180*pi);
    y_vdes= omega/180*pi*4*cos(t*omega/180*pi);
    z_vdes=3/25;

    s_des(1)=x_des;
    s_des(2)=y_des;
    s_des(3)=z_des;
    s_des(4)=x_vdes;
    s_des(5)=y_vdes;
    s_des(6)=z_vdes;

    %desired yaw angle in the flight
    des_yaw = mod(0.1 * pi * t, 2 * pi);
    ypr = [des_yaw, 0.0, 0.0];
    Rot = ypr_to_R(ypr);
    q_des = R_to_quaternion(Rot);
    s_des(7:10) = q_des;

end
```

# MATLAB仿真

示例

1.悬停

2.圆形和钻石形轨迹