



智能控制技术 实验报告

实验名称	HW03 模糊控制
实验地点	教 7 304
姓 名	PhilFan
学 号	19260817
实验日期	January 10, 2025
指导老师	刘山

Contents

1 实验目的和要求	1
1.1 实验目标	1
2 问题建模分析	2
2.1 状态方程	2
2.2 物理系统的 S-function 实现	3
3 算法设计	5
3.1 模糊规则的建立与选择 fuzzy rules	6
3.2 模糊规则矩阵	8
3.3 模糊规则的调整	10
4 实验结果表现与分析	12
4.1 $m = 0.1kg$	13
5 实验探究与可探索方向	15
5.1 动画仿真	15
A 附录 1: 源程序代码	16
B 附录 2: 版本更新记录	22

1 实验目的和要求

1.1 实验目标

(1). 系统方程:

$$F - G = m \frac{d^2 X}{dt^2} \quad (1)$$

其中, F 为电磁吸力, m 为钢球的质量, 重力 $G = mg$, g 为重力加速度。

(2). 电磁力方程:

$$F = K \left(\frac{I}{X} \right)^2 \quad (2)$$

其中, K 为电磁力系数。

(3). 电磁线圈方程:

$$U - K \frac{I}{X} \frac{dX}{dt} = L \frac{dI}{dt} + IR \quad (3)$$

其中, U 为控制电压, L 为电感, R 为线圈电阻。

假定系统参数如下表所示:

参数	值
m	0.05kg
g	9.81m/s ²
K	0.005 Nm ² /A ²
R	5Ω
L	0.01H

请完成以下任务:

Question.1.

- (1). **推导磁悬浮系统的状态空间模型:** (提示: 以钢球位置 X 、速度 \dot{X} 和电流 I 为状态变量)
- (2). **针对上述磁悬浮系统, 设计模糊控制器使钢球位置稳定在期望位置 $X_d = 0.05m$ 。**
假设初始钢球位置为 $X(0) = 0.03m$, 初始速度和初始电流均为 0, 仿真实现系统的模糊控制, 绘制钢球位置随时间变化曲线、控制电压随时间变化曲线, 并分析仿真结果。(输入输出的论域范围自行选择, 可尝试位置误差范围 $[-0.04, 0.04]m$, 位置误差变化率范围 $[-0.5, 0.5]m/s$, 控制电压的范围 $[-10, 10]V$)
- (3). **若改变钢球质量为 0.1kg, 其他参数不变, 重新进行仿真并分析对系统控制性能的影响, 讨论如何调整模糊控制器参数以适应钢球质量的变化。**

2 问题建模分析

(1). 系统的非线性特征

磁悬浮系统作为一个典型的非线性复杂系统，磁悬浮系统可以通过两种方式进行控制：一是在特定工作点进行局部线性化后应用线性控制方法，二是直接采用非线性控制理论进行系统设计。

(2). 系统的不确定性

磁悬浮系统运行过程中存在模型误差、电磁干扰以及其他外部环境因素的影响。

(3). 系统的开环不稳定性

系统仅在电磁力与重力达到平衡时存在唯一的稳定状态，且这种平衡必须在闭环控制下才能维持。在开环状态下，即使极小的外部扰动也会导致系统失去平衡

2.1 状态方程

根据系统的动力学方程，我们可以推导状态空间模型。选取状态变量：

$$\begin{aligned} x_1 &= X \\ x_2 &= \dot{X} = \frac{dX}{dt} \\ x_3 &= I \end{aligned} \quad (4)$$

从机械运动方程可得：

$$m \frac{d^2 X}{dt^2} = K \left(\frac{I}{X} \right)^2 - mg \quad (5)$$

整理得到状态方程：

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{K}{m} \left(\frac{x_3}{x_1} \right)^2 - g \\ \dot{x}_3 &= \frac{1}{L} \left(U - K \frac{x_3}{x_1} x_2 - R x_3 \right) \end{aligned} \quad (6)$$

其中， x_1 为钢球位置， x_2 为钢球速度， x_3 为电流。这是一个非线性系统的状态空间表达式。

2.2 物理系统的 S-function 实现

根据系统的动力学方程，和上述的状态方程，可以写出物理系统的 S-function 表达。

Listing 1: 物理系统的 S-function 实现

```
1 function [sys, x0, str, ts, simStateCompliance] = mdlInitializeSizes()
2     % 初始化回调子函数
3
4     sizes = simsizes;
5     sizes.NumContStates = 3; % 连续状态数
6     sizes.NumDiscStates = 0; % 离散状态数: [theta_min, cost_function]
7     sizes.NumOutputs = 2; % 输出个数
8     sizes.NumInputs = 1; % 输入个数
9     sizes.DirFeedthrough = 0; % 允许直馈通道
10    sizes.NumSampleTimes = 1; % 采样时间数
11    sys = simsizes(sizes); % 返回 sizes 数据结构
12    x0 = [0.03 0 0]; % 初始状态:
13    str = []; % 保留参数
14    ts = [0 0]; % 采样时间
15    simStateCompliance = 'UnknownSimState'; % 仿真状态合规性
16 end
17
18 function sys = mdlDerivatives(t, x, u, m, g, K, R, L)
19     % 导数回调子函数 (不使用, 空实现)
20     xx = x(1);
21     dx = x(2);
22     I = x(3);
23
24
25     x_dot = dx;
26     x_dot2 = (K * I^2 / xx^2 - m*g)/m;
27     I_dot = (u - K*I/xx*dx - I*R)/L;
28
29     %disp(u)
30     %disp([x_dot; x_dot2; I_dot]);
31     %disp(xx);
32     sys = [x_dot; x_dot2; I_dot];
33 end
```

这里由于是实现物理系统，所以 s-funtion 可以直接使用非线性的公式，无需进行线性化处理。

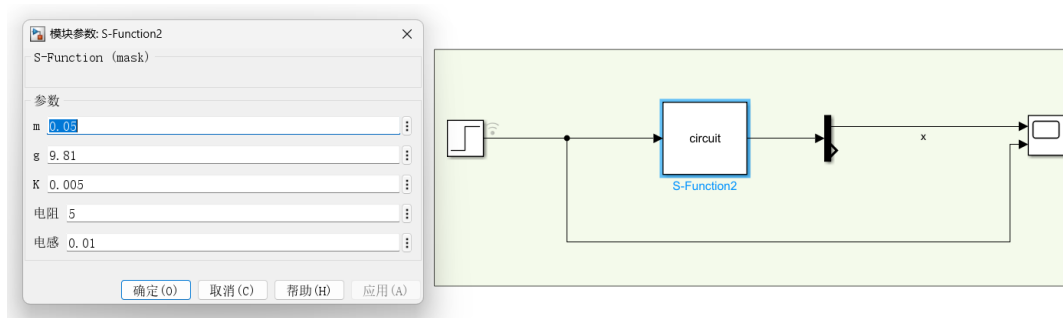


Figure 1: 搭建的物理模型图

搭建好物理模型后，我使用之前学习的 Mask 的方法，为模型添加了可以直接在 simulink 中修改参数的界面，无需进入代码即可修改参数

3 算法设计

模糊控制的基本概念

模糊控制的定义

- 是将模糊数学理论应用于自动控制领域的控制方法
- 基于模糊集合理论、模糊语言变量及模糊逻辑推理

工作原理

- 通过观察过程输出精确量转化为模糊量
- 经过人脑思维与逻辑推理进行模糊判决
- 将判决结果的模糊量转化为精确量

控制特性

- 属于非线性控制
- 不依赖于控制对象的精确模型
- 仅依靠少量控制规则
- 具有较强的鲁棒性
- 适用于数学模型未知、复杂的非线性系统控制

根据系统逻辑，将系统 slx 模型搭建如下

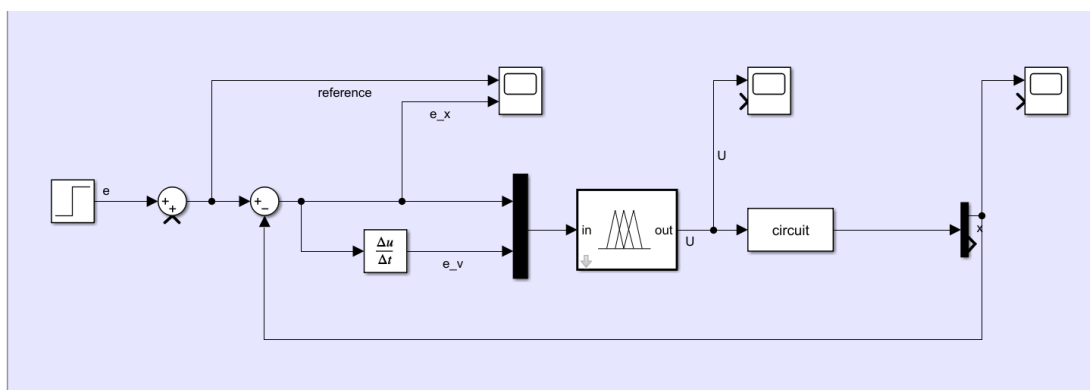


Figure 2: 系统模型图

快

3.1 模糊规则的建立与选择 fuzzy rules

在刚建立的时候，我感觉上课讲的步骤都是在 ui 界面下进行操作，所以就上网搜索了一下如何使用代码建立。

x, v 和 U 我都设置的是高斯函数

```
1 clear;
2 close all;
3
4 % 创建模糊推理系统，使用 newfis
5 fuzzyController = mamfis( ...
6     'NumInputs',1,'NumInputMFs',2,...
7     'NumOutputs',1,'NumOutputMFS',2,...
8     'AddRule','none');
9
10 % 定义输入变量 $x$ （位置误差）及其隶属度函数，范围  $[-0.04, 0.04]$ 
11 fuzzyController.Inputs(1).Name = 'x';
12 fuzzyController.Inputs(1).Range = [-0.04 0.04];
13
14 fuzzyController = addMF(fuzzyController, 'input', 1, 'EN', 'gaussmf',
15     [0.01 -0.04]); % EN: Extremely Negative
16 fuzzyController = addMF(fuzzyController, 'input', 1, 'VN', 'gaussmf',
17     [0.01 -0.03]); % VN: Very Negative
18 fuzzyController = addMF(fuzzyController, 'input', 1, 'N', 'gaussmf',
19     [0.01 -0.02]); % N: Negative
20 fuzzyController = addMF(fuzzyController, 'input', 1, 'NZ', 'gaussmf',
21     [0.01 -0.01]); % Z: Zero
22 fuzzyController = addMF(fuzzyController, 'input', 1, 'PZ', 'gaussmf',
23     [0.01 0.01]); % Z: Zero
24 fuzzyController = addMF(fuzzyController, 'input', 1, 'P', 'gaussmf',
25     [0.01 0.02]); % P: Positive
26 fuzzyController = addMF(fuzzyController, 'input', 1, 'VP', 'gaussmf',
27     [0.01 0.03]); % VP: Very Positive
28 fuzzyController = addMF(fuzzyController, 'input', 1, 'EP', 'gaussmf',
29     [0.01 0.04]); % EP: Extremely Positive
30
31 % 可视化 $x$ 的隶属度函数
32 figure;
33 subplot(311);
```



```
26 plotmf(fuzzyController, 'input', 1);
27 title('位置误差 (x) 的隶属度函数');
28
29 % 定义输入变量  $dx/dt$  (位置误差变化率) 及其隶属度函数, 范围  $[-0.5, 0.5]$ 
30 fuzzyController.Inputs(2).Name = 'dx';
31 fuzzyController.Inputs(2).Range = [-0.5 0.5];
32 fuzzyController = addMF(fuzzyController, 'input', 2, 'EN', 'gaussmf',
    [0.1 -0.5]); % EN: Extremely Negative
33 fuzzyController = addMF(fuzzyController, 'input', 2, 'VN', 'gaussmf',
    [0.1 -0.4]); % VN: Very Negative
34 fuzzyController = addMF(fuzzyController, 'input', 2, 'N', 'gaussmf', [0.1
    -0.3]); % N: Negative
35 fuzzyController = addMF(fuzzyController, 'input', 2, 'NZ', 'gaussmf',
    [0.1 -0.1]); % Z: Zero
36 fuzzyController = addMF(fuzzyController, 'input', 2, 'PZ', 'gaussmf',
    [0.1 0.1]); % Z: Zero
37 fuzzyController = addMF(fuzzyController, 'input', 2, 'P', 'gaussmf', [0.1
    0.3]); % P: Positive
38 fuzzyController = addMF(fuzzyController, 'input', 2, 'VP', 'gaussmf',
    [0.1 0.4]); % VP: Very Positive
39 fuzzyController = addMF(fuzzyController, 'input', 2, 'EP', 'gaussmf',
    [0.1 0.5]); % EP: Extremely Positive
40
41 % 可视化  $dx/dt$  的隶属度函数
42
43 subplot(312);
44 plotmf(fuzzyController, 'input', 2);
45 title('位置误差变化率 (dx/dt) 的隶属度函数');
46
47 % 定义输出变量  $U$  (控制电压) 的隶属度函数, 范围  $[-10, 10]$ 
48 fuzzyController = addvar(fuzzyController, 'output', 'U', [-10 10]);
49 fuzzyController = addMF(fuzzyController, 'output', 1, 'EL', 'trimf', [-10
    -10 -7]); % EL: Extremely Low
50 fuzzyController = addMF(fuzzyController, 'output', 1, 'VL', 'trimf', [-10
    -7 -4]); % VL: Very Low
51 fuzzyController = addMF(fuzzyController, 'output', 1, 'L', 'trimf', [-7
    -4 -1]); % L: Low
52 fuzzyController = addMF(fuzzyController, 'output', 1, 'NZ', 'trimf', [-3
```

```

-1 1]); % NZ:negetive ZERO
53 fuzzyController = addMF(fuzzyController, 'output', 1, 'PZ', 'trimf', [-1
    1 3]); % M: positive ZERO
54 fuzzyController = addMF(fuzzyController, 'output', 1, 'H', 'trimf', [1 4
    7]); % H: High
55 fuzzyController = addMF(fuzzyController, 'output', 1, 'VH', 'trimf', [4 7
    10]); % VH: Very High
56 fuzzyController = addMF(fuzzyController, 'output', 1, 'EH', 'trimf', [7
    10 10]); % EH: Extremely High

```

但是值得注意的是，根据调试信息可以看出，这种方法已经逐步被淘汰了，在接下来的版本当中将全被替换成为 ui 界面操作

3.2 模糊规则矩阵

其实使用代码生成模糊矩阵比手动一个个点还是要快一些的，这也是我为什么选择这种方法的原因。

这里我使用了另一个课本中给出的一个参考 8x8 矩阵作为调参的 backbone，在下一节中，我将在这个的基础上进行调参和修改。

Listing 2: 生成规则矩阵

```

1 fuzzyController = readfis('Controller4.fis');
2
3 table = [
4     [1, 1, 1, 1, 6, 5, 5, 5],
5     [1, 2, 2, 1, 7, 5, 5, 5],
6     [2, 2, 2, 1, 2, 6, 6, 6],
7     [2, 3, 3, 2, 7, 6, 6, 6],
8     [3, 3, 3, 2, 8, 7, 6, 7],
9     [3, 3, 4, 2, 8, 7, 7, 7],
10    [3, 4, 4, 2, 8, 8, 7, 8],
11    [4, 4, 4, 4, 8, 8, 8, 8]];
12 % 生成规则矩阵
13 rules = [];
14 for i = 1:8
15     for j = 1:8
16         disp(table(i,j));
17         output = table(i,j);

```

```
18         rules = [rules; i j output 1 1]; % 1 1表示使用 'min' 合成和 '
           centroid' 解模糊
19     end
20 end
21
22 % disp(rules)
23
24 % 添加规则到模糊系统
25 fuzzyController = addrule(fuzzyController, rules);
26 %showrule(fuzzyController, 'Format', 'symbolic');
```

Listing 3: 可视化规则

```
1 % 可视化规则
2 ruleview(fuzzyController);
3
4 %figure;
5 %plotfis(fuzzyController);
6
7 % 输出 surface
8 figure;
9 gensurf(fuzzyController, [1,2], 1);
10 saveas(gcf, 'surf.jpg');
11
12 % 保存为 .fis 文件
13 writefis(fuzzyController, 'Controller41.fis');
```

我们可以将 surf 打印出来。

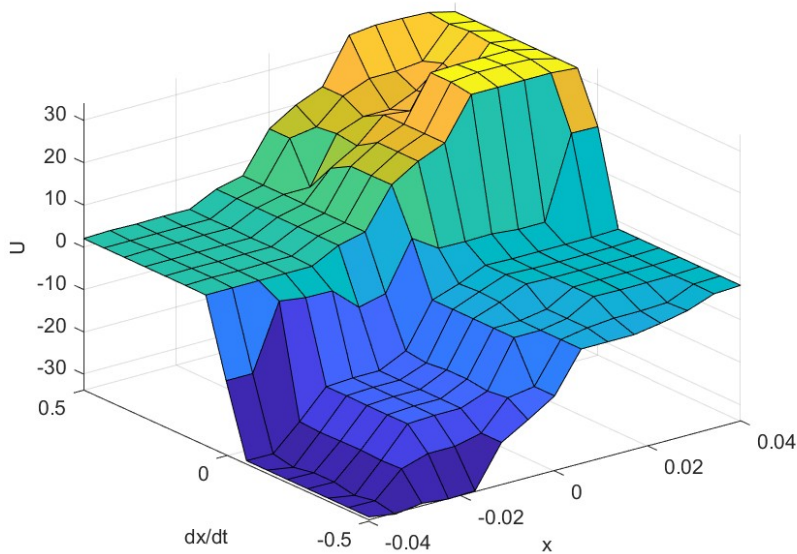


Figure 3: 模糊控制平面

3.3 模糊规则的调整

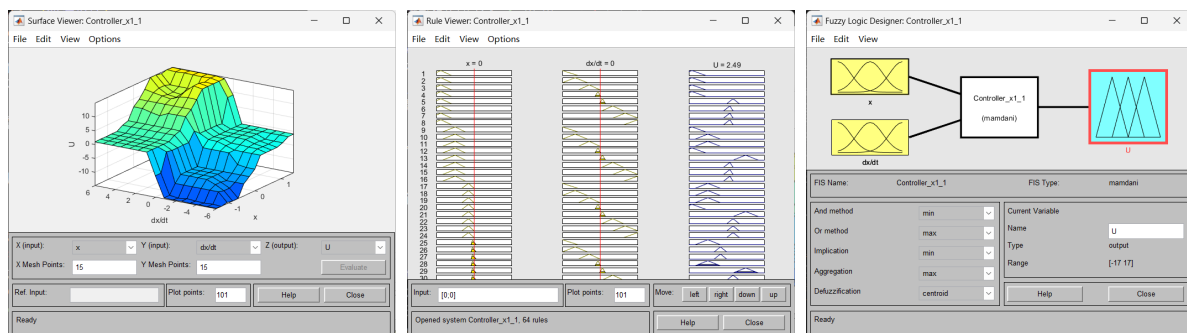
建立了初步的规则，我就在系统上进行了尝试，但是发现效果并不好，经常会出现不收敛、不稳定、震荡大等问题。

所以需要对规则进行调整。

经过反复多次的修改和尝试，最后得到了一版规则

circuit.m	2024/12/4 22:10	MATLAB Code	3 KB
Controller.fis	2024/12/4 16:31	FIS 文件	4 KB
Controller_x1.fis	2024/12/4 22:08	FIS 文件	3 KB
Controller_x1_1.fis	2024/12/13 21:04	FIS 文件	3 KB
Controller_x2.fis	2024/12/4 22:15	FIS 文件	3 KB
Controller31.fis	2024/12/4 17:43	FIS 文件	3 KB
Controller33.fis	2024/12/4 21:14	FIS 文件	3 KB
Controller41.fis	2024/12/4 17:41	FIS 文件	3 KB
Controller41fis.fis	2024/12/4 21:17	FIS 文件	3 KB
Controller50.fis	2024/12/4 17:30	FIS 文件	4 KB
Controller51fis.fis	2024/12/4 22:29	FIS 文件	3 KB
Controller60.fis	2024/12/4 17:27	FIS 文件	4 KB

Figure 4: 经过了反复修改和迭代模型

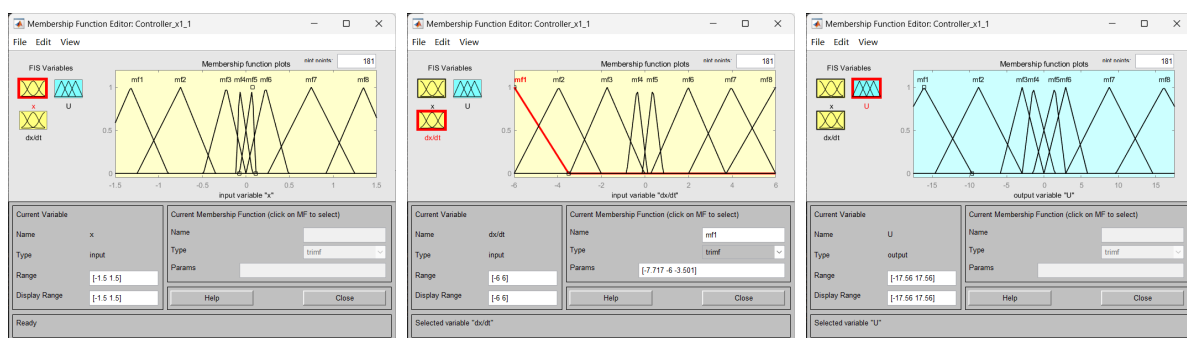


(a) surface 图片

(b) rules 图片

(c) fis 文件

Figure 5: 最终 fis 文件



(a) x

(b) v

(c) U

Figure 6: 最终 fis 文件

4 实验结果表现与分析

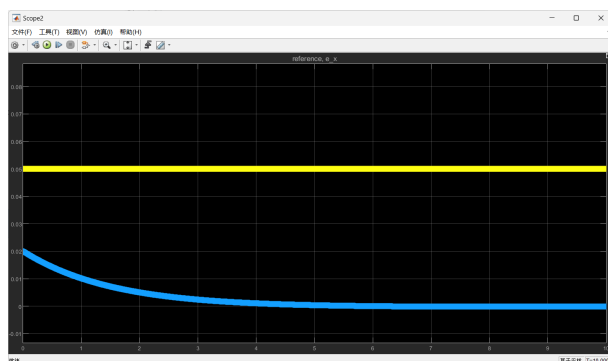


Figure 7: reference add e_x

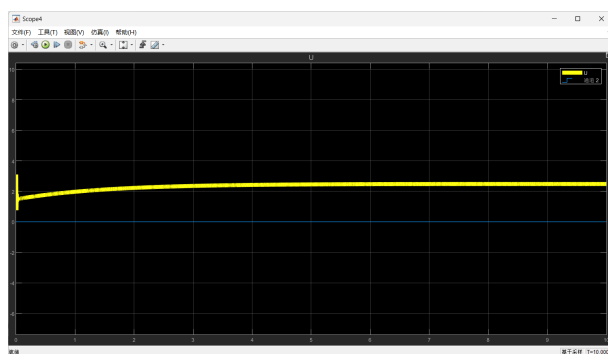


Figure 8: U

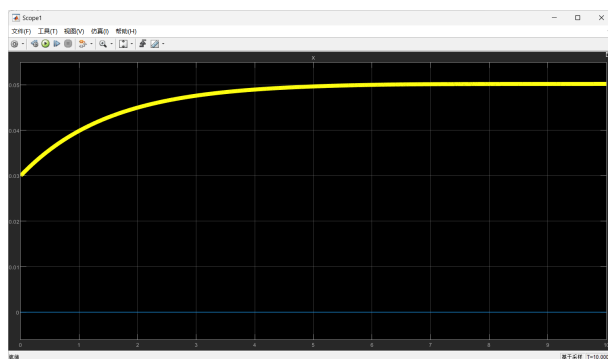


Figure 9: x

从上面的仿真结果可以看出：

- (1). 系统响应性能良好。从图中可以看出，系统能够快速跟踪给定的参考输入信号，位置误差在短时间内就能收敛到参考位置附近。
- (2). 控制输出平稳。控制电压 U 的变化比较平滑，没有出现剧烈的震荡，这说明模糊控制器的规则库设计合理，控制作用适中。

- (3). 稳态精度高。在稳态时, 位置 x 能够很好地跟随参考输入, 稳态误差很小, 满足控制要求。

总的来说, 所设计的模糊控制器实现了对磁悬浮系统的有效控制, 系统表现出良好的动态性能和稳态性能。这验证了模糊控制在处理非线性系统时的优越性。

4.1 $m = 0.1kg$

更改到 $m = 0.1kg$ 后, 经过调参以后, 系统表现如下

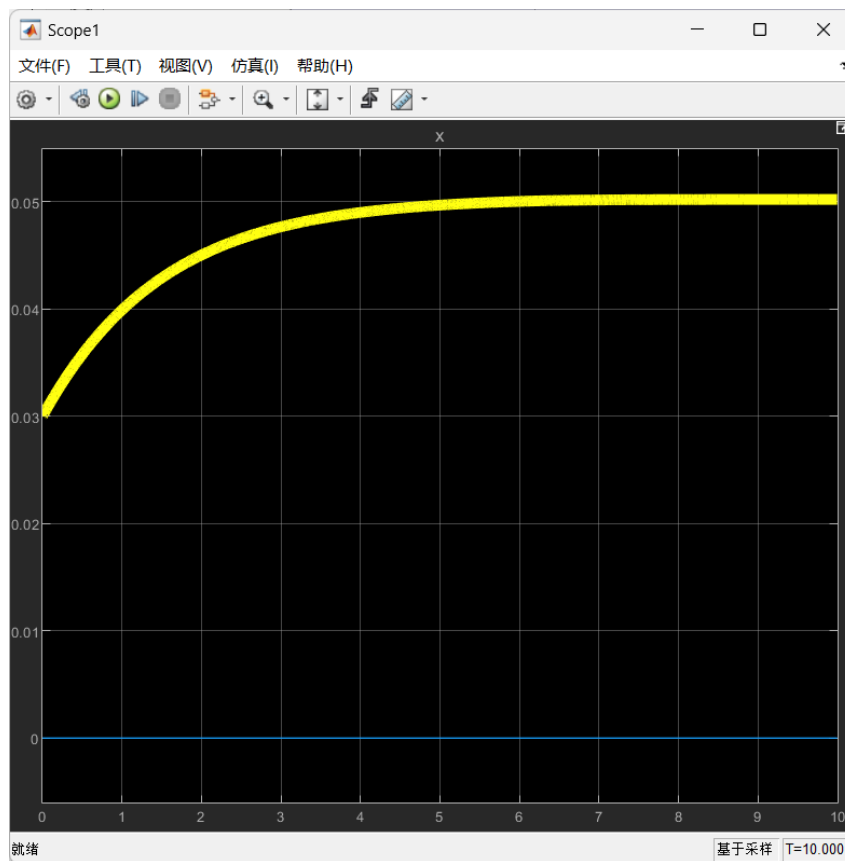


Figure 10: $m=0.1kg$

可以看出系统可以稳定悬浮。

从图11可以看出, 当钢球质量增加到 $0.1kg$ 时, 系统仍然能够实现稳定悬浮, 但是相比于 $m=0.05kg$ 时:

- (1). 系统响应变慢。由于质量增大, 系统的惯性也随之增大, 导致系统响应速度变慢。
- (2). 控制电压增大。为了克服更大的重力, 控制器需要输出更大的电压来产生足够的电磁力。

为了适应钢球质量的变化, 主要采取了以下调整措施:

- (1). 扩大控制电压的论域范围。原先控制电压的范围是 $[-20,20]\text{V}$ ，为了适应更大的质量，将范围调整为 $[-27,27]\text{V}$ 。这样可以提供更大的控制作用来克服增大的重力。
- (2). 保持位置误差和速度误差的论域几乎不变
- (3). 适当调整了模糊规则中的输出值大小，使控制作用更强。

这些调整措施使得控制器能够较好地适应质量变化，保持系统的基本性能。但是从长远来看，更理想的解决方案是设计自适应模糊控制器，能够根据系统参数的变化自动调整控制参数。

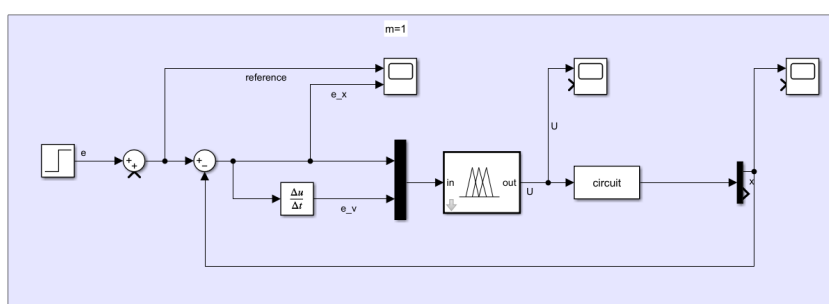


Figure 11: $m=0.1\text{kg}$

5 实验探究与可探索方向

5.1 动画仿真

为了得到小球的真实运动轨迹，我还学习了 simulink 的 3D Animation 库，并把小球位置作为参数输入了 VR 3D 中，最后效果如下图，还是比较有意思的。

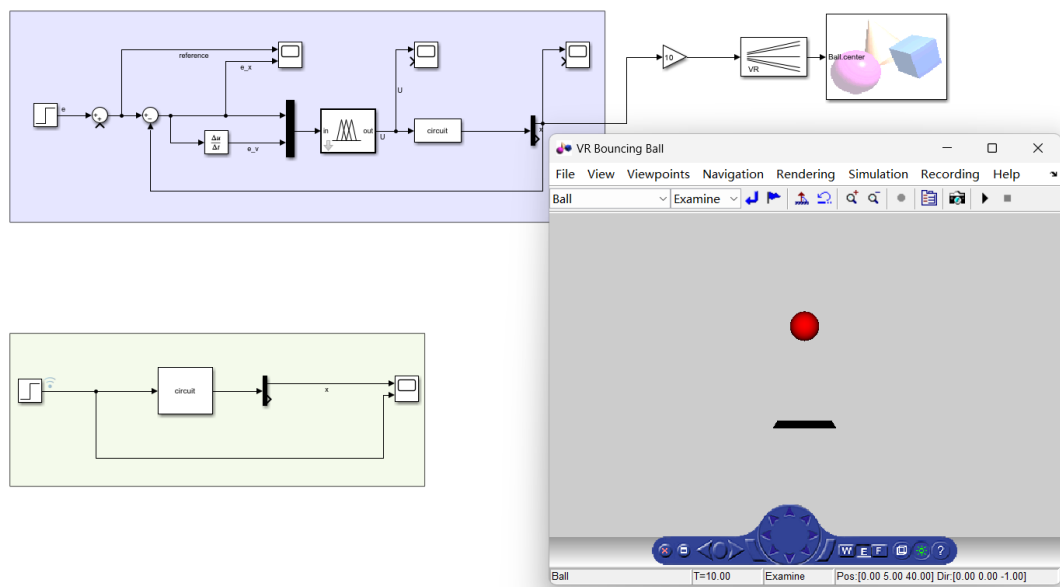


Figure 12: 3D Animation 库

A 附录 1：源程序代码

resources	资源文件夹
circuit.m	磁悬浮系统仿真代码
Controller_x1_1.fis	模糊控制器配置文件
fis.m	模糊控制系统生成
fuzz.slx	Simulink 仿真模型

Listing 4: fis.m

```

1 clear;
2 close all;
3
4 %% 创建模糊推理系统, 使用 newfis
5 % fuzzyController = mamfis( ...
6 %     'NumInputs',1,'NumInputMFs',2,...
7 %     'NumOutputs',1,'NumOutputMFS',2,...
8 %     'AddRule','none');
9 %
10 %% 定义输入变量x (位置误差) 及其隶属度函数, 范围 [-0.04, 0.04]
11 % fuzzyController.Inputs(1).Name = 'x';
12 % fuzzyController.Inputs(1).Range = [-0.04 0.04];
13 %
14 % fuzzyController = addMF(fuzzyController, 'input', 1, 'EN', 'gaussmf',
    [0.01 -0.04]); % EN: Extremely Negative
15 % fuzzyController = addMF(fuzzyController, 'input', 1, 'VN', 'gaussmf',
    [0.01 -0.03]); % VN: Very Negative
16 % fuzzyController = addMF(fuzzyController, 'input', 1, 'N', 'gaussmf',
    [0.01 -0.02]); % N: Negative
17 % fuzzyController = addMF(fuzzyController, 'input', 1, 'NZ', 'gaussmf',
    [0.01 -0.01]); % Z: Zero
18 % fuzzyController = addMF(fuzzyController, 'input', 1, 'PZ', 'gaussmf',
    [0.01 0.01]); % Z: Zero
19 % fuzzyController = addMF(fuzzyController, 'input', 1, 'P', 'gaussmf',
    [0.01 0.02]); % P: Positive
20 % fuzzyController = addMF(fuzzyController, 'input', 1, 'VP', 'gaussmf',
    [0.01 0.03]); % VP: Very Positive
21 % fuzzyController = addMF(fuzzyController, 'input', 1, 'EP', 'gaussmf',
    [0.01 0.04]); % EP: Extremely Positive

```

```
22 %
23 % % 可视化 $x$ 的隶属度函数
24 % figure;
25 % subplot(311);
26 % plotmf(fuzzyController, 'input', 1);
27 % title('位置误差 ( $x$ ) 的隶属度函数');
28 %
29 % % 定义输入变量 $dx/dt$  (位置误差变化率) 及其隶属度函数, 范围  $[-0.5, 0.5]$ 
30 % fuzzyController.Inputs(2).Name = 'dx';
31 % fuzzyController.Inputs(2).Range = [-0.5 0.5];
32 % fuzzyController = addMF(fuzzyController, 'input', 2, 'EN', 'gaussmf',
    [0.1 -0.5]); % EN: Extremely Negative
33 % fuzzyController = addMF(fuzzyController, 'input', 2, 'VN', 'gaussmf',
    [0.1 -0.4]); % VN: Very Negative
34 % fuzzyController = addMF(fuzzyController, 'input', 2, 'N', 'gaussmf',
    [0.1 -0.3]); % N: Negative
35 % fuzzyController = addMF(fuzzyController, 'input', 2, 'NZ', 'gaussmf',
    [0.1 -0.1]); % Z: Zero
36 % fuzzyController = addMF(fuzzyController, 'input', 2, 'PZ', 'gaussmf',
    [0.1 0.1]); % Z: Zero
37 % fuzzyController = addMF(fuzzyController, 'input', 2, 'P', 'gaussmf',
    [0.1 0.3]); % P: Positive
38 % fuzzyController = addMF(fuzzyController, 'input', 2, 'VP', 'gaussmf',
    [0.1 0.4]); % VP: Very Positive
39 % fuzzyController = addMF(fuzzyController, 'input', 2, 'EP', 'gaussmf',
    [0.1 0.5]); % EP: Extremely Positive
40 %
41 % % 可视化 $dx/dt$ 的隶属度函数
42 %
43 % subplot(312);
44 % plotmf(fuzzyController, 'input', 2);
45 % title('位置误差变化率 ( $dx/dt$ ) 的隶属度函数');
46 %
47 % % 定义输出变量 $U$  (控制电压) 的隶属度函数, 范围  $[-10, 10]$ 
48 % fuzzyController = addvar(fuzzyController, 'output', 'U', [-10 10]);
49 % fuzzyController = addMF(fuzzyController, 'output', 1, 'EL', 'trimf',
    [-10 -10 -7]); % EL: Extremely Low
50 % fuzzyController = addMF(fuzzyController, 'output', 1, 'VL', 'trimf',
```

```

    [-10 -7 -4]); % VL: Very Low
51 % fuzzyController = addMF(fuzzyController, 'output', 1, 'L', 'trimf',
    [-7 -4 -1]); % L: Low
52 % fuzzyController = addMF(fuzzyController, 'output', 1, 'NZ', 'trimf',
    [-3 -1 1]); % NZ:negetive ZERO
53 % fuzzyController = addMF(fuzzyController, 'output', 1, 'PZ', 'trimf',
    [-1 1 3]); % M: positive ZERO
54 % fuzzyController = addMF(fuzzyController, 'output', 1, 'H', 'trimf', [1
    4 7]); % H: High
55 % fuzzyController = addMF(fuzzyController, 'output', 1, 'VH', 'trimf',
    [4 7 10]); % VH: Very High
56 % fuzzyController = addMF(fuzzyController, 'output', 1, 'EH', 'trimf',
    [7 10 10]); % EH: Extremely High
57 %
58 % % 可视化U的隶属度函数
59 %
60 % subplot(313);
61 % plotmf(fuzzyController, 'output', 1);
62 % title('控制电压 (U) 的隶属度函数');
63
64 fuzzyController = readfis('Controller4.fis');
65
66 table = [
67     [1, 1, 1, 1, 6, 5, 5, 5],
68     [1, 2, 2, 1, 7, 5, 5, 5],
69     [2, 2, 2, 1, 2, 6, 6, 6],
70     [2, 3, 3, 2, 7, 6, 6, 6],
71     [3, 3, 3, 2, 8, 7, 6, 7],
72     [3, 3, 4, 2, 8, 7, 7, 7],
73     [3, 4, 4, 2, 8, 8, 7, 8],
74     [4, 4, 4, 4, 8, 8, 8, 8]];
75 % 生成规则矩阵
76 rules = [];
77 for i = 1:8
78     for j = 1:8
79         disp(table(i,j));
80         output = table(i,j);
81         rules = [rules; i j output 1 1]; % 1 1表示使用 'min' 合成和 '

```

```

            centroid' 解模糊
82     end
83 end
84
85 % disp(rules)
86
87 % 添加规则到模糊系统
88 fuzzyController = addrule(fuzzyController, rules);
89 %showrule(fuzzyController, 'Format', 'symbolic');
90 % 可视化规则
91 ruleview(fuzzyController);
92
93 %figure;
94 %plotfis(fuzzyController);
95
96 % 输出 surface
97 figure;
98 gensurf(fuzzyController, [1,2], 1);
99 saveas(gcf, 'surf.jpg');
100
101
102
103 % 保存为 .fis 文件
104 writefis(fuzzyController, 'Controller41.fis');
105
106 testInputs = [0.01, -0.25]; % 例如, x=0.01, dx/dt=-0.25
107 output = evalfis(fuzzyController, testInputs);
108 disp(['控制电压U的输出: ', num2str(output)]);

```

Listing 5: 物理系统实现

```

1  function [sys, x0, str, ts, simStateCompliance] = circuit(t, x, u,
      flag, m, g, K, R, L)
2      % Critic S-Function
3      % 输入: theta, 目标值 theta_target (无实际意义),
4      % 输出: cost_function (基于最小 theta 的最大化)
5
6      switch flag
7          case 0

```

```
8         [sys, x0, str, ts, simStateCompliance] = mdlInitializeSizes();
9     case 1
10         sys = mdlDerivatives(t, x, u,m,g,K,R,L);
11     case 2
12         sys = mdlUpdate(t, x, u);
13     case 3
14         sys = mdlOutputs(t, x, u);
15     case 4
16         sys = mdlGetTimeOfNextVarHit(t, x, u);
17     case 9
18         sys = mdlTerminate(t, x, u);
19     otherwise
20         DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag))
21         ;
22
23 end
24
25 %% 子函数定义
26
27 function [sys, x0, str, ts, simStateCompliance] = mdlInitializeSizes()
28     % 初始化回调子函数
29
30     sizes = simsizes;
31     sizes.NumContStates = 3; % 连续状态数
32     sizes.NumDiscStates = 0; % 离散状态数: [theta_min, cost_function]
33     sizes.NumOutputs = 2; % 输出个数
34     sizes.NumInputs = 1; % 输入个数
35     sizes.DirFeedthrough = 0; % 允许直馈通道
36     sizes.NumSampleTimes = 1; % 采样时间数
37     sys = simsizes(sizes); % 返回sizes数据结构
38     x0 = [0.03 0 0]; % 初始状态:
39     str = []; % 保留参数
40     ts = [0 0]; % 采样时间
41     simStateCompliance = 'UnknownSimState'; % 仿真状态合规性
42 end
43
44 function sys = mdlDerivatives(t, x, u,m,g,K,R,L)
45     % 导数回调子函数 (不使用, 空实现)
46     xx = x(1);
47     dx = x(2);
```

```
45         I = x(3);
46
47
48         x_dot = dx;
49         x_dot2 = (K * I^2 / xx^2 - m*g)/m;
50         I_dot = (u - K*I/xx*dx - I*R)/L;
51
52         %disp(u)
53         %disp([x_dot;x_dot2;I_dot]);
54         %disp(xx);
55         sys = [x_dot;x_dot2;I_dot];
56     end
57
58     function sys = mdlUpdate(t, x, u)
59         sys = [];
60     end
61
62     function sys = mdlOutputs(t, x, u)
63         sys = [x(1);x(2)];
64     end
65
66     function sys = mdlGetTimeOfNextVarHit(t, x, u)
67         % 计算下一个采样时间（定时采样）
68         sampleTime = 1; % 固定采样时间（1秒）
69         sys = t + sampleTime; % 下一次采样时间
70     end
71
72     function sys = mdlTerminate(t, x, u)
73         % 仿真结束时的回调（输出成本函数）
74         sys = [];
75     end
76 end
```

B 附录 2：版本更新记录

日期	更新内容
2024.12.4	初始化项目；完成基本的模糊控制器设计；搭建 Simulink 仿真模型
2024.12.8	优化模糊控制规则；调整隶属度函数参数；完善仿真模型
2024.12.10	添加 3D Animation 动画效果；完成不同质量下的对比实验
2024.12.13	完成论文撰写；添加实验数据分析；整理实验结果图表；完善文档格式

Table 1: 版本更新记录

References

- [1] weixin42686879. 完整的模糊推理系统介绍以及 matlab 中从零实现, 2020.
- [2] weixin42686879. 完整的模糊推理系统介绍以及 matlab 中从零实现, 2020.
- [3] 模糊控制方法在磁悬浮系统中的应用.
- [4] Generate code for fuzzy system using simulink coder.