

## 模糊控制

Author: 章翰宇

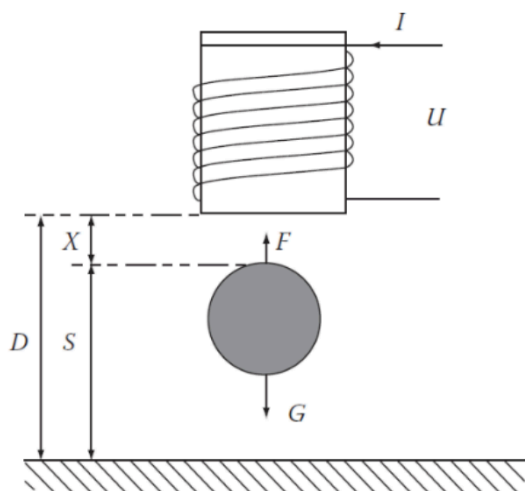
ID: 3220104133

### Abstract

如图所示的磁悬浮系统,钢球在电磁力和重力的共同作用下悬浮在空中,推导磁悬浮系统的状态空间模型;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;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影响。

**Keywords:** 智能控制, 模糊控制

### 物理建模



由于方程为 (为了取正值,  $x$  的正方向为向下, 因此需调整题目中的正负符号):

$$\begin{cases} -K \frac{I^2}{x^2} + mg = m \frac{d^2x}{dt^2} \\ U + K \frac{I}{x} \frac{dx}{dt} = L \frac{dI}{dt} + IR \end{cases}$$

容易知道, 整理成为状态空间方程, 直接以  $x_1 = x, x_2 = \dot{x}, I$  三个变量为状态变量即可:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{K}{m} \frac{I^2}{x_1^2} + g \\ \dot{I} = \frac{1}{L} (U + K \frac{I}{x_1} x_2 - IR) \end{cases}$$

其实写成便于画 Simulink 的方块图的话, 写为:

$$\begin{cases} \ddot{x} = g - \frac{K}{m} \frac{I^2}{x^2} \\ \dot{I} = \frac{1}{L} (U + K \frac{I}{x} \frac{dx}{dt} - IR) \end{cases}$$

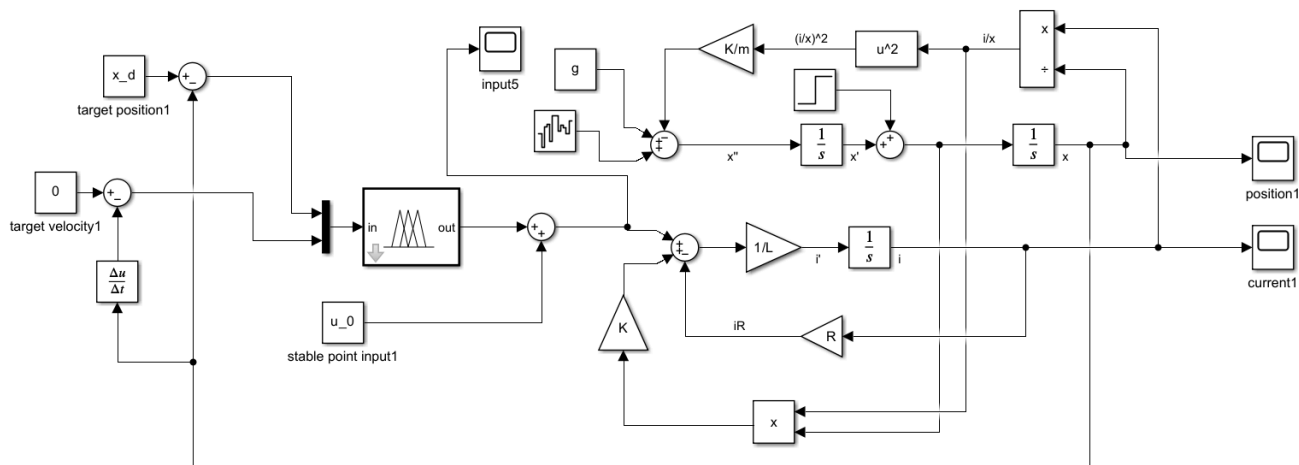
可以据此绘制 Simulink 中的模糊控制框图如下:

```

m = 0.05; g = 9.81; K = 0.005; R = 5; L = 0.01;
x_d = 0.05; x_0 = 0.03; u_0 = R * sqrt(m*g*x_d^2/K); %% 参数

```

### Simulink



## 模糊控制器设定

FIS 使用 **Mamdani** 类型，下面展示本次实验使用的模糊控制器：

### Fuzzification

模糊控制器的首个输入定义为位移偏移量  $\Delta x$ （即与目标稳态点差值），第二个输入为  $\Delta \dot{x}$ （即位移  $x$  的导数），现在，将位移偏移，和位移导数的论域按照如下定义，设计两者的各自的一组隶属度函数：

#### 位移偏移

其论域为  $[-0.04, 0.04]$ ，取 7 个分类，均用三角形函数：

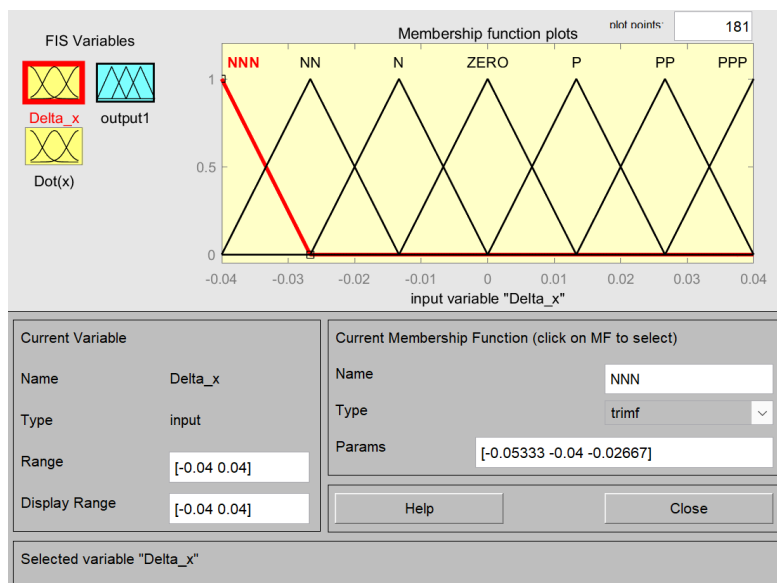


Figure 2:  $\Delta x$  的模糊规则

## 位移导数

其论域为 $[-0.5, 0.5]$ ，取 7 个分类，也均用三角形函数：

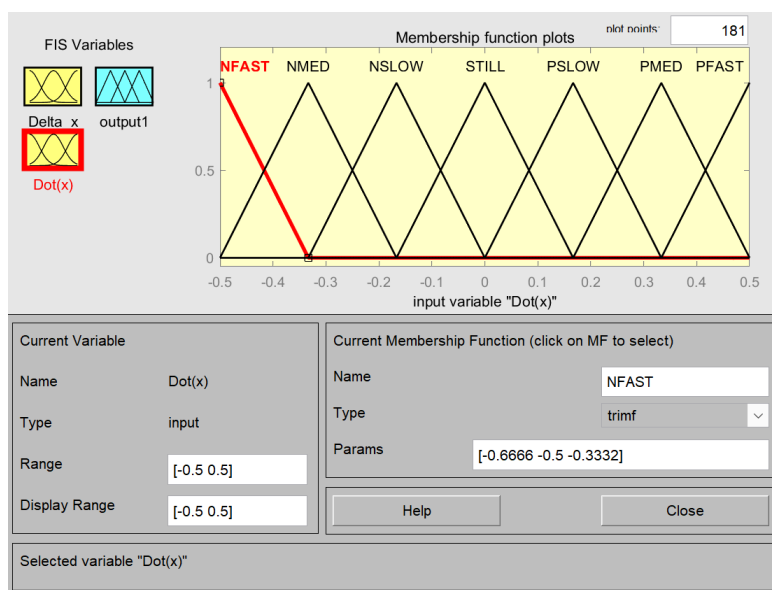


Figure 3:  $\Delta x$  的模糊规则

## 操作变量

输入的控制电压论域为 $[-10, 10]$ ，取了 9 个分类，也均使用三角形函数：

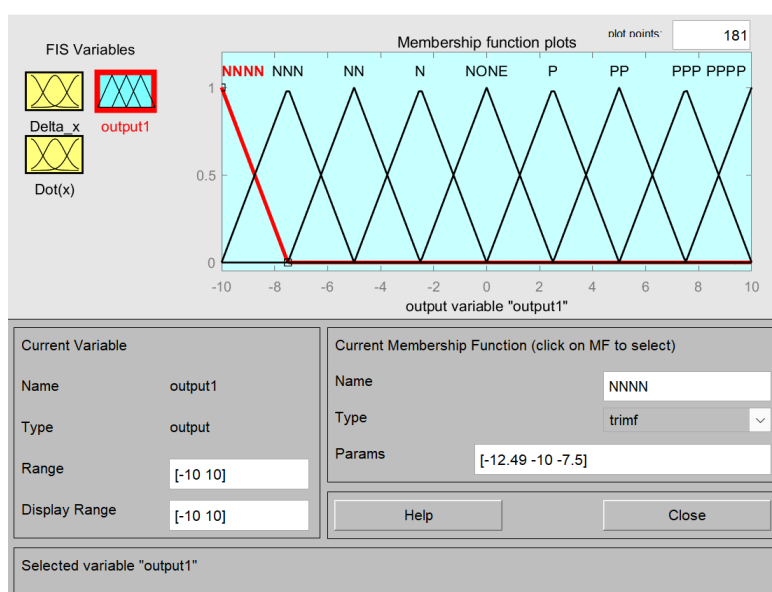


Figure 4:  $U$  的模糊规则

## Inference

在我的控制器模糊推断操作中，使用隶属度函数的max来作为 OR（或）操作，用min来作为 AND（和）操作，如下图所示：

And method	min	▼
Or method	max	▼
Implication	min	▼
Aggregation	max	▼
Defuzzification	centroid	▼

Figure 5: 模糊逻辑 推断规则

### Defuzzification

使用 centroid 方法（见上图），即解模糊化的方法是计算两个区域的面积再取重心的横坐标（参见后文效果图）

## 模糊控制规则

### Rules

以下是我的控制规则：

		$\Delta x$						
		NNN	NN	N	ZERO	P	PP	PPP
$\dot{x}$	NFAST	PPPP	PPPP	PPP	PP	P	NONE	N
	NMED	PPPP	PPP	PPP	P	NONE	N	NN
	NSLOW	PPP	PPP	PP	P	N	NN	NN
	STILL	PPP	PP	P	NONE	N	NN	NNN
	PSLOW	PP	PP	P	N	NN	NNN	NNN
	PMED	PP	P	NONE	N	NNN	NNN	NNNN
	PFAST	P	NONE	N	NN	NNN	NNNN	NNNN

该控制规则的设计思路是：“远则快速接近，近则微调”。对于位移和速度均有正偏差时（说明小球要掉下去了），以较大的正向输入进行调整，对于位移有负偏差而速度有正偏差时，说明已经在返回平衡点的路上了，因此只用施加少量的负向输入，甚至不需要施加输入。对于正反互换的情况完全同理，因此整张表呈现出中心对称的特点。（要注意正反作用的方向，因为输入控制器的是设定值与测量值的差值，有个负号）

### Visualization

这是 centroid 解模糊化规则的可视化：

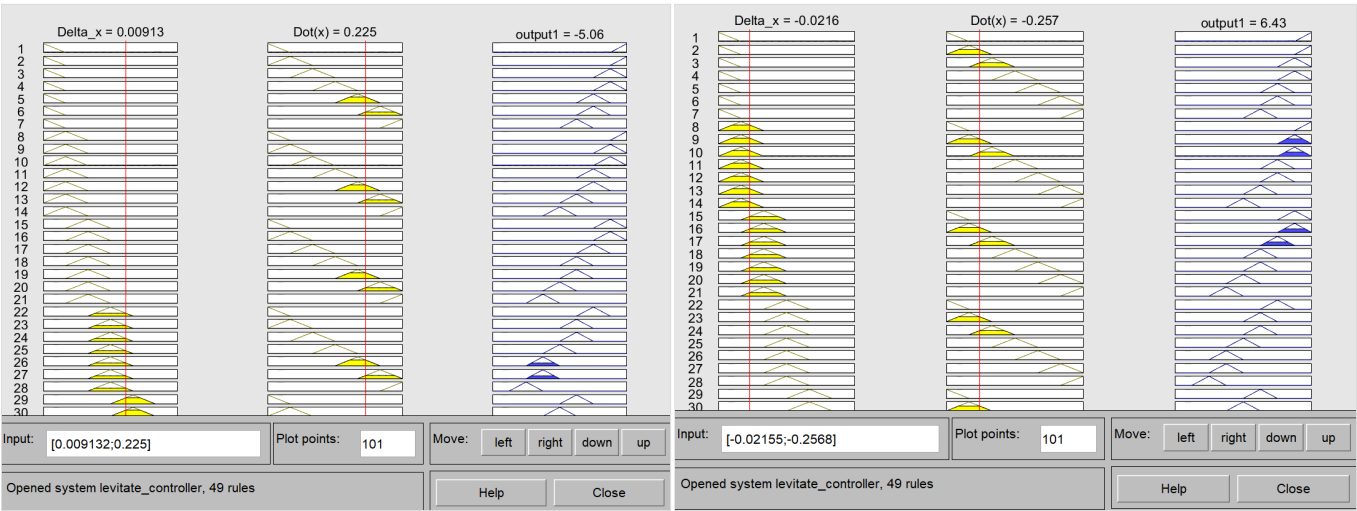


Table 1: 模糊规则的可视化

以下是通过控制曲面的方法可视化模糊控制器：

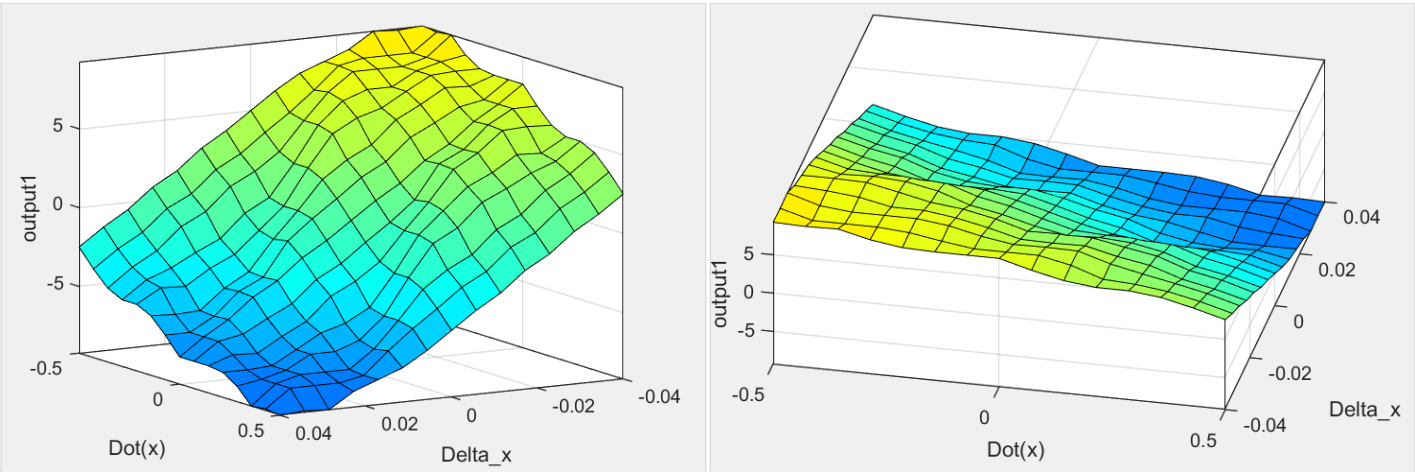


Table 2: 控制曲面

## 控制效果

### 稳态工作点讨论

使用模糊控制器时，注意模糊控制器类似于一个 PD 控制器，因此比较适合稳态工作点时操作变量为 0 的例子。

然而，欲使小球稳定悬浮，稳态工作点时操作变量不为 0。取方程的稳态解，发现 $u_0 = R\sqrt{mgx_d^2/K}$ 。为了达到较好的控制效果，需要在输入中加上这个 $u_0$ 的偏置（当然，智能控制本身的初衷是无需知道模型具体参数的，但在本例中，为了消除余差需要加上偏置）

### 抗扰动测试

如下，展示了该控制器在无噪音（理想情况）、有小噪音、大噪音下的控制效果：

可以看到（以无噪音为例），在距离平衡点较远的阶段，小球的位置曲线以较为快速的方式上升，接近平衡位置，但是在快到目标平衡点时，突然收到一个较大的电压输入峰，防止小球“冲刺过头”，速度很快缓了下来。平滑接近目标点。这正符合控制器的设计目标。

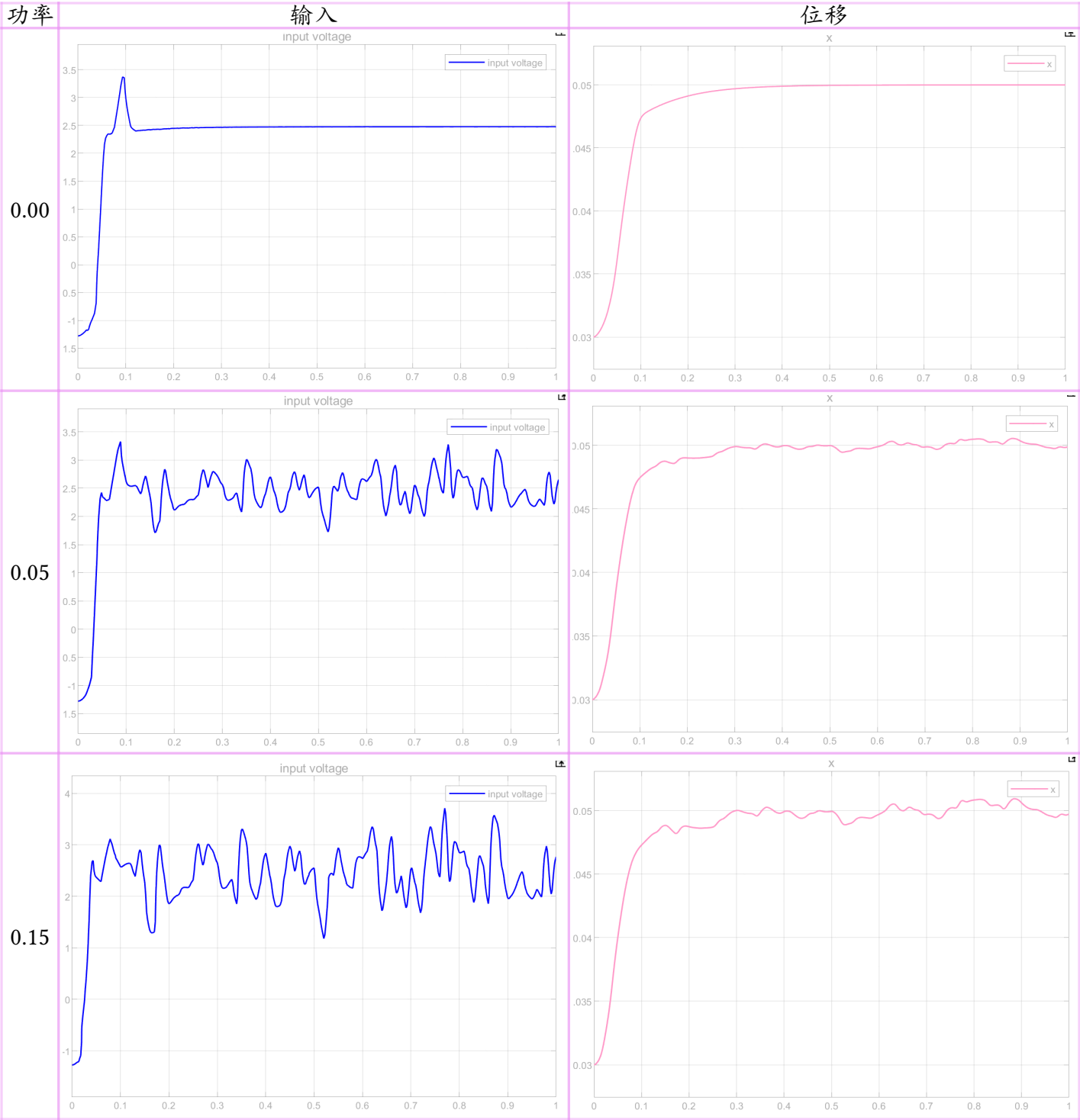


Table 3: 不同噪音功率下的控制效果对比

以下，外界在 0.8 秒时，对于小球速度施加阶跃扰动（正向速度 0.5，负向速度 0.3），模拟锤子敲击小球或意外撞击。当负向施加的速度为 0.4 以上时，就失控了，即：负向比正向的阶跃扰动更易导致不稳定（下图也可见，负向的阶跃使得输入值变化要更大），这也符合物理图像，因为（平方）反比的关系，靠近 0 处相较于远离 0 处，有更大的变化。

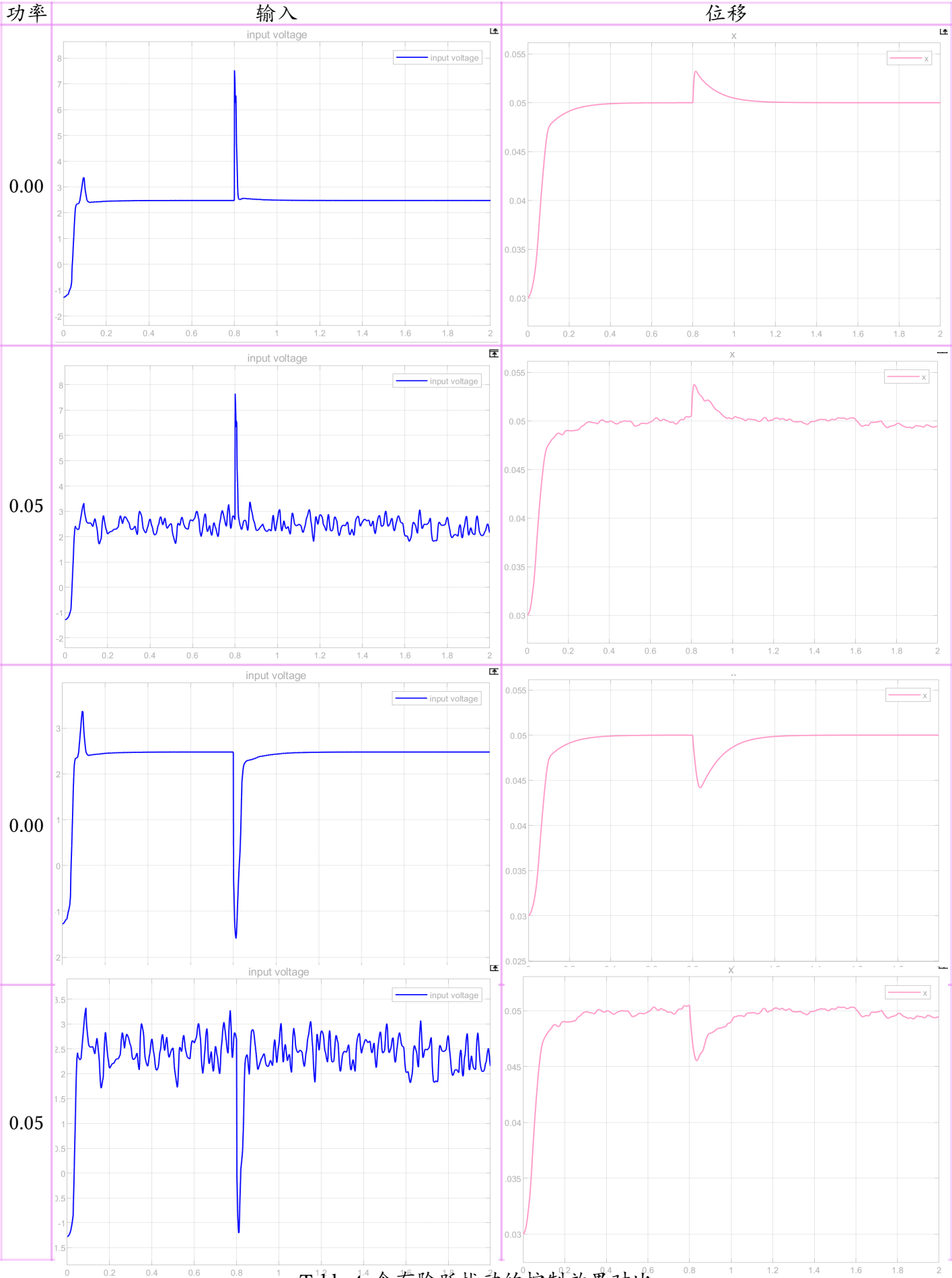
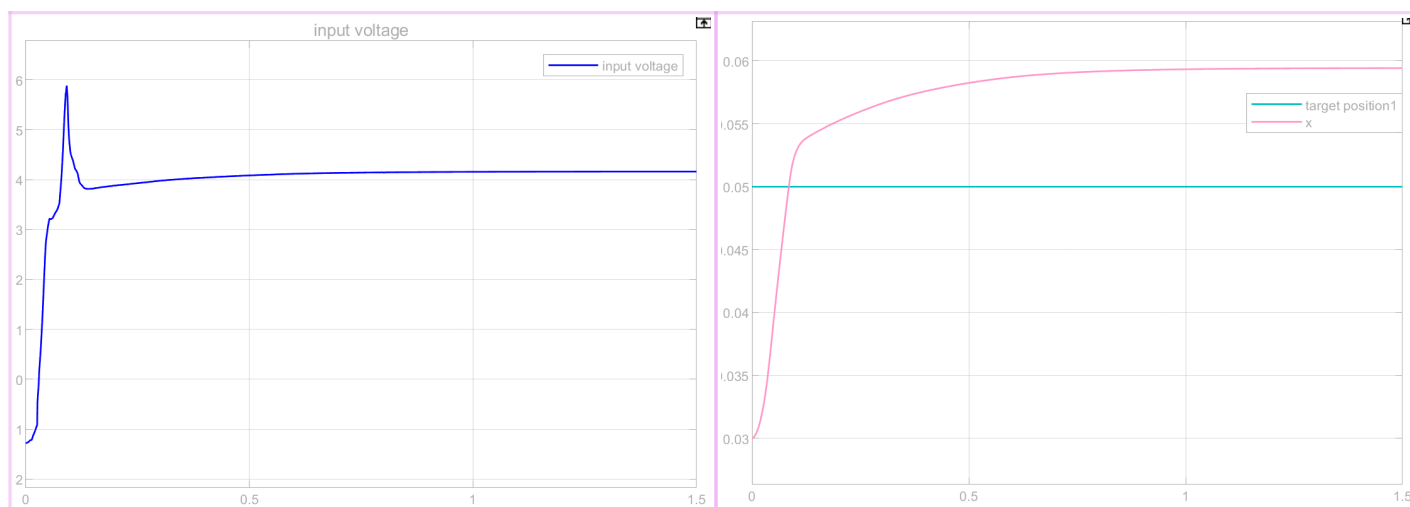


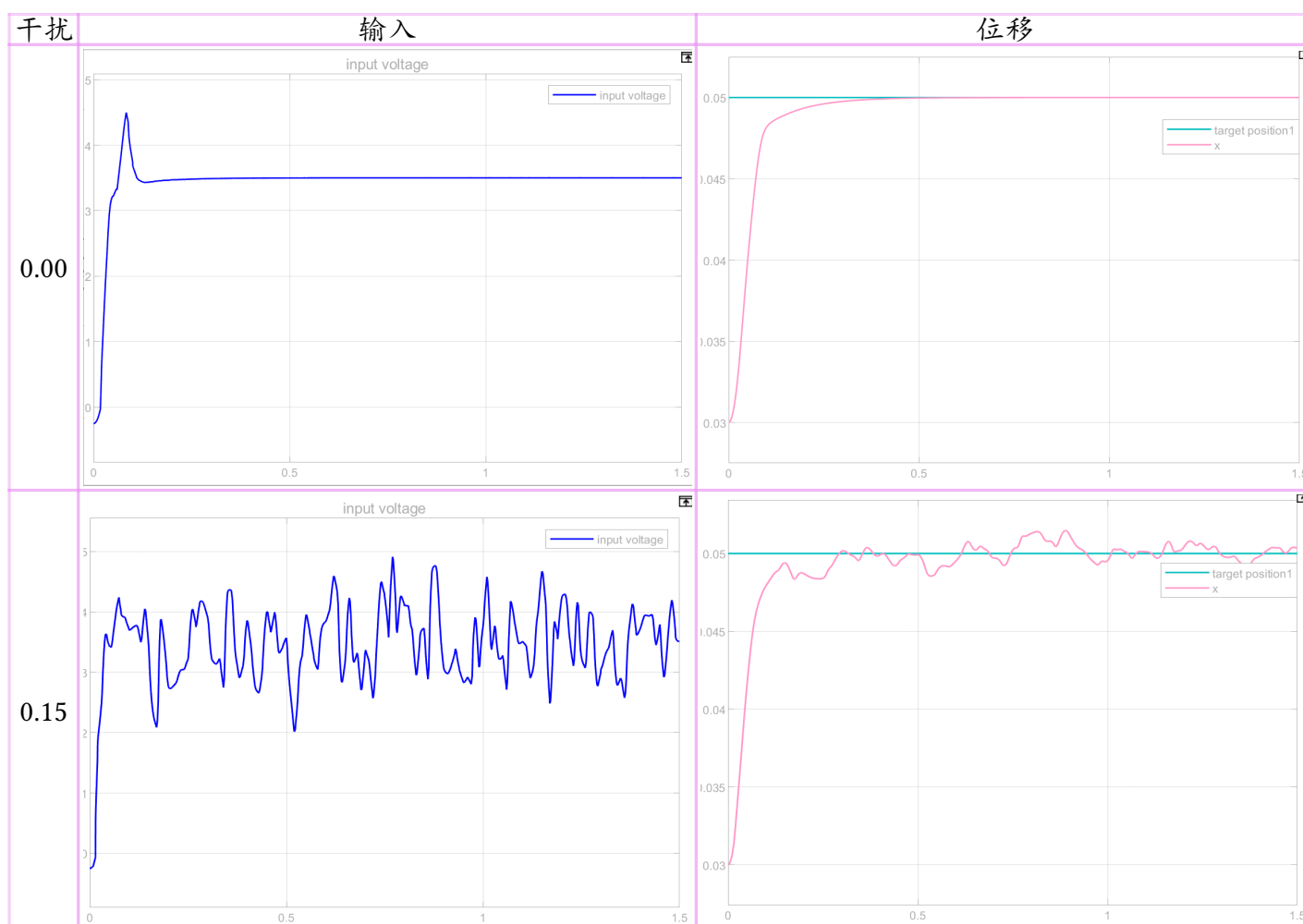
Table 4: 含有阶跃扰动的控制效果对比

## 参数变化后的控制

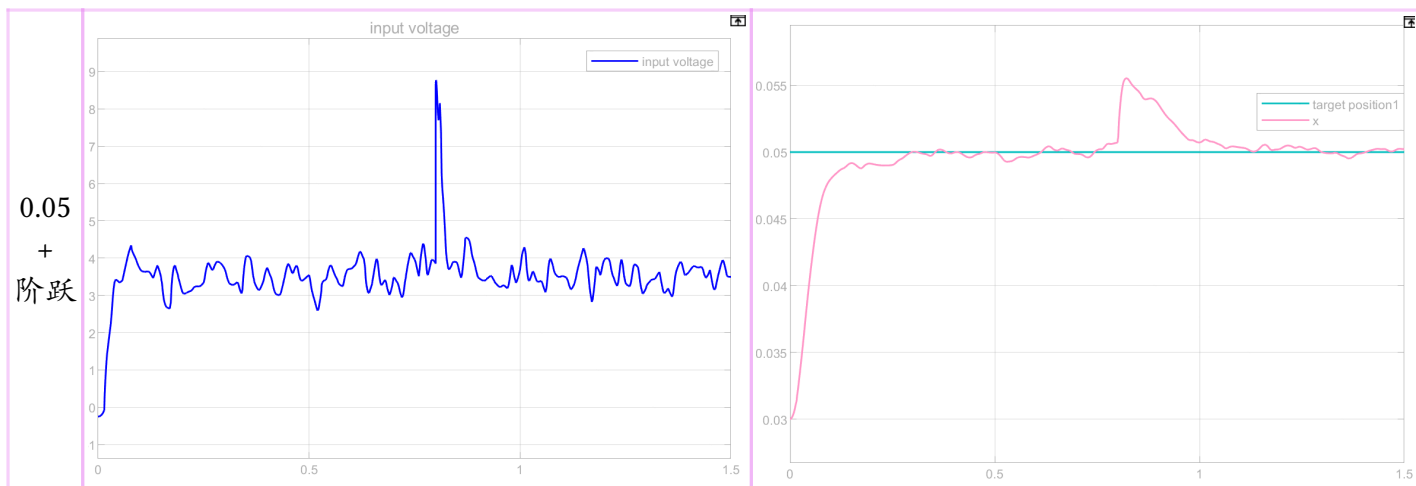
钢球质量的变化为 0.1kg，假如不改变控制策略（即增加的稳态偏置仍然不变），则将出现明显的余差如下：这当然符合直觉，因为球重了之后，平衡位置会偏下， $x$ 也就稳定在更大的值了。



当把稳态工作点的输入电压 $u_0$ 重新计算得出后，再施加上去，就能得到较好的控制效果：







下面分析模糊控制器的调整：

**注：**以上都是加入了额外的稳态偏置的方法，而非调整控制器的参数的方法。但实际上这两者很大程度上等效！原因是，修改模糊控制器输出的论域，将整个论域向右平移 $u_0$ 个长度，就完全等同于模糊控制器论域不变，而额外给控制器输出叠加一个 $u_0$ 的值，该求和作为实际的操作变量传入系统。

为验证这一点，参考附件中的 levitate\_controller2.fis 这一文件，将 simulink 仿真文件中的那个模糊控制器改成“导入 levitate\_controller2.fis”，并且把偏置 $u_0$ 注释掉，得到的效果完全等同于“导入 levitate\_controller.fis”，并且把 $u_0$ 加上。

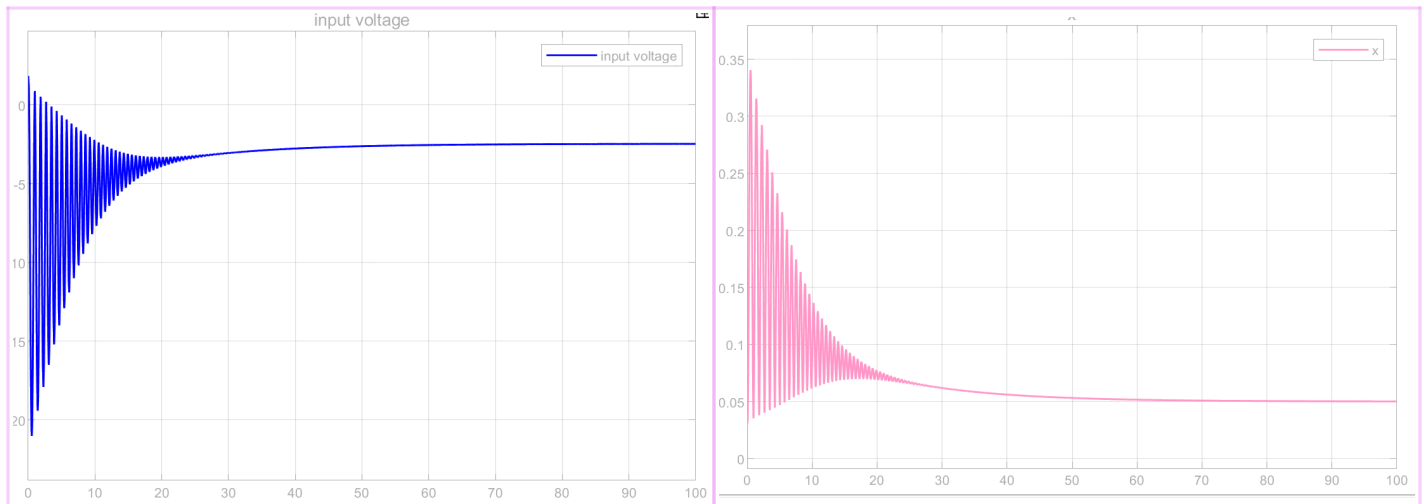
因此，在小球质量增加时，模糊控制器只需要调整输出的论域，将整体向右平移即可。

## 模糊 PID 尝试失败

由于用普通模糊控制，为了消除余差，需要先验的稳态工作点的知识，但这就需要对于对象本身有精确的理解与建模。所以，我企图用积分器试凑得到稳态的 $u_0$ 值，于是尝试用如下的模糊PID控制规则，其中P、I、D三个系数是模糊控制器的输出，三个参数分别划分为三挡：S、M、L（小中大，三角形函数），但是没有调参成功，估计是论域选择有问题，没试出合适的参数。

MV	$\Delta x$																					
		NNN			NN			N			ZERO			P			PP			PPP		
		P	I	D	P	I	D	P	I	D	P	I	D	P	I	D	P	I	D	P	I	D
$\dot{x}$	NFAST	L	S	L	L	S	L	M	S	L	S	S	L	S	S	L	S	S	M	M	S	S
	NMED	L	S	L	L	S	L	M	S	L	S	S	L	S	S	M	M	S	S	M	S	S
	NSLOW	L	S	L	L	S	L	M	M	M	S	M	M	S	M	S	M	S	S	L	S	S
	STILL	L	S	S	L	M	S	M	L	S	S	L	S	M	L	S	L	M	S	L	S	S
	PSLOW	L	S	S	M	S	S	S	M	S	S	S	M	M	M	M	L	S	L	L	S	L
	PMED	M	S	S	M	S	S	S	S	M	S	S	L	M	S	L	L	S	L	L	S	L
	PFAST	M	S	S	S	S	M	S	S	L	S	S	L	M	S	L	L	S	L	L	S	L

分析原因，该系统的非线性性比较强，积分器的作用是试凑出稳态工作点，可能在这个试凑过程中已经发散了。如果比例作用弱，一开始就飞了，若比例作用大，则太振荡，如下（下图是普通PID的结果）：



这种振荡显然不合适，MV 的范围也超出可接受的上下限。因此，对于本例中对象特性一无所知的情况下，找稳态工作点的输入还是较困难的。

## 代码附件与运行说明

> 使用的 Matlab 版本为 2023b

附件含有：

- > levitate\_model.slx 仿真模型
- > levitate\_param.m 含有参数
- > levitate\_controller.fis 模糊控制器
- > levitate\_controller2.fis (输出论域偏移的模糊控制器)
- > levitate\_controller3.fis (模糊 PID 尝试，但参数调不好)

测试时，请先运行 levitate\_param.m 文件（使得 workspace 中有参数的数据）然后即可运行 slx 文件。slx 文件中含有两组，一组是普通 PID（对照组，说明无稳态工作点较难调），一组是模糊控制。

如有疑问，请访问 [https://github.com/Maythics/Control\\_simulink.git](https://github.com/Maythics/Control_simulink.git)