中 国 地 质 大 学

课程结课报告

课程名称:	过程控制原理与应用技术 I：过程控制原理与仪表
题目名称:	用于阀门控制的强化学习
学　　号:	20211003337
学生姓名:	曾康慧
专业班级:	220211
任课教师:	安剑奇
完成时间:	2024.1.15

目录

1. 引言

强化学习（RL）是一种模仿人类和动物学习能力的机器学习技术。OpenAI 已使用 RL 应用程序对机器人手进行编程，以前所未有的人类灵巧性操纵物理对象，斯坦福大学的 CARMA 自动驾驶计划，并研究了更快的从头分子设计。
本文研究了 RL 作为最优控制策略的应用。RL 通过直接与被控对象交互并学习最佳控制，而无需对被控对象进行精确建模，从而实现更好的控制。
阀门被选为受控工厂，因为它们在过程控制中无处不在，并且几乎用于所有可以想象的制造和生产行业。工业过程回路可能涉及数千个阀门，并且可能无法准确建模。

PID（比例-积分-微分）是覆盖 95%以上工业控制器的事实上的控制策略。然而，应用这种传统策略可能会影响流程的质量和效率，并增加大量成本。

将计算机连接到真实的物理工厂，并让强化学习代理通过直接互动学习，可能并不总是可行的。一个通常采用的实际方法是尽可能地模拟真实工厂，这也是我们采用的方法。我们使用 MATLAB Simulink®来模拟非线性阀门、工业过程、代理训练电路，最后是统一的强化学习-PID 验证电路。在 RL 术语中称为“代理”的控制器，使用 MATLAB 最近推出的（R2019a）Reinforcement Learning Toolbox™和 DDPG（Deep Deterministic Policy-Gradient）算法进行训练。。

将计算机连接到真实的物理工厂，并让 RL 代理通过直接交互进行学习可能并不总是可行的，并且通常采用的实用方法包括尽可能接近真实工厂，这是我们使用的方法。MATLAB Simulink® 用于仿真非线性阀门、工业过程、智能体训练电路，最后是统一的 RL-PID 验证电路。该控制器在 RL 术语中称为“代理”，使用 MATLAB 最近推出的强化学习工具箱进行训练，该工具箱使用 DDPG（深度确定性策略梯度）算法。

最后，虽然阀门是本文的重点，但这些方法适用于任何工业系统。

2. 强化学习入门

在本节中，我们将简要介绍传统的最优控制求解方法，然后概述 RL 及其与最优控制的联系，最后选择用于实现的 DDPG 算法。

2.1. 最优控制和 RL

反馈控制器传统上采用两种设计理念：自适应控制和最优控制。自适应控制器是在线学习者，通过测量实时数据来学习控制未知系统。然而，它们没有被优化，因为设计过程不涉及最小化工厂的任何性能指标。
另一方面，传统的最优控制设计是通过求解汉密尔顿-雅可比-贝尔曼 （HJB）方程离线执行的。

2.2. 最优控制

汉密尔顿-雅各比-贝尔曼 （HJB） 为最优性提供了充分条件。

$$0 = min_u \left[g(x,u) + \frac{\partial J^*}{\partial x} f(x,u) + \frac{\partial J^*}{\partial t} \right] \quad (1)$$

控制器*策略*（即行为）表示为π和最佳策略π_*。策略$\pi(x,t)$以及相关的成本函数$J^{\pi}(x,t)$，定义如下：

$$J^{\pi}(x,t) = J^*(x,t), \quad \pi(x,t) = \pi^*(x,t) \quad (2)$$

式（1）假设成本函数在x和t由于情况并非总是如此，因此它不能满足所有最优控制问题。在 Tedrake（2009）中，Tedrake 表明求解 HJB 取决于工程猜测，例如，一阶调节器是用猜测解设计的$\pi(x,t) = -sgn(x)$.线性二次稳压器的设计与此类似。对于复杂的动态机械系统，除非非常近似，否则很难猜测这种初始解，因此在这样的情况下，RL 显示了可以相对容易*地学习*现实世界的最优控制。

2.3. RL 框架

RL 的核心元件如图 1 所示，与等效的控制系统元件叠加。

学习者和决策者称为*代理*。代理不断与其*环境*交互，选择操作一个A_t，环境通过呈现新情况来做出反应S_{t+1}.环境通过*奖励*（或惩罚）提供绩效反馈，R_{t+1}.奖励是标量值。随着时间的流逝，智能体试图最大化（或最小化）奖励，这*强化*了好的行为而不是坏的行为，使其能够学习最佳*策略*。

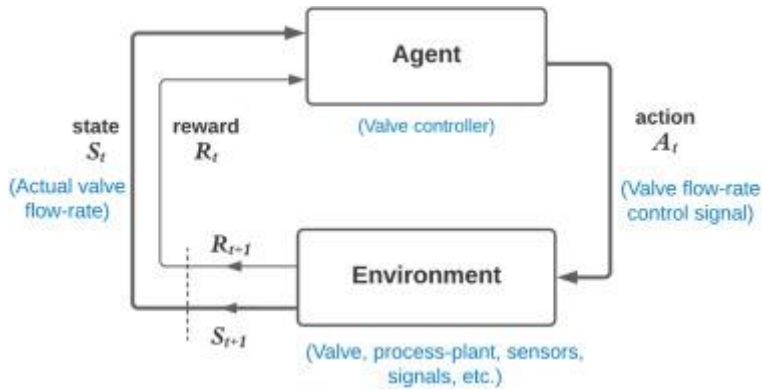


图 1.RL 和等效控制系统元件的构建块。

在控制系统术语中，智能体是正在设计的控制器，环境由控制器外部的系统组成，即阀门、工业过程、参考信号、其他传感器等。策略是设计人员寻求的最

优控制行为。RL 允许学习这种行为，而不必进行显式编程或对被控对象进行极其详细的建模。

策略：智能体的决策能力基于相对于其所处状态的最佳行动的概率映射。正是这种映射构成了策略 π_t 和 $\pi_t(a|s)$ 是动作的概率 $A_t = a$，如果状态 $S_t = s$.

回报：回报代表长期奖励，随着时间的推移而积累。

$$G_t = R_{(t+1)} + R_{(t+2)} + R_{(t+3)} \dots R_T \tag{3}$$

折扣：折扣提供了一种机制，用于控制选择立即操作与在遥远的未来获得奖励的操作的影响。

$$G_t = R_{(t+1)} + \gamma R_{(t+2)} + \gamma^2 R_{(t+3)} \cdots = \sum_{k=0}^{\infty} \gamma^k R_{((t+1)+k)},\ 0 \le \gamma \le 1 \tag{4}$$

值函数：作为状态-动作对的函数，它们提供了在给定状态下执行给定动作的好坏的估计。奖励信号在短期内提供有关当前行动的“好”程度的反馈。相比之下，价值函数提供了长期“好”的衡量标准，并根据未来预期回报来定义。

值，表示 $v_\pi(s)$，是状态的预期回报 s，从该状态开始测量 s 并遵循政策 π 此后。

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{(t+k+1)} \mid S_t = s\right] = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right] \tag{5}$$

方程（5）称为**贝尔曼方程**，是近似计算和学习的基础 v_π 并且是所有 RL 算法的核心。

Q 函数：通过包含动作，$q_\pi(s,a)$ 定义为从状态开始的预期回报 s，执行操作 a 此后遵循政策 π.

Q-learning： Q-learning 是一种偏离策略的 TD 控制算法，允许迭代学习 Q 值。对于每个状态-动作对;价值 $Q(s,a)$ 被跟踪。当 a 操作在 s 状态下执行，来自环境反馈的两个要素——奖励 R 和下一个状态 S_{t+1} 在（6）所示的更新中使用。α 是学习率。

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma . \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right], \tag{6}$$

最优价值函数：始终存在至少一个最优策略，以保证最高预期回报，表示为 v_* 以及最优的动作-价值-函数 q_*.

基于模型和无模型的强化学习方法：准确的环境模型允许“规划”下一步行动和奖励。环境的模型意味着可以访问给定操作时处于状态的概率“表”;以及相关的奖励。

使用环境模型的 RL 方法称为*基于*模型的方法，而不是更简单的*无模型*方法。无模型的智能体只能通过反复试验来学习。

Actor-Critic 方法：Actor-critic 结构允许实时实现的 RL 算法的正向类。在策略下，*actor* 组件将操作应用于环境，并接收由评论家评估的反馈。学习是一个两步走的机制——由批评者进行政策评估，然后由行动者进行政策改进（见图 2）。

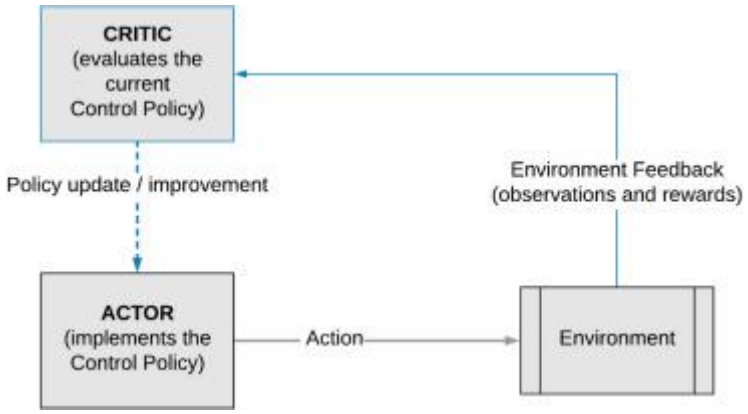


图 2. Actor-Critic 架构。

2.4. DDPG 算法

MATLAB 的 R2019a 版本提供了六种 RL 算法。DDPG 是唯一适用于连续动作控制的算法。

DDPG 克服了 DQN（深度 Q 网络）算法的缺点，而 DQN 算法又是基本 Q 学习算法的扩展。

DDPG 是一种无模型、策略梯度和策略外（通过使用以前经验的内存重放缓冲区）。作为一种 Actor-critic 方法，它使用了两个神经网络。参与者网络接受当前状态作为输入，并输出一个单一的实值（即阀门控制信号），表示从连续动作空间中选择的动作。critic 网络通过估计给定该动作的当前状态的 Q 值来对参与者的输出（即动作）进行评估。参与者网络权重由*确定性*策略梯度算法更新，而 critic 权重则由从 TD 误差信号获得的梯度更新。因此，DDPG 算法通过交错来同时学习 Q 函数和策略。

探索与利用：对于强化学习来说，就像在人类中一样，性能改进是通过利用过去最好的行动来实现的。但是，要发现这些操作，代理必须首先探索未尝试的操作。在不断改进最佳行动的同时平衡这一发现是一个共同的挑战。已经制定了各种勘探-开发策略。

DDPG 使用 Ornstein-Uhlenbeck 流程 （OUP） 进行勘探。有趣的是，OUP 是为模拟布朗粒子速度而开发的，具有摩擦力，并产生*时间相关的*值。更简单的加性高斯噪声模型会导致从一个时间步长到另一个时间步长的突然不相关变化。OUP 更接近于模仿表现出惯性的现实生活中的致动器。

勘探政策 π' 是通过向所选动作添加噪声来构造的，该操作是从 OUP 噪声过程中采样的 $\mathcal{N}$.

$$\pi'(s_t) = \pi(s_t \mid \theta^\pi) + \mathcal{N}_t \tag{7}$$

3. 控制阀和 RL

控制阀使用由控制系统控制的执行机构来调节流体流量。加工厂由大型控制阀网络组成，旨在控制过程变量，以确保最终产品的质量。

3.1. 阀门中的非线性

与大多数其他物理系统一样，控制阀具有非线性流动特性，例如摩擦和间隙。反过来，摩擦力有两个组成部分 - *静*摩擦力，静摩擦力，是在两个表面之间发生任何相对运动之前必须克服的惯性力，是阀门死区的主要原因;而动态摩擦是运动中的摩擦（图 4）。
非线性会导致振荡阀输出（图 3），进而导致过程输出的振荡，从而导致最终产品有缺陷、能源消耗低效和制造系统过度磨损。30% 的过程回路振荡问题是由控制阀引起的，阀门是 32% 的受访控制器效率低下的主要原因。控制阀中的阻滞是工业控制回路中持续振荡的主要来源。

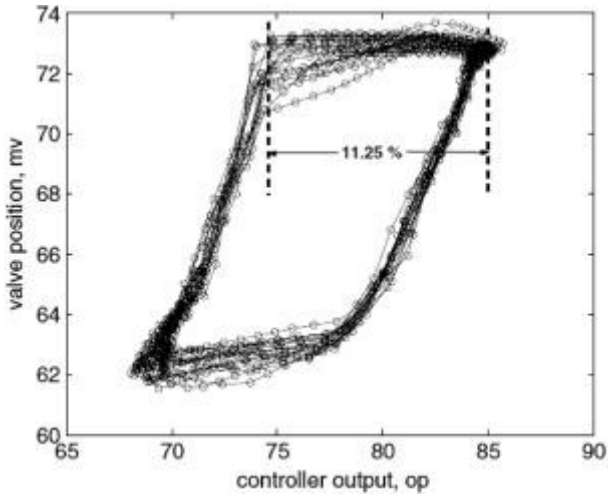


图 3.实际阀门运动轨迹

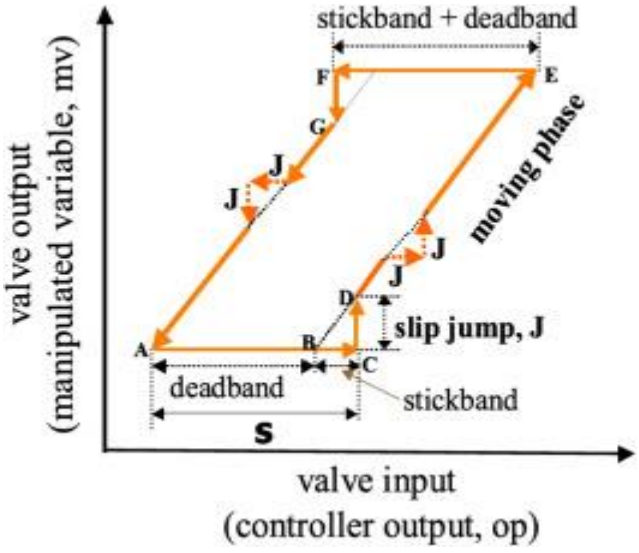


图 4.非线性阀门操作特性，具有静力

3.2. 数学阀门模型

RL 需要经验进行培训。模拟环境通常为培训代理提供快速且低成本的环境。由于构建控制器的目的是使其在现实世界中使用，因此必须努力创建尽可能准确的环境。这似乎与之前提出的 RL 不需要精确的系统模型的说法相矛盾，但是这里假设真实的物理环境是无法访问的，另一方面，如果可访问或可用，则可以允许 RL 代理（控制器）直接从真实经验中学习。

在本文中，我们使用第一性原理对阀门进行建模，

通过以下方式描述了阀门的非线性记忆动力学 $x_k = N_v\left(x_{k-1}, u_k\right)$，以$k$为时间步长，当$N_v$由关系（8）表示。当控制器输出$u$，阀门达到的实际位置表示为$x$，$e_k$表示阀门位置误差。$f_s$和$f_k$是静态（静力）和动态摩擦参数，取决于阀门类型、尺寸和应用。“实验设置”部分介绍了阀门的 Simulink 建模。

$$x_k = \begin{cases} x_{k-1} + \left[e_k - sign\left(e_k\right) f_D\right], & \text{if} \quad |e_k| > f_S \\ x_{k-1}, & \text{if} \quad |e_k| \le f_S \end{cases}, \quad e_k = u_k - x_{k-1} \tag{8}$$

4. 实验装置

本节介绍如何使用 MATLAB 和 Simulink 创建实验设置，用于设计和评估 RL 和 PID 控制器。图 9 显示了核心组件。

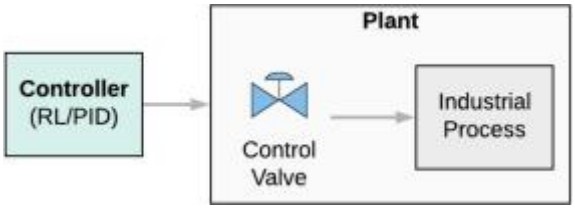


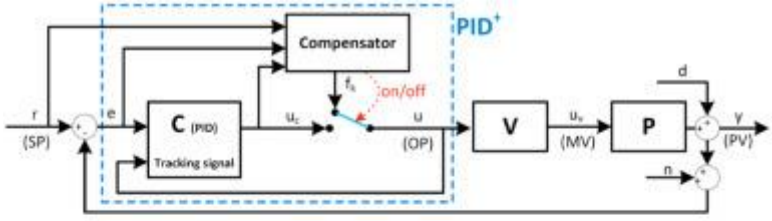
图 9.基本块组件。

传统的 PID 控制器仅根据过程动力学进行调整，导致由于整体组件导致持续振荡，导致控制动作的过度变化以克服静摩擦。作为解决方案，基于 PID 的新型控制器，如图 10（a）所示，其中通过采用两步控制序列作为阀门输入来克服阻滞。

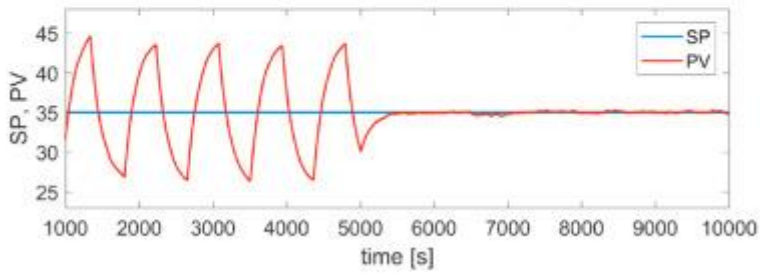
$$u_{k+1} = \begin{cases} \hat{x}_{ss} - \hat{f}_D, & u_{k-1} \ge \hat{x}_{ss} \\ \hat{x}_{ss} + \hat{f}_D, & u_{k-1} < \hat{x}_{ss} \end{cases} \tag{11}$$

$\hat{f}_S$ 和 $\hat{f}_D$ 是静力和动态摩擦的估计值，以及 $\hat{x}_{ss}$ 是阀门稳态位置的估计值。方程（11）高度依赖于摩擦参数的精确估计。

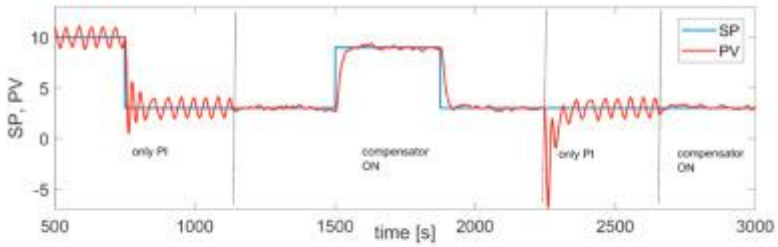
设置组件：



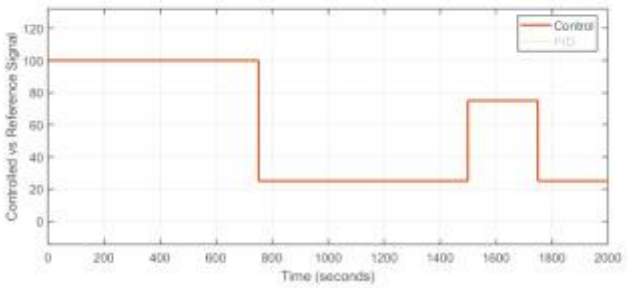
(a) Two-move compensator



(b) Compensator results on a constant reference signal



(c) Compensator results on a process with loop perturbations



(d) Regenerated "benchmark waveform"

图 10."PID 补偿器"

- 1.
  使用 MATLAB 的自动调谐功能调谐的 PID（带滤波器）控制器。
- 2.
  使用 DDPG 算法的 RL 代理的训练设置。
- 3.
  控制器实验和评估的统一框架 4.

  非线性阀门模型，包括阀门摩擦值 f_S 和 f_D .

- 5.
  阀门控制的两个工业过程：
  - （一）
    正常过程
  - （二）
    具有环路扰动的过程
- 6.
  带有噪声参数的“基准波形”曲线

4.1. 阀门建模

使用第一性原理，对非线性阀进行数学建模。（11）所示的代数重排方程得到（12）;然后，在 Simulink 中使用“用户定义函数”和图 11 所示的“memory”模块中的代码实现这些方程，其中 f_S =8.40 和 f_D =3.524.

$$x_k = \begin{cases} u_k - f_D, & \text{if} \quad u_k - x_{k-1} > f_S \\ u_k + f_D, & \text{if} \quad u_k - x_{k-1} < -f_S \\ x_{k-1}, & \text{if} \quad |u_k - x_{k-1}| \le f_S \end{cases} \tag{12}$$

Listing 1: MATLAB script for modelling a non-linear valve

```
function xk = fcn(fD, fS, uk, xkp)
    t_xk = 0.0;
    if ((uk-xkp) > fS)
        t_xk = uk - fD;
    elseif ((uk-xkp) < -1*fS)
        t_xk = uk + fD;
    elseif (abs(uk-xkp) < fS)
        t_xk = xkp;
    end

    xk = t_xk;
```

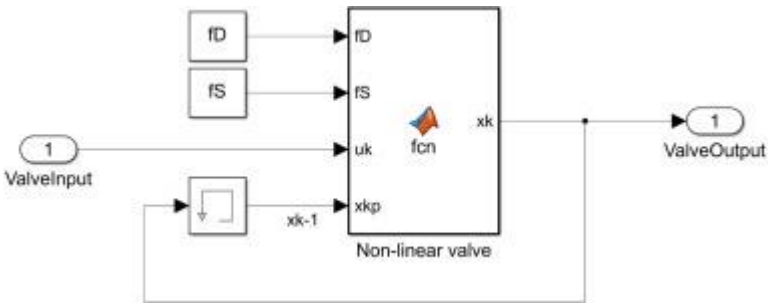


图 11.Simulink 阀模型。

4.2. 对“工业”过程进行建模

基准“工业过程”被建模为一阶加延时（FOPTD）过程（13），使用传递函数和延时模块，如图 12 所示。

$$G(s)=\frac{k}{(1+Ts)}e^{-Ls}, \qquad k=3.8163, T=156.46, L=2.5. \tag{13}$$

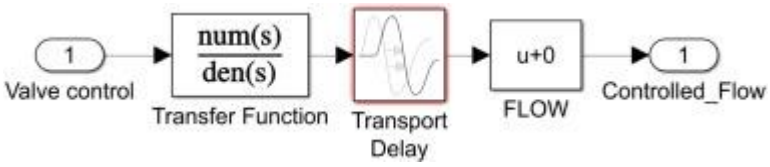


图 12.FOPTD 过程模型。

4.3. PID 控制器设置

PID 控制输出是反馈误差的函数，在时域中表示为：

$$u(t)=K_p e+K_i\int e(t).dt+K_d\frac{de}{dt} \tag{14}$$

u是所需的控制信号，并且$e(t)=r(t)-y(t)$是所需输出之间的跟踪误差r和实际输出y.该误差信号被馈送到 PID 控制器，控制器计算该误差信号的导数和积分与时间的关系，提供设定点跟踪效果，这在闭环中连续工作，直到控制器生效（见图 13）。

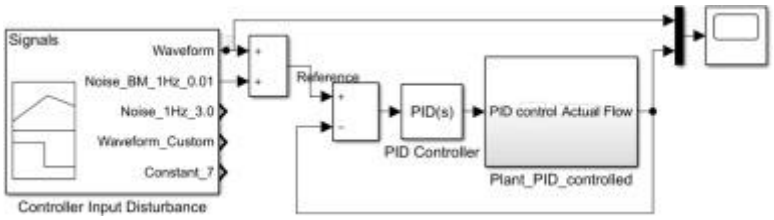


图 13.PID 控制设置。

对于高频信号，理想的理论 PID 形式存在一个缺点，即微分作用导致非常高的增益。因此，高频测量噪声会在控制信号中产生较大的变化。实际实现通过替换K_d一阶滤波器的项（其中$K_d.de/\,dt$表示为$K_d.s$以拉普拉斯形式）。

$$K_p+K_i\frac{1}{s}+K_d\frac{N}{\left(1+N\frac{1}{s}\right)} \tag{15}$$

滤波系数N确定滤波器的极点位置，以帮助衰减高频噪声的高增益。一个N之间 2 和 20 是推荐的。

使用 MATLAB 自动整定功能对 PID 进行整定，得到的系数为$K_p=0.3631$, K_i =0.0045,K_d=−1.72 和N=0.0114.低N抑制导数项的作用。

4.4. RL 控制器设置

图 14 显示了 Simulink 设置，用于训练和评估 RL 控制器。
训练智能体涉及重要的超参数调整和开关，允许对通过“信号生成器”模块馈送的大量信号进行快速实验。

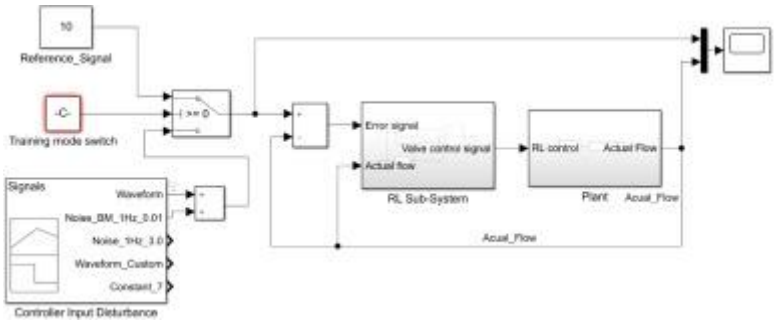


图 14.RL 代理培训设置。

4.4.1. RL 控制器设计

图 15 显示了 DDPG Agent 模块，其中包含通过观察向量通道化的环境反馈。它还显示了计算奖励的模块和控制剧集终止的 Stop-simulation 模块。

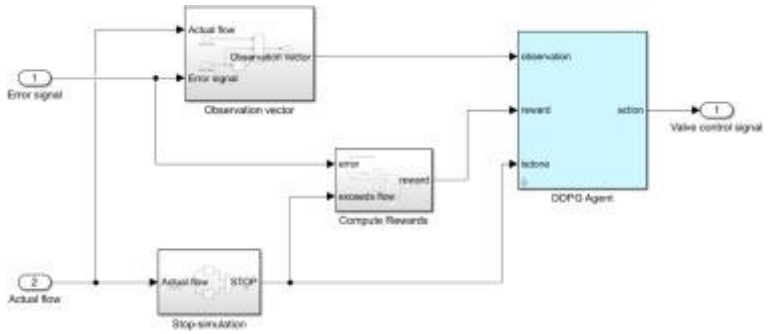


Fig. 15. RL DDPG agent details.

4. 4. 2. 环境设计

在构建环境时，需要考虑几个设计因素，以便有效地训练智能体跟踪控制信号的轨迹。它们大致可分为与代理相关和环境相关。

与智能体相关的因素是观察向量和奖励策略的组成。与环境相关的因素包括训练策略、训练信号、环境的初始条件和终止发作的标准。

4. 4. 3. 训练策略

人们可以训练 RL 智能体遵循精确的基准轨迹［图 10（d）］，但这是一个非常受限的策略。取而代之的是，智能体被训练为遵循随机水平的直线（恒定）信号。此外，智能体还面临着学习从随机初始化的流值开始的挑战。这共同形成了一种有效且通用的训练策略，以教导智能体遵循*由*直线组成的任何控制信号轨迹。RL 工具箱允许覆盖默认的“重置功能”，以帮助实现上述策略。

```
env.ResetFcn = @(in)localResetFcn(in,
        VALVE_SIMULATION_MODEL);
```

4.4.4. 观测向量

如图 17 所示建模的观测向量由以下部分组成：$\left[y;e;\int e.dt\right]^T$，$y$是实际实现的流量，e 关于参考的错误 r，最后是误差的积分。

误差积分：瞬时误差没有记忆。误差积分是随时间推移的曲线下面积，它提供了一种机制来计算随时间收集的总误差，并驱动智能体降低该误差（图 16）。这是 RL 控制器训练中经常使用的重要观察输入。

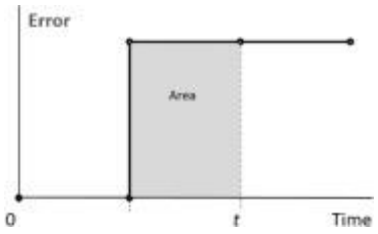


图 16. 误差积分。

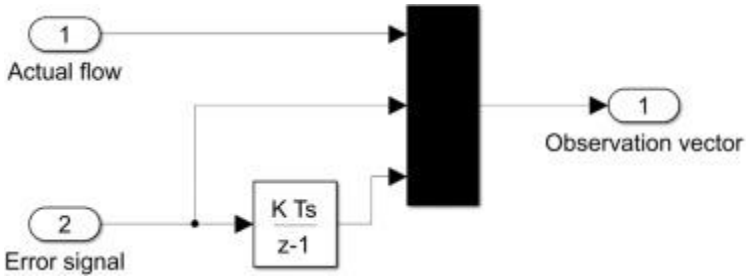


图 17. RL 观测向量。

4.4.5. 奖励策略

奖励可以通过离散、连续或混合功能进行分配。 方程（16）是简单的离散形式。

$$Reward=\begin{cases}10, & \text{if} \quad |e|<\Delta \\ -1, & \text{if} \quad |e|\geq\Delta \\ -100, & \text{if} \quad (y\leq 0, y>Max\_Flow),\end{cases} \qquad \Delta\text{是允许的误差范围} \qquad (16)$$

式（17）显示了作为误差函数而连续变化的奖励$e.\lambda$是一个小常数，可避免除以零误差。

$$Reward=\begin{cases}-100, & \text{if} \quad (y\leq 0, y>Max\_Flow) \\ \dfrac{1}{(e+\lambda)} & \text{otherwise}\end{cases} \qquad (17)$$

图 18 显示了作为混合形式的最终实现。绝对误差的倒数使控制器能够学会将误差驱动得越来越低。奖励的离散部分是“惩罚”块，它为超过流量限制分配设定的惩罚。

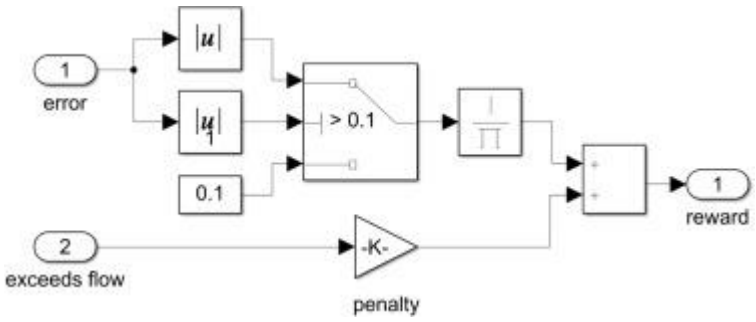
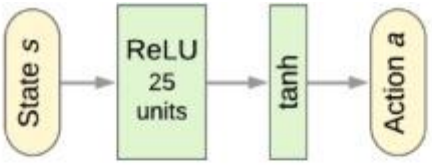


图 18.RL 奖励计算块。

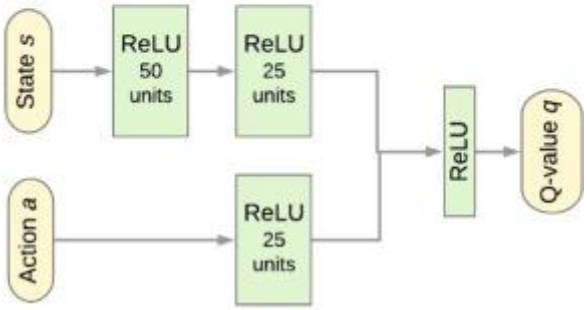
4.4.6. The actor–critic 网络

The actor–critic DDPG 组件的实现如图 19 所示。网络具有完全连接的层，在开始训练之前使用小的随机权重进行初始化。

使用 **tanh** 层将 actor 网络输出归一化为介于 `[-1, 1]` 之间。这允许更好的学习和收敛，以实现连续行动空间。



(a) Policy (actor) network



(b) Critic (action-value) network

图 19.DDPG 网络架构。

4.4.7. Ornstein-Uhlenbeck （OU）动作噪声参数

$$Variance.\sqrt{T_s} = (1\% \text{ to } 10\%) \text{ of } ActionRange \quad T\text{是采样时间} \tag{18}$$

在确定方差因子的半衰期时，以时间步长计算衰减率，然后使用（19）计算衰减率

$$HalfLife = \frac{\log\left(\frac{1}{2}\right)}{\log(1 - VarianceDecayRate)} \tag{19}$$

4.4.8. 最终的 DDPG 超参数

表 1 总结了最后一组 DDPG 超参数。

表 1.DDPG 超参数设置。

超参数	**设置**
评估学习率	$1e^{-03}$
改进学习率	$1e^{-04}$
评估隐藏层-1	50 个全连接
改进隐藏层-2	25 全连接
改进路径神经元	25 全连接
操作路径绑定	Tanh 层
λ	0.9
批量大小	64
OUP 方差	1.5
OUP 方差衰减率	$1e^{-05}$

4.5. 比较研究的设置

图 20 显示了结合 PID 和 RL 策略进行比较评估的环境。它允许对各种参考信号进行实验，研究在三个干扰点（即控制器的输入、控制器的输出（即被控对象的输入）和最后的被控对象输出）处添加的噪声的影响。
它提供了一个方便的平台，可以使用设定点滤波器、输出平滑滤波器等元素进行额外的实验。

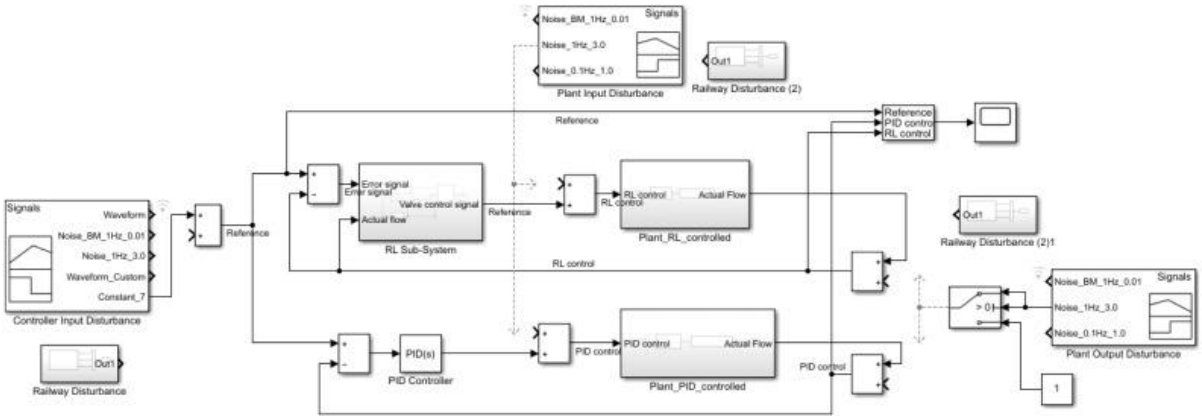
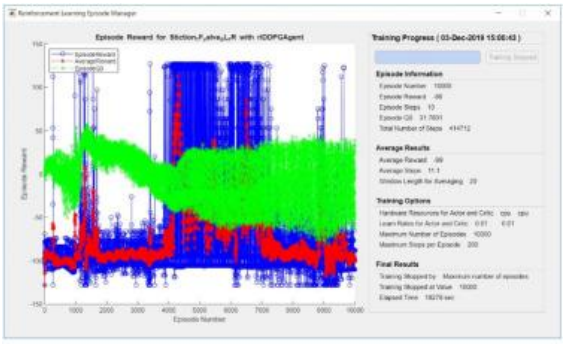


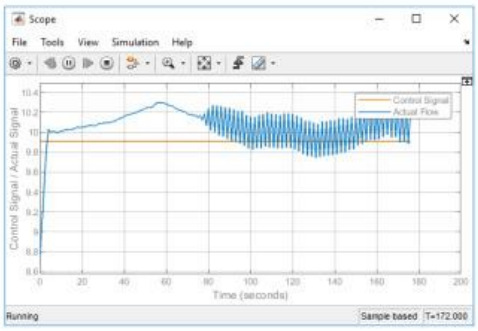
图 20.用于比较 RL 和 PID 控制策略的统一设置。

5. 分级学习

“分级学习”是一种渐进*式辅导方法*。
在 MATLAB 中，应用自动化课程学习并不容易获得;另一方面，分级学习不需要编程，可以由控制工程师轻松实现。
图 21 显示了训练过程中面临的众多挑战的示例，有时会导致数千集的实验没有产生稳定的学习曲线，有时会导致莫名其妙的控制器操作。一些训练试验持续了 20,000 次发作，持续了 20 多小时，因此简化这些工作非常重要。
分级学习有助于避免其中一些挑战。分级学习的直觉是基于观察人类教师如何为学徒组织新技能*的指导*。

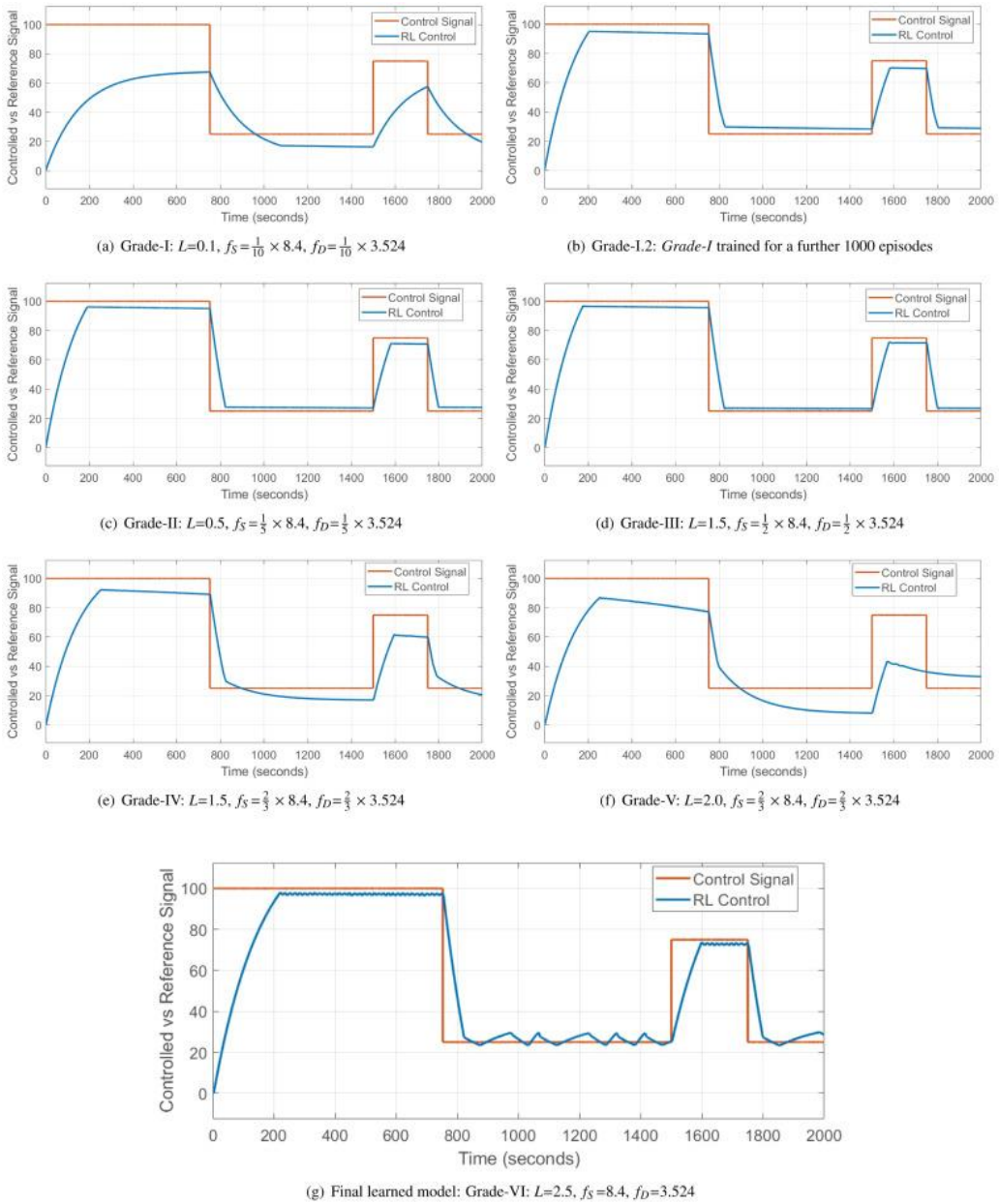


(a) Inexplicable learning curves



(b) Inexplicable controller actions

图 21.RL 代理：我们试验



(a) Grade-I: L=0.1, $f_S=\frac{1}{10}\times 8.4$, $f_D=\frac{1}{10}\times 3.524$

(b) Grade-I.2: *Grade-I* trained for a further 1000 episodes

(c) Grade-II: L=0.5, $f_S=\frac{1}{5}\times 8.4$, $f_D=\frac{1}{5}\times 3.524$

(d) Grade-III: L=1.5, $f_S=\frac{1}{2}\times 8.4$, $f_D=\frac{1}{2}\times 3.524$

(e) Grade-IV: L=1.5, $f_S=\frac{2}{3}\times 8.4$, $f_D=\frac{2}{3}\times 3.524$

(f) Grade-V: L=2.0, $f_S=\frac{2}{3}\times 8.4$, $f_D=\frac{2}{3}\times 3.524$

(g) Final learned model: Grade-VI: L=2.5, f_S=8.4, f_D=3.524

图 22.分级学习：任务的渐进式学习。

分级学习将迭代分阶段方法扩展到强化学习。RL 任务首先被分解到其基本级别，一个智能体被训练n发作或直到满足收敛标准。在上一个任务的基础上增加了下一个级别的复杂性。*迁移学习*用于确保保留和建立以前的经验。一旦学习了这个级别的任务，增加复杂性的过程就会继续，每次迁移学习都允许在前几个级别获得的经验的基础上再接再厉。

迁移学习是一种机器学习技术，用于将学习（即神经网络的稳定权重）从一个任务（或一般领域）“转移”到另一个任务，而无需从头开始训练神经网络。

图 22 演示了该方法的实际应用和代理在*难度递增*的六个阶段中演变。逐渐增加的参数包括：延时L、静摩擦 f_S 和动态摩擦 f_D.

接下来介绍的稳定性分析和实验结果表明，应用于阀门控制（以及可能的其他复杂工业系统）的分级学习似乎是指导 RL 智能体的有效方法。

表 2.分级学习：分阶段学习参数和训练时间。

级别	L	f_s	f_s	迭代	时间 （h）
I.1 级	0.1	110×8.4	110×3.524	930	1.67
I.2 级	0.1	110×8.4	110×3.524	2000	12.35
二级	0.5	15×8.4	15×3.524	1000	5.31
III 级	1.5	12×8.4	12×3.524	1000	5.21
IV 级	1.5	23×8.4	23×3.524	1000	4.65
V 级	2.0	23×8.4	23×3.524	500	2.27
VI 级	2.5	8.4	3.524	2000	7.59
总和				8430	39.05

6. 实验、结果和讨论

在本节中，我们将介绍为评估 RL 控制器的性能并将其与 PID（带滤波器）控制器进行比较而进行的实验结果。

在进行实验之前，必须对 RL 控制器进行稳定性分析。

6.1. RL 控制的稳定性分析

本节尝试对 RL 控制进行基本稳定性分析。

系统的开环传递函数为$C(s)\cdot P(s)$.工厂的传递功能$P(s)=V(s)\cdot G(s)$，$G(s)$是FOPTD 过程的传递函数 （13） 和$V(s)$是未知的非线性阀的传递函数，必须估计。

Simulink 的控制设计线性化分析™工具提供了一个基于 GUI 的界面，用于生成非线性系统的线性近似值，并在指定的输入和输出点上进行计算。但是，与 MATLAB 的 tfest 函数相比，这不允许对估计进行任何控制。

编程方法允许用户通过指定极点数 （np） 和零点 （nz） 来估计传递函数。

此外，iodelay 参数允许在物理系统中试验时间延迟的影响。

```
sys = tfest(data, np, nz, iodelay)
```

框图 23 显示了数据点u_1和y_1将用于估计控制器传递函数$C(s)$和积分u_2和y_2估计完整的植物传递函数$P(s)$.图 24 是用于辅助估算的 Simulink 设置。

估计的传递函数：被控对象的估计连续时间传递函数为（20），拟合度为97.15%，MSE 为 0.7921，而控制器的连续时间传递函数为（21）。

$$\frac{0.002255s^2 - 1.904\times10^{-5}s + 8.563\times10^{-7}}{s^3 + 0.01305s^2 + 9.451\times10^{-5}s + 2.278\times10^{-7}} \tag{20}$$

$$\frac{0.09455s^2 + 0.0005729s + 1.609\times10^{-6}}{s^3 + 0.2312s^2 + 0.001939s + 1.195\times10^{-7}} \tag{21}$$

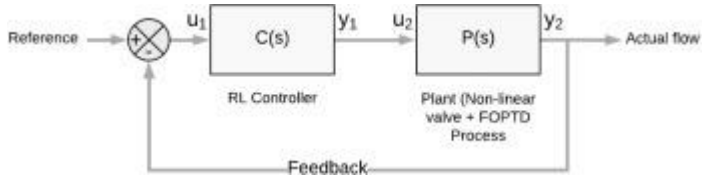


图 23.单回路控制系统框图。

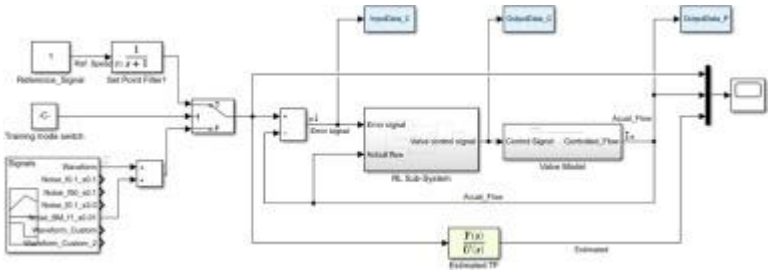


图 24.传递函数估计的设置。

我们绘制了图 25，使用估计的传递函数与实际响应的植物响应;确保估计值对于进行基本稳定性分析是合理的。

稳定性分析：图 26 中的阶跃响应显示了一个稳定的闭环系统。图 27 所示的开环波特图显示增益裕度为 10.9 dB，相位裕度为 68.0 度，表明系统相当稳定。

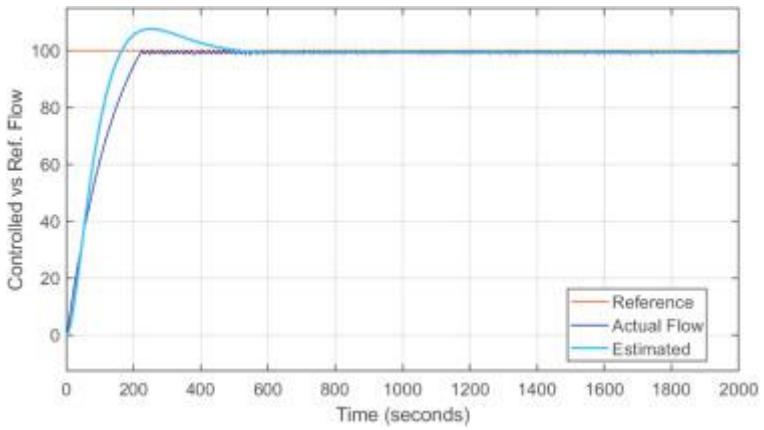


图 25.稳定性分析：使用估计的传递函数比较植物响应。

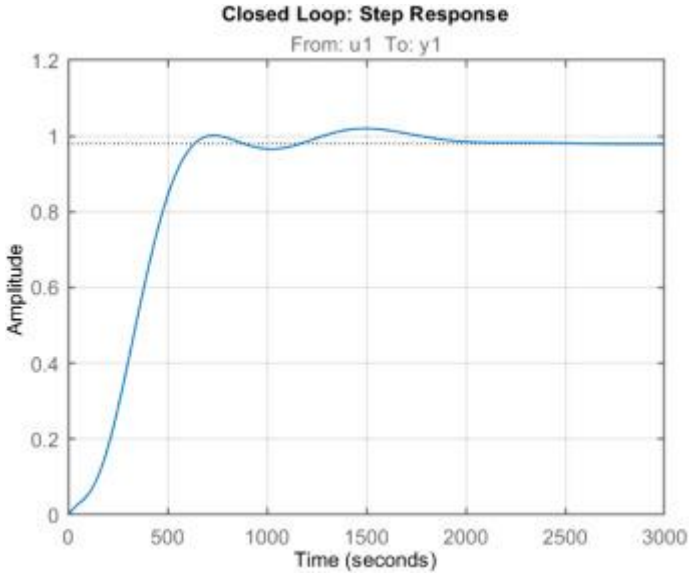


图 26.RL 控制器：阶跃响应。

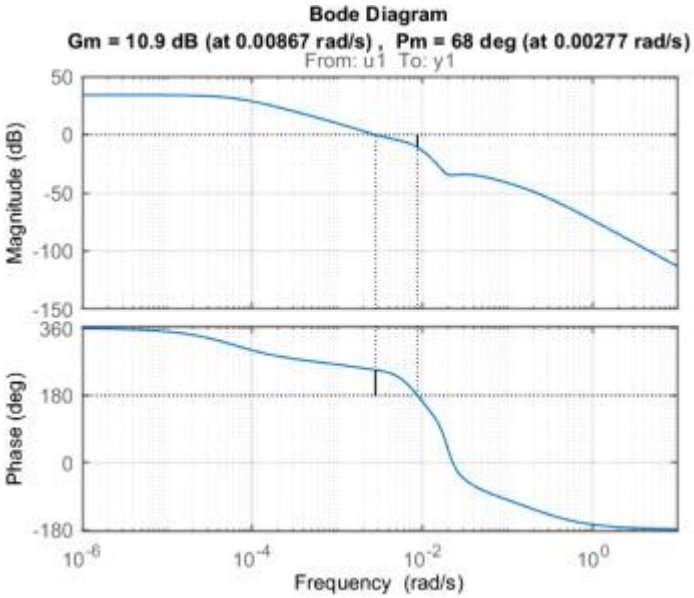


图 27.RL 控制器：开环波特环。

6.2. 实验和结果

在本节中，我们将介绍在统一框架上进行的实验结果，该框架测试了两种阀门控制策略——PID（带滤波器）和 DDPG RL。本文对具有不同控制信号、不同噪声强度和干扰点的实验以及具有过程回路扰动的工厂的影响进行了关键的时域分析。

进行的实验：

- 1.
  任意假定恒定的参考电平。

- 2.

基准波形（带噪声）。

- 3.
  基准波形在以下位置受到干扰：
  - 控制器输入（即参考信号）。
  - 工厂输入（即馈送到工厂的受控信号）。
  - 工厂输出（即系统输出）。
- 4.
  “供水”阀的实际例子，受到过往列车的地面振动的影响。
- 5.
  工厂经历过程回路扰动。
- 6.
  任意控制波形。

6.2.1. 实验-1：恒定参考信号

实验：基本分析最好在任意设置为 100 的简单恒定参考流速上进行，运行时间超过 2000 秒。 参考信号与基准高斯噪声 $(\mu = 0.0, \sigma = 0.01)$.

观察结果：在图 28 中，PID 显示出较大的过冲，并在大约 700 秒内建立。RL 策略显示出接近理想的阻尼和更快的建立时间（约 220 s）。RL 轨迹显示出与 PID 更平滑的轮廓相比的微小涟漪。这些振荡会降低机械系统的剩余使用寿命（RUL），我们通过进行（简化的）双因素 DOE（实验设计）来研究这一点。

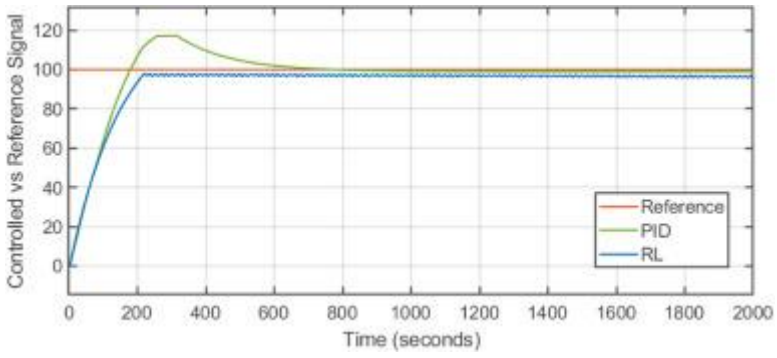


图 28.Expt.-1：恒定参考信号。

我们改变了这两个因素;延时和阀门摩擦（静态和动态组合），如表 3 所示。time-delay 的默认值L=2.5、静摩擦 f_S =8.40 和 f_D =3.524 被视为高电平，我们将每个电平降低 100 倍以获得低电平，如表 4 所示。

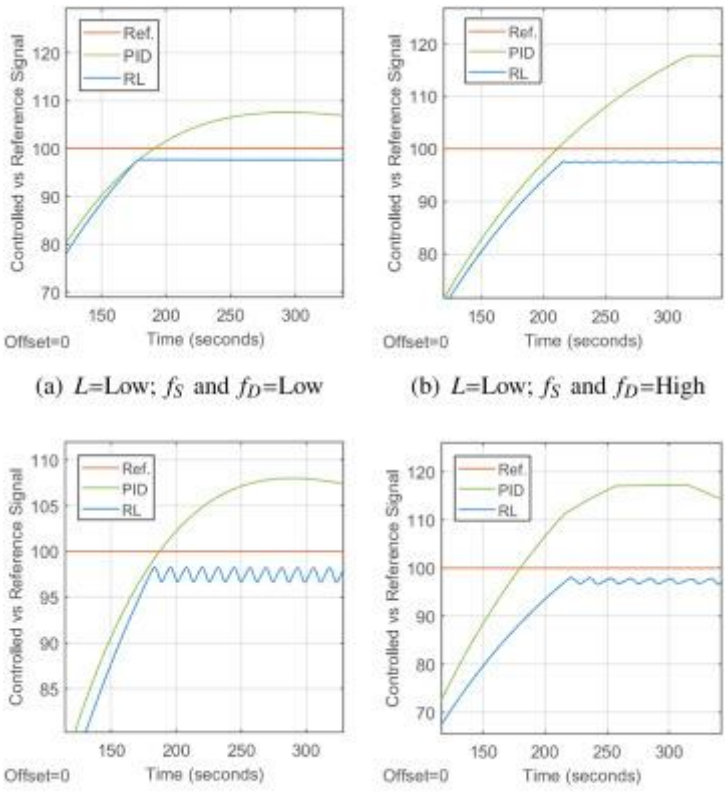
图 29（a）突出显示了当两个因子都较低时，RL 产生非常平滑轮廓的能力。这意味着振荡不是由 RL 技术引入的。图 29（c）显示，振荡行为的原因主要是由于时滞因素。

表 3.DoE 表。

延时 （*L*）	**摩擦值 （f_S，f_D）**
低	低
低	高
高	低
高	高

Table 4. DoE table with actual values.

L	f_S	f_D
0.025	0.084	0.0352
0.025	8.400	3.524
2.500	0.084	0.0352
2.500	8.400	3.524



(a) L=Low; f_S and f_D=Low (b) L=Low; f_S and f_D=High

(c) L=High; f_S and f_D=Low (d) L=High; f_S and f_D=High

图 29. Expt.-1：具有延时和摩擦参数的 DoE。

虽然 PID 策略是通过抑制噪声的滤波器实现的，但 RL 设置中没有添加滤波器，以更好地了解 RL 控制策略的自然响应。

6.2.2. 实验 2：基准信号

实验：使用波形剖面，高斯噪声 $(\mu = 0.0, \sigma = 0.01)$，受制于这两种策略。我们还放大了时域图（图 30）的部分，并在图 31 中更仔细地观察它们。
据观察，PID 显示出更高的过冲和欠冲。如果这样的阀门控制流体流量，则较高和较低的流体量可能会对产品质量造成不利影响。在 30 秒内，如果 PID 波形偏移取决于流体流动的时间，则 800 秒后偏移的 PID 波形可能对过程有害。相比之下，RL 控制显示出对参考信号的更好跟踪。

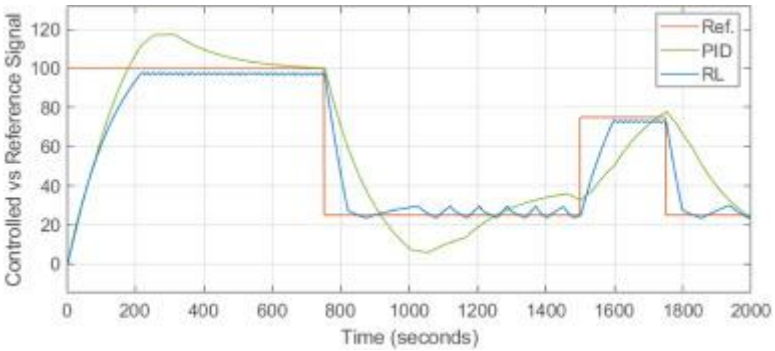
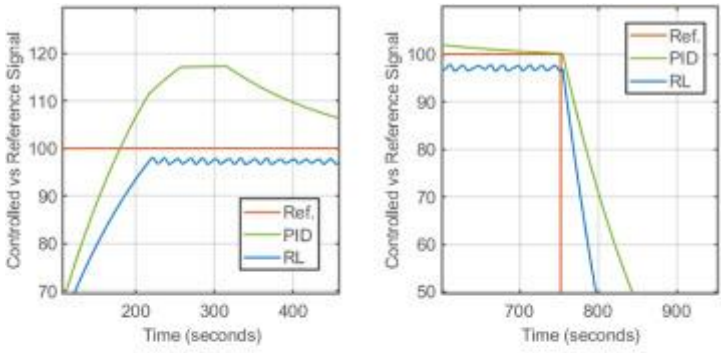
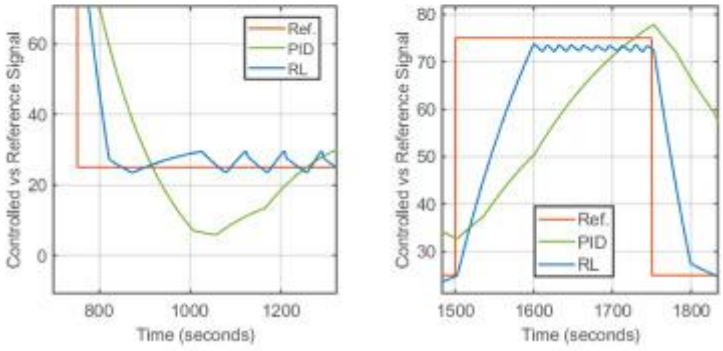


图 30.Expt.-2：基准波形。



(a) Zoomed section 1　　(b) Zoomed section 2



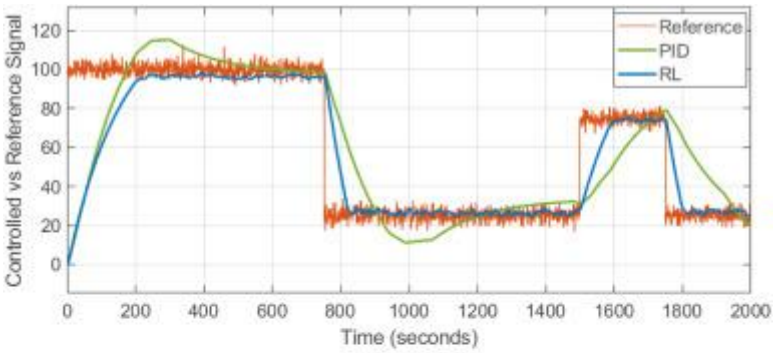
(c) Zoomed section 3　　(d) Zoomed section 4

图 31.Expt.-2：基准信号的缩放部分。

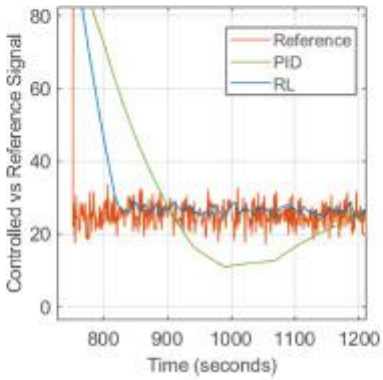
6.2.3. 实验-3.a：控制器输入端的噪声

实验：增加控制器输入端的噪声 $(\mu = 0, \sigma = 3.0,1 \quad \text{Hz})$

观察结果：图 32 显示，与实验 2 相比，对 PID 几乎没有影响（输入噪声较低），但对 RL 轨迹的影响增加，表明 PID 策略具有出色的噪声衰减能力。RL 继续密切跟踪参考信号（以及噪声）。



(a) Entire trajectory plot



(b) Zoomed section

图 32. Expt.-3. a：控制器输入端的噪声 $(\mu = 0, \sigma = 3.0,1 \quad \text{Hz})$

6.2.4. 实验-3. b：被控对象输入端的噪声

实验：将噪声源转移到被控对象输入端 $(\mu = 0, \sigma = 3.0,1 \quad \text{Hz})$

观察结果：图 33 显示 PID 轨迹现在受到影响，并且失去了在实验 1 和实验 2 中看到的相对平滑的输出。另一方面，与实验-1 相比，RL 策略*不受影响*。PID 策略根据误差信号进行自我调整，因此表现出行为变化，而 RL 策略则不会。

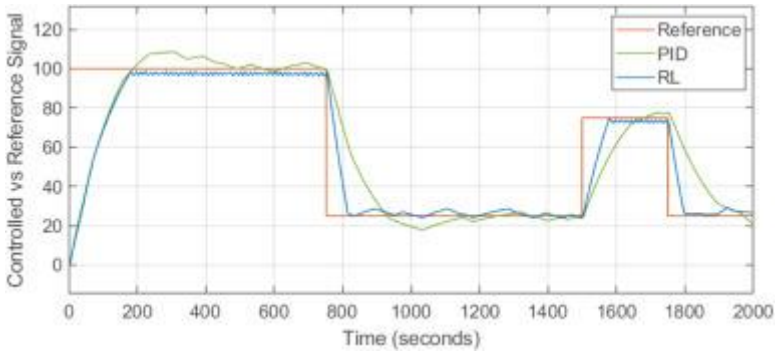
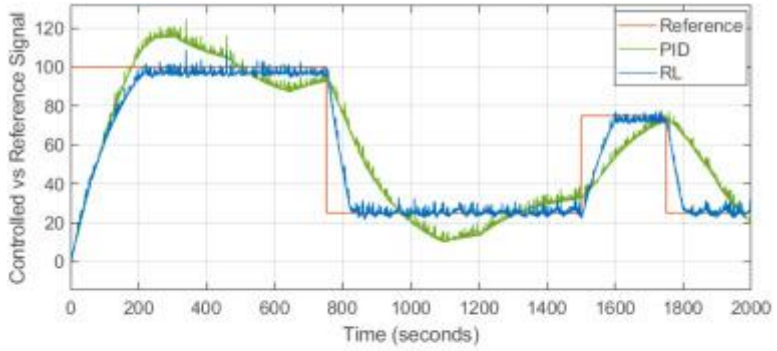


图 33. Expt.-3. b：被控对象输入端的噪声 $(\mu = 0, \sigma = 3.0,1 \quad \text{Hz})$.

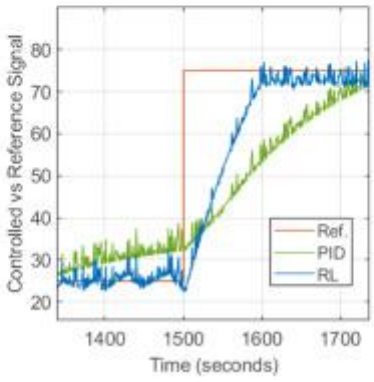
6.2.5. 实验-3. c：被控对象输出时的噪声

实验：噪声对被控对象输出的影响 $(\mu = 0, \sigma = 3.0,1 \quad \text{Hz})$

观察结果：图 34（a） 显示 RL 和 PID 策略受到的影响相同。



(a) Entire trajectory plot



(b) Zoomed section

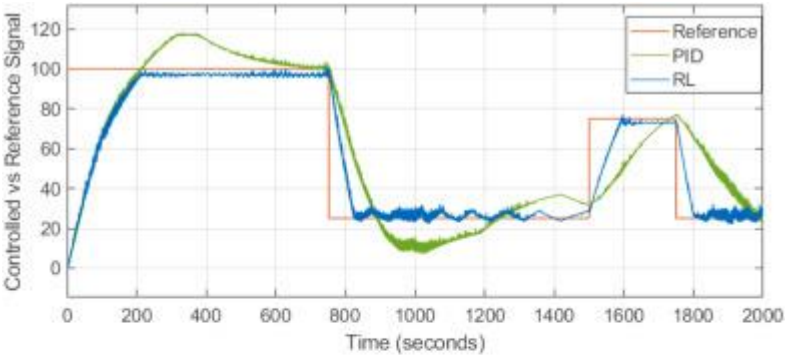
图 34. Expt-3c：工厂输出噪声

6.2.6. 实验-4：受地面振动影响的供水阀

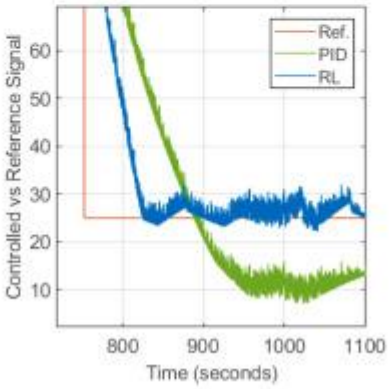
实验：阀门应用经常暴露在极其恶劣的条件下。例如，供水系统可能面临地面振动，例如来自过往铁路的振动，其范围约为 30-200 Hz，振幅各不相同。由

于控制阀组件通常放置在屏蔽环境中，因此假设频率在 30–100 Hz 之间进行仿真。

观察结果：图 35 表明，与实验-3.c 一样，噪声对两种策略的影响相似，RL 继续比 PID 更好地跟踪参考信号。



(a) Entire trajectory plot



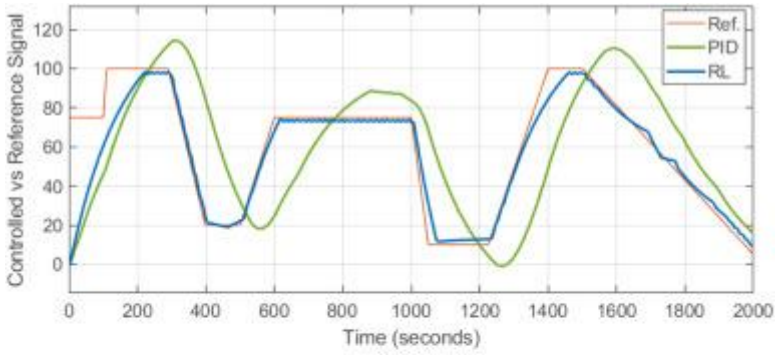
(b) Zoomed section

图 35.例-4：经过的列车的地面振动。

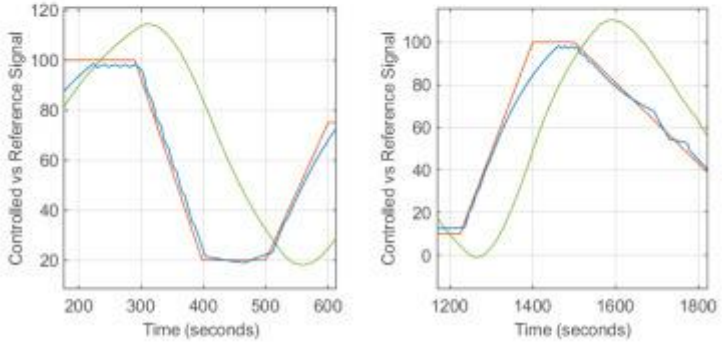
6.2.7. 实验-5：具有基准噪声信号的任意控制波形

实验：测试 RL 训练策略相对于 PID 调优泛化的泛化能力。两种策略的“训练”信号都是基准波形，本实验使它们受到完全不同的波形的影响。

观察结果：图 36 显示，在本实验中，RL 控制器的性能明显优于 PID 策略。RL 控制器更密切地跟踪任意参考，这证明了训练策略在有效泛化中的重要性。另一方面，PID 轨迹在跟踪参考时显示出明显的滞后，如果这样的阀门控制流体流量，则不合时宜的流体量升高或降低可能会损害产品质量。



(a) Arbitrary control waveform



(b) Zoomed section 1 (c) Zoomed section 2

图 36.Expt.-5：对任意控制波形的响应。

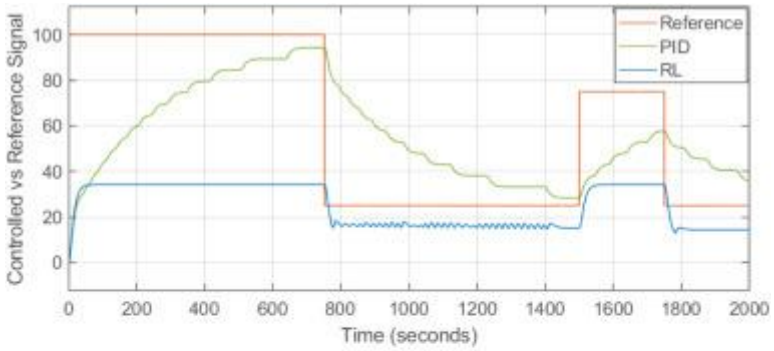
在 RL 控制轨迹的部分中，小的涟漪是显而易见的。

6.2.8. 实验-6：具有过程回路扰动的基准工厂

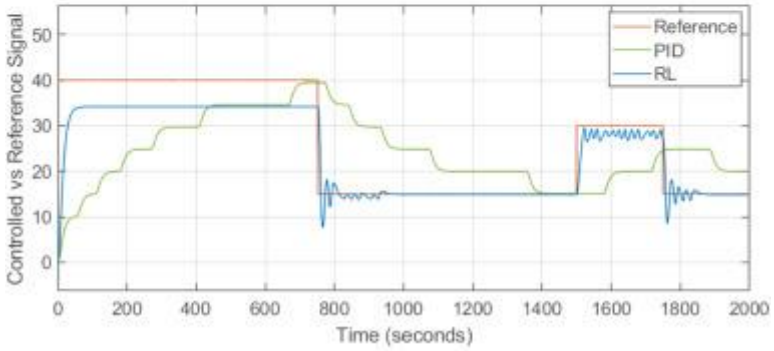
实验：评估对模拟为三阶传递函数的严重过程环路扰动的抵抗力。

$$G(s)=\frac{1}{(1s+1)(5s+1)(10s+1)}$$

观察结果：RL 控制器的严重*局限性*在该实验中显而易见。图 37 显示了一个明显发育不良的输出，在 35.0 左右平滑地夹紧。然后，在图 37（b）中较低幅度的参考上测试该装置，RL 继续箝位在同一水平 35.0。在扰动的影响下，PID 似乎会缩放到不同的水平，尽管存在很大的误差。RL 控制器在较低流量水平下显示出增加的振荡行为。



(a) Response to benchmark signal



(b) Response to benchmark signal with lower strength

图 37. Expt.-6：对受扰动的植物的响应。

7. 结论

参数调优需要大量的努力和耐心，才能构建一个稳定的控制器。
以上进行了实验以评估 RL 与传统 PID 控制。RL 策略的轨迹跟踪似乎优于 PID，而与 RL 控制信号上出现的干扰相比，PID 表现出更好的干扰抑制。PID 似乎滞后于参考控制信号，而 RL 控制器在跟踪未经过训练的控制配置文件时表现更好，并且在应用于同一环境中的不同控制任务时将表现出多功能性，而无需重新训练。

总体而言，RL 控制过程似乎保证了更好的过程质量，而 PID 控制过程将大大降低阀门操作的应力，并减少磨损。

增强功能和未来工作：设计的 RL 控制器需要一种机制来降低存在高振幅高频干扰时的振荡行为。对于控制器输入和输出端的噪声，低通滤波器可能有助于降低高方差。
有必要进一步研究定义目标和奖励函数的方法，以防止嘈杂的 RL 轨迹行为。如果成功，这将是比应用过滤器更好的解决方案，否则会减慢响应速度。