

基于 BP 神经网络 PID 控制参数自适应优化^{*}

张硕¹ 骆星九² 张孝强² 杨猛^{2†} 陈立群¹

(1. 上海大学 力学与工程科学学院, 上海 200072)

(2. 海军军医大学 中国人民解放军海军特色医学中心, 上海 200433)

摘要 本研究针对传统的反向传播(BP)神经网络比例积分微分(PID)算法存在的波动振荡、容易发散的问题,提出改进方法.首先采用 He 初始化方法对神经网络进行初始化,并对学习率进行衰减,同时对算法梯度进行裁剪.然后,在此基础上继续对比研究不同激活函数(Sigmoid、Tanh、ReLU 和 Leaky ReLU)和不同平滑技术(指数平滑、移动平均、Savitzky-Golay 滤波器和 Butterworth 滤波器)对算法性能的影响.最后,采用极端激励对平滑器的鲁棒性进行测试.结果表明:相较于传统的 Sigmoid 激活函数,使用 ReLU、Leaky ReLU 和 Tanh 激活函数具有更强的稳定性,并且 Leaky ReLU 激活函数综合性能最好;在平滑效果方面,指数平滑和 Savitzky-Golay 滤波器具有更明显的优势,更适合于需要快速响应和精确平滑的应用领域;平滑技术可以使算法更快恢复稳定,提高算法稳定性.

关键词 改进的 BP 神经网络, 平滑技术, 激活函数, 参数优化

中图分类号: TP273; TP183

文献标志码: A

Adaptive Optimization of PID Control Parameters Based on BP Neural Network^{*}

Zhang Shuo¹ Luo Xingjiu² Zhang Xiaoqiang² Yang Meng^{2†} Chen Liqun¹

(1. School of Mechanics and Engineering Science, Shanghai University, Shanghai 200072, China)

(2. PLA Naval Medical Center, Naval Medical University, Shanghai 200433, China)

Abstract In this study, an improved method is proposed to solve the problems of fluctuating oscillation and easy divergence of the traditional back propagation (BP) neural network proportional integral differential (PID) algorithm. Firstly, the He initialization method is used to initialize the neural network, the learning rate is attenuated, and the algorithm gradient is clipped. Then, on this basis, the effects of different activation functions (Sigmoid, Tanh, ReLU and Leaky ReLU) and different smoothing techniques (exponential smoothing, moving average, Savitzky-Golay filter and Butterworth filter) on the performance of the algorithm are further compared. Finally, the robustness of the smoother is tested by extreme disturbance. The results show that compared with the traditional Sigmoid activation function, the ReLU, Leaky ReLU and Tanh activation functions have stronger stability, and the Leaky ReLU activation function has the best comprehensive performance. In terms of smoothing effect, exponential smoothing and Savitzky-Golay filters have more obvious advantages and are more suitable for applications that require fast response and precise smoothing. The smoothing techniques can make the algorithm recover faster and improve the stability of the algorithm.

Key words improved BP neural network, smoothing techniques, activation function, parameter optimization

2025-02-15 收到第 1 稿, 2025-02-25 收到修改稿.

^{*} 军队装备综合研究项目(24AZ0502), Comprehensive Research Project on Army Equipment (24AZ0502).

[†] 通信作者 E-mail: zhousidun@sohu.com

引言

传统的比例积分微分(PID)控制算法因其强大的适应性和易于调整优化的特点使其成为自动控制领域中应用最广泛的反馈控制算法之一^[1]. 该算法通过计算系统的误差(即期望值与实际输出之间的差异)来生成控制信号,进而调整系统行为以实现稳定控制. 但是 PID 控制算法也存在明显的局限性,主要包括缺乏预测能力以及参数调整的复杂性^[2].

为了克服这些限制,研究者们开始探索使用先进的智能控制算法,如模糊控制 PID^[3]、神经网络 PID^[3]和遗传算法 PID^[4]等,这些算法强大的非线性拟合能力和自我学习能力^[5,6]能够很好地弥补传统 PID 控制算法的缺陷^[7]. 反向传播(BP)神经网络 PID 控制^[8-9]是一种先进的控制方法,它将传统的 PID 控制器与人工神经网络的自适应能力相结合. 然而,传统的 BP 神经网络在训练过程中存在很多问题,如网络层数、神经元个数的选择缺乏理论指导,以及学习速度慢、波动振荡^[10]等问题. 针对上述传统 BP 神经网络算法存在的波动振荡和容易发散的问题,研究人员已提出了各种改进的 BP 神经网络控制算法:文献[11]提出了一种增加多个动量项的方法,以加快学习速度并保持稳定性;文献[12]通过优化权值修正公式和学习速率来解决算法收敛速度慢的问题;文献[13]提出了一种自适应调整动量系数的改进算法;文献[14]通过引入动量项和变步长法来提高算法的收敛速度;文献[15]利用附加动量项增加算法的稳定性. 以上优化改进主要集中在动态调整学习率^[12]和增加动量项^[11,13-15],但在实践应用中,这样的改进计算效果并不理想,波动振荡仍然存在. 因此,本研究旨在通过引入平滑器、改变激活函数,采用更稳定的初始化方法,并优化学习率,以实现消除波动振荡,降低发散可能性的目的.

1 基于 BP 神经网络的 PID 控制算法的改进方法

神经网络的训练性能取决于参数设置和算法选择. 本章首先对基于 BP 神经网络的 PID 控制算法进行参数优化,参数优化的主要措施包括:采用 He 初始化方法对神经网络进行初始化;对学习率进行衰减;对算法梯度进行裁剪. 然后在上述参数优化的基础上,重点讨论探讨 Sigmoid、Tanh、

ReLU 和 Leaky ReLU 函数作为隐含层激活函数的性能表现,通过对比这 4 种激活函数的优势和局限性,为后续的仿真实验提供理论依据;此外,为了提高神经网络输出的稳定性,引入 4 种平滑器,分别为指数平滑、移动平滑、Savitzky-Golay 滤波器和 Butterworth 滤波器,分别对比其平滑效果,以期能够减小波动振荡.

1.1 参数优化

1.1.1 学习率衰减

在赋予各层权值对应的学习速率后,每次迭代更新权值矩阵时,学习率均乘以预设的学习衰减因子 β ^[16],实现学习率的递减. 数学表达式为:

$$\eta \Rightarrow \beta \eta \quad (1)$$

其中: η 为当前的学习率, β 为学习衰减因子.

1.1.2 He 初始化

采用标准正态分布随机生成权重矩阵,权重 W 的初始值从均值为 0,方差为 $2/(n_{in} + n_{out})$ 的正态分布中抽取,即:

$$W \sim N\left(0, \frac{2}{n_{in} + n_{out}}\right) \quad (2)$$

其中: n_{in} 为前一层的神元数量, n_{out} 为当前层的神元数量, W 为初始权值矩阵.

1.1.3 梯度裁剪

本文引入一种梯度裁剪^[18]方法,通过设置梯度阈值来限制梯度的大小,如果梯度超过这个阈值,就将其强制裁剪至该范围内,实现过程如下:

$$g_t = \min[\max(g_{t-1}, -C), C] \quad (3)$$

其中: C 为梯度阈值, g_{t-1} 为裁剪前的梯度, g_t 为裁剪后的梯度.

本研究针对参数优化引入 3 种优化策略:学习率衰减、He 初始化和梯度裁剪,后续的研究都将在这些优化策略基础上展开.

1.2 激活函数

本节重点探讨 Sigmoid、Tanh、ReLU 和 Leaky ReLU 函数 4 种激活函数作为隐含层激活函数的性能表现,通过对比这 4 种激活函数的优势和局限性,为后续的仿真实验提供理论依据. 在 PID 控制领域,控制信号通常需要被限制在特定的范围内以确保系统的稳定性和响应性,Sigmoid 函数能够确保输出信号维持在这一预设范围内. 因此在本文后续的研究中,输出层的激活函数均使用 Sigmoid 函

数,实现对控制信号的有效调控。

1.2.1 Sigmoid 函数

Sigmoid 函数及其导函数表达式如下:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

$$\dot{S}(x) = \frac{e^{-x}}{(1 + e^{-x})^2} \quad (5)$$

1.2.2 Tanh 函数

Tanh 函数表达式如下:

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6)$$

1.2.3 ReLU 函数

ReLU 函数^[19]表达式如下:

$$\text{ReLU}(x) = \max(0, x) \quad (7)$$

1.2.4 Leaky ReLU 函数

Leaky ReLU 函数^[20]表达式如下:

$$\text{Leaky ReLU}(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (8)$$

1.3 平滑器

梯度下降法的局限性^[21]、不恰当的学习速率和干扰噪声都会导致神经网络在最优解附近发生振荡,无法稳定地收敛到最优解。通过应用平滑器,可以降低这些波动,使得输出结果更加稳定,从而提高控制的精确度和可靠性。现引入 4 种平滑器,分别为指数平滑、移动平滑、Savitzky-Golay 滤波器和 Butterworth 滤波器。

1.3.1 指数平滑

指数平滑^[21,22]是一种时间序列预测方法,它给过去的观测值赋予指数递减的权重,其基本公式为:

$$S_t = \alpha y_t + (1 - \alpha)S_{t-1} \quad (9)$$

其中: S_t 为 t 时刻的平滑值, y_t 为 t 时刻的实际观测值, α 为平滑因子, S_{t-1} 为前一时间的平滑值。

1.3.2 移动平滑

移动平滑是一种简单的数据平滑方法^[23],通过计算一定窗口内数据点的平均值来减少噪声和波动,同时保留数据的趋势,其基本公式为:

$$S_t = \frac{\sum_{i=1}^n (y_{t-i} + y_{t+i}) + y_t}{2n + 1} \quad (10)$$

其中: n 为滑动窗口半径, y_t 为 t 时刻的实际观测值,为 t 时刻的平滑值。

1.3.3 Savitzky-Golay 滤波器

Savitzky-Golay 滤波器^[24]基本公式为:

$$S_t = \sum_{k=0}^n b_{nk} t^k \quad (11)$$

其中: n 为多项式的次数, b_{nk} 为多项式的系数,可以由最小二乘法求出。

1.3.4 Butterworth 滤波器

Butterworth 滤波器^[25]基本公式为:

$$S_t = \frac{1}{\sqrt{1 + \left(\frac{b}{\omega_c}\right)^{2n}}} \quad (12)$$

其中: b 为复频率变量 ω_c 是截止频率, n 是滤波器的阶数。

1.4 算法原理

通过上述改进后的 BP 神经网络的 PID 控制算法,其结构如图 1 所示。为了更好地贴近工程实际,本文均采用在线平滑技术,具体步骤如下:

(1)确定参数,确定初始学习因子、学习率衰减因子、各层神经元个数等参数;

(2)初始化,使用 He 初始化方法初始化隐藏层和输出层的权重;

(3)计算,计算隐藏层和输出层的输入和输出;

(4)输出,将输出的 PID 参数(即 K_p 、 K_i 和 K_d)输入 PID 控制器,PID 控制器输出控制量;

(5)更新,计算并更新输出层和隐藏层权重系数,并调节更新 PID 控制参数;

(6)平滑器实现,使用在线平滑技术对输出进行平滑处理;

(7)循环,返回步骤 1。

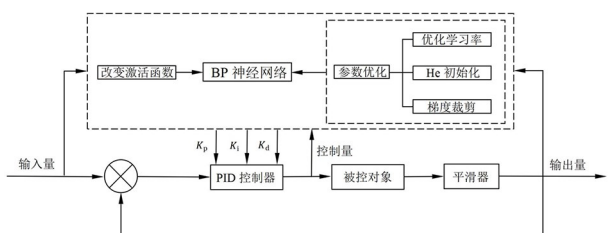


图 1 改进后 BP 神经网络的 PID 控制系统结构简图
Fig. 1 Schematic diagram of PID control system structure of improved BP neural network

2 仿真分析

为了验证本文理论与方法的有效性,本节将进行一系列仿真实验^[26-28],以评估改进后的 BP 神经网络 PID 控制算法的性能。本研究中所采取的非线性仿真系统^[29]模型,该模型能够模拟复杂系统的动态行为:

$$y(k) = \frac{a(k)y(k-1)}{1+y^2(k-1)} + u(k-1) \quad (13)$$

其中: $a(t) = 1.3(1 - 0.6e^{-0.2t})$.

具体实现过程如下:首先,分析 Sigmoid、Tanh、ReLU 和 Leaky ReLU 函数作为隐含层激活函数时系统性能的表现;然后,评估指数平滑、移动平滑、Savitzky-Golay 滤波器和 Butterworth 滤波器 4 种平滑器对输出曲线平滑性能的影响;最后,引入极端噪声干扰,对上述 4 种平滑器的鲁棒性进行测试^[30-32].

2.1 激活函数对比

本节通过改进的 BP 神经网络 PID 控制器 (BP-PID) 算法,研究不同激活函数对系统控制性能的影响.在神经网络的运算中,隐含层神经元个数、学习率、动量因子这三个参数对系统的运算性能影响最显著,为确保研究结论的普适性和准确性,现通过改变隐含层神经元个数 N 、学习率 η 、动量因子 α 这三个参数,对 8 组不同的参数配置进行仿真对比实验: ($N_1=9$ 、 $N_2=6$; $\eta_1=0.1$ 、 $\eta_2=0.4$; $\alpha_1=0.5$ 、 $\alpha_2=1$).

设置算法的关键初始条件和参数.

2.1.1 网络结构

隐含层神经元数量: $N_1=9$ 、 $N_2=6$;

输入层节点数: 4[对应输入信号 $r(t)$ 、系统输出 $y(t)$ 、误差 $e(t)$ 以及偏置项];

输出层节点数: 3(对应 PID 控制系统的 3 个参数 K_p 、 K_i 和 K_d).

2.1.2 权重初始化

采用 He 初始化方法对权重进行初始化:

隐含层权重矩阵 W_h 初始化为随机值,范围符合 $\sqrt{2/(n_{in} + n_{out})}$,其中 $n_{in}=4$ 和 $n_{out}=9$;

输出层权重矩阵 W_o 初始化为随机值,范围符合 $\sqrt{2/(n_{in} + n_{out})}$,其中 $n_{in}=10$ 和 $n_{out}=3$.

2.1.3 学习率与衰减因子

初始学习率 $\eta_1=0.1$ 、 $\eta_2=0.4$;

学习衰减因子 $\beta=0.999$.

2.1.4 控制参数

动量因子 $\alpha_1=0.5$ 、 $\alpha_2=1$;

采样时间 $T_s=0.001$.

2.1.5 数值稳定性措施

梯度裁剪阈值 $C=1$;

2.1.6 训练迭代次数

训练过程共进行了 $T=5000$ 步的迭代.

通过上述初始条件和参数设置,本文能够清晰地展示不同激活函数在 BP-PID 控制器中的表现差异,为算法实现提供明确指导,并为后续研究者提供参考框架.

2.1.7 仿真对比实验

对 8 组不同的参数配置进行仿真对比实验:以阶跃信号^[33]为例,对 Sigmoid、Tanh、ReLU 和 Leaky ReLU 函数作为隐含层激活函数时系统性能的表现进行对比.如图 2 所示,在改变了隐含层的激活函数后,系统的振荡幅度得到了显著地减小.表 1 对比了图 2(a)中 4 种情况下系统响应的上升时间、峰值时间和超调量.结果表明:相较于传统的 Sigmoid 激活函数,使用 ReLU、Leaky ReLU 和 Tanh 激活函数具有更强的稳定性.其中,ReLU 激

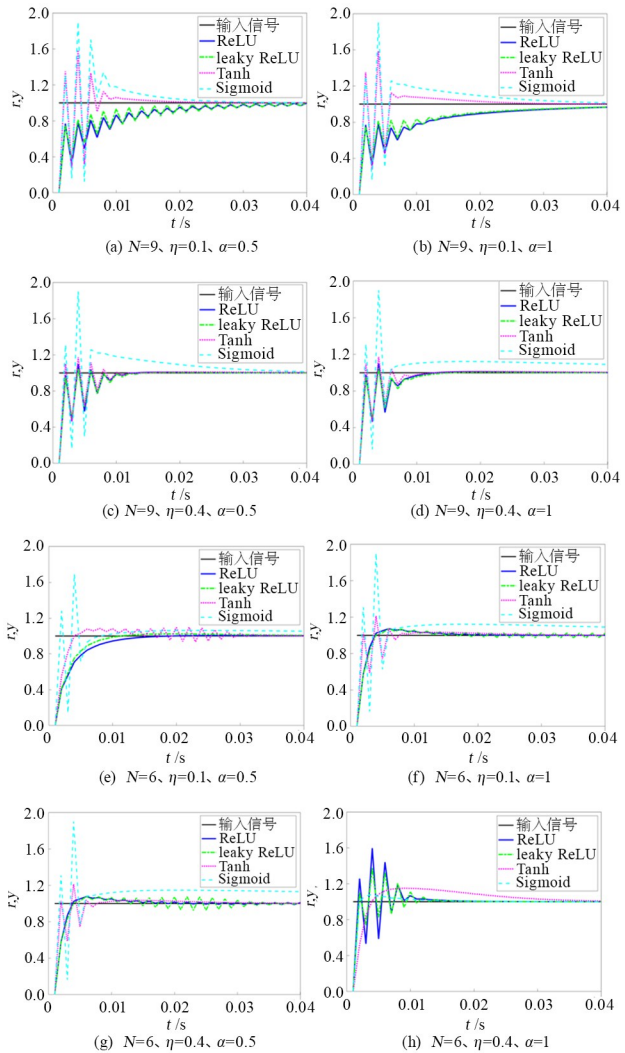


图2 不同参数下使用4种不同激活函数的系统输出曲线
Fig.2 System output curves using 4 different activation functions for different parameters

表 1 图 2(a)中不同激活函数下的系统动态性能比较
Table 1 Comparison of the dynamic performance of the system under different activation functions in figure 2(a)

激活函数类型	上升时间/s	峰值时间/s	超调量/%
ReLU	0.014	0.041	1.019
Leaky ReLU	0.009	0.030	0.438
Tanh	0.001	0.004	58.609
Sigmoid	0.001	0.004	90.148

活函数在收敛速度上明显快于 Leaky ReLU 和 Tanh 激活函数,尽管 ReLU 激活函数自身存在局部振荡. Leaky ReLU 激活函数综合了 Sigmoid 和 ReLU 激活函数的优势与不足,在本文后续的研究中,均使用 Leaky ReLU 作为隐含层的激活函数.

2.2 平滑器对比

为了评估 4 种平滑器对系统输出稳定性的影响,首先对算法参数进行设置(初始条件及参数设置同 2.1 节),接着对各种平滑器的参数进行设置:指数平滑的平滑因子 $\alpha=0.1$;移动平均滑动窗口半径 $n=10$;Savitzky-Golay 滤波器的多项式的次数 $n=3$,窗口长度为 21;Butterworth 滤波器的复频率变量 $b=1/t$,截止频率 $\omega_c=10$,滤波器的阶数 $n=3$.

为确保研究结论的一般性和准确性,对 8 组不同的参数配置进行仿真对比实验:($N_1=9$ 、 $N_2=6$; $\eta_1=0.1$ 、 $\eta_2=0.4$; $\alpha_1=0.5$ 、 $\alpha_2=1$). 图 3 为不同参数下使用 4 种平滑器的系统输出曲线. 表 2 列出了应用 4 种不同平滑器后系统响应的上升时间、峰值时间和超调量[以图(a)为例].

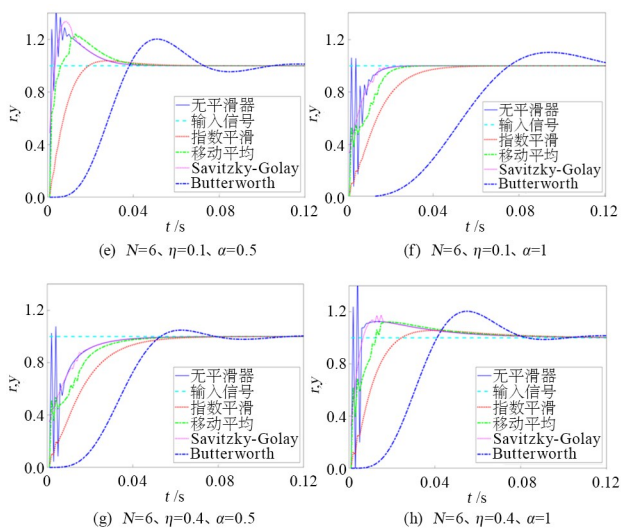
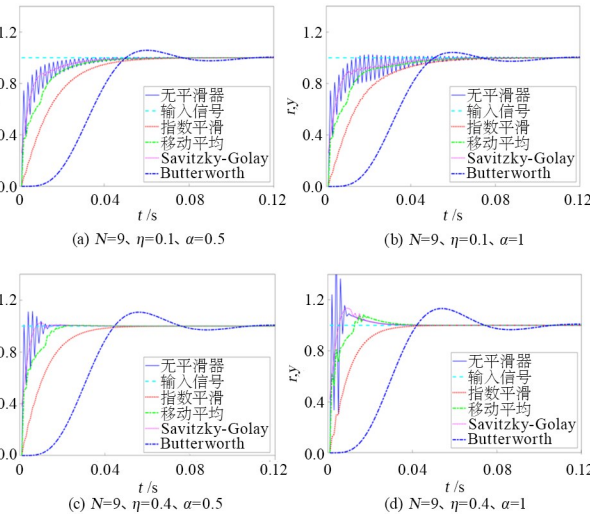


图 3 不同参数下使用 4 种平滑器的系统输出曲线
Fig. 3 System output curves using 4 smoothers for different parameters

结果表明:在应用了 4 种平滑器后,系统的振荡的问题得到了显著地解决. Savitzky-Golay 滤波器和移动平均比指数平均和 Butterworth 滤波器的收敛速度更快,但移动平均存在局部振荡现象. 在平滑效果方面,指数平滑和 Savitzky-Golay 滤波器具有更明显的优势,更适合于需要快速响应和精确平滑的应用领域.

表 2 图 3(a)中系统应用不同平滑器后的动态性能比较
Table 2 Comparison of the dynamic performance of the system after applying different smoothers in figure 3(a)

平滑器类型	上升时间/s	峰值时间/s	超调量/%
指数平滑	0.046	0.071	0.198
移动平滑	0.037	0.061	0.438
Savitzky-Golay	0.022	0.057	0.586
Butterworth	0.041	0.048	4.148

2.3 算法对比

现有的 BP 神经网络算法的改进措施主要有两种:一种是添加动量项^[29]的 BP 神经网络 PID 控制算法(MBPPID),另一种是动态调整学习率^[34]的 BP 神经网络 PID 控制算法(VLBPPID). 为了验证本文算法的有效性,现将本文算法与现有的 BP 神经网络算法进行对比实验.

首先对 MBPPID 算法进行参数设置:动量因子 $\alpha_1=0.5$ 、动量因子 $\alpha_2=0.1$;学习率 $\eta=0.1$;隐含层神经元数 $N=9$;输入层节点数为 4;输出层节点数为 3;采样时间 $T_s=0.001$.

接着对 VLBPPID 算法进行参数设置:初始学习率 $\eta=0.1$;动态学习阈值参数 $\xi_1=0.04$, $\xi_2=0.05$;隐含层神经元数 $N=9$;输入层节点数为 4;输出层节点数为 3;采样时间 $T_s=0.001$.

最后,本文算法对参数的设置同 2.1 节(隐含层神经元数量: $N=9$,学习率 $\eta=0.1$,动量因子 $\alpha=0.5$),隐含层采用 Leaky ReLU 激活函数,并采用 Savitzky-Golay 平滑技术(同 2.2 节).最后在图 4 展示了 3 种控制算法的输出曲线对比结果.从图中可以明显看出,本文提出的改进算法在稳定性方面相较于 MBPPID 和 VLBPPID 算法有显著提升.

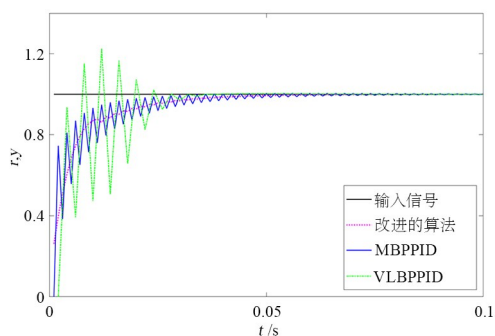


图 4 3 种控制算法的输出曲线

Fig. 4 Output curves of the three control algorithms

2.4 鲁棒性测试

为了评估系统在引入平滑器后对干扰的鲁棒性,在 0.3 s 时刻对系统施加了一个大小为 0.2,方向为负的极端干扰信号.在此基础上,对指数平滑、移动平均和 Savitzky-Golay 滤波器进行鲁棒性测试.参数设置同 2.1 节,其中隐含层神经元数量 $N=9$,学习率 $\eta=0.1$,动量因子 $\alpha=0.5$.此外,平滑器参数设置同 2.2 节.根据图 5 可知,在系统受到干扰后,指数平滑、移动平均和 Savitzky-Golay 滤波器均能迅速响应并趋于稳定.表明这些平滑器在面

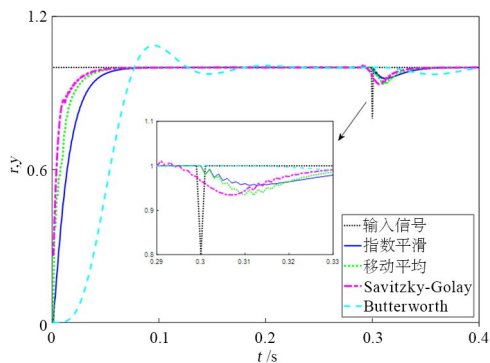


图 5 加入极端干扰后的系统输出曲线

Fig. 5 System output curve with the addition of extreme interference

对于干扰时的有效性和鲁棒性,能够在保持系统稳定性方面发挥关键作用.观察局部放大图可以发现,相较于其他 3 种平滑器,Butterworth 滤波器在受到极端干扰时的影响明显较小,具有更快的响应速度而且能更快地趋于稳定.

3 结论

本文提出了一种基于改进 BP 神经网络的 PID 参数自适应优化算法,并通过仿真实验验证了不同激活函数和平滑器对系统性能的影响,得出以下结论:

(1) Leaky ReLU 在梯度消失抑制、鲁棒性及收敛速度方面展现出综合优势.相较于 Sigmoid/Tanh,其正值区导数为 1 的特性加速了梯度更新;负值区小斜率设计($\alpha=0.01$)则缓解了神经元抑制问题,实现了动态响应与稳定性的平衡.

(2) Savitzky-Golay 滤波器通过多项式拟合在保留信号趋势的同时抑制高频噪声,适用于快速响应场景;Butterworth 滤波器凭借优化的通带—阻带特性,在抗干扰与长期稳定性上表现更优.

(3) 动态响应需求优先的场景推荐采用 Leaky ReLU 与 Savitzky-Golay 滤波器组合;复杂噪声环境下,建议选用 Butterworth 滤波器以提升鲁棒性.两类方法的核心差异源于频谱处理机制:前者注重局部特征提取,后者通过频率响应优化实现全局稳定性.

综上所述,本文提出的改进 BP 神经网络 PID 控制算法通过引入 Leaky ReLU 激活函数和多种平滑器,显著提升了系统的动态响应性能和稳定性.这些改进不仅提高了控制精度,还为非线性复杂系统的 PID 控制提供了一种新的解决方案.

参考文献

- [1] BENNETT S. The past of pid controllers [J]. Annual Reviews in Control, 2001, 25: 43—53.
- [2] MESHRAM P M, KANOJIYA R G. Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor [C]//IEEE-International Conference on Advances In Engineering, Science and Management (ICAESM-2012). New York: IEEE, 2012: 117—122.
- [3] 周春燕, 赵彪. 基于准零刚度隔振器的自适应模糊

- 控制[J]. 动力学与控制学报, 2023, 21(9): 74—82.
- ZHOU C Y, ZHAO B. Adaptive fuzzy control based on quasi-zero stiffness vibration isolator [J]. Journal of Dynamics and Control, 2023, 21(9): 74—82. (in Chinese)
- [4] 刘建文, 丁洁玉, 耿国志. 基于遗传算法的空间伸展臂展开过程优化控制[J]. 动力学与控制学报, 2016, 14(1): 64—69.
- LIU J W, DING J Y, GENG G Z. Optimal control on deployment process of space deployable mast based on genetic algorithm [J]. Journal of Dynamics and Control, 2016, 14(1): 64—69. (in Chinese)
- [5] 蒋鼎国. 基于改进型 BP 神经网络 PID 控制器的温室温度控制技术[J]. 实验室研究与探索, 2015, 34(1): 9—13.
- JIANG D G. The temperature control of greenhouse based on novel PID controller [J]. Research and Exploration in Laboratory, 2015, 34(1): 9—13. (in Chinese)
- [6] 汪圣祥, 金朝永. 基于 BP 神经网络的 PID 改进和研究[J]. 湖南理工学院学报(自然科学版), 2017, 30(1): 26—31.
- WANG S X, JIN C Y. Improvement and research of PID controller based on BP neural network [J]. Journal of Hunan Institute of Science and Technology (Natural Sciences), 2017, 30(1): 26—31. (in Chinese)
- [7] BORASE R P, MAGHADE D K, SONDKAR S Y, et al. A review of PID control, tuning methods and applications [J]. International Journal of Dynamics and Control, 2021, 9(2): 818—827.
- [8] 李永歌, 张潇, 许勇. 航空发动机分数阶 PID 控制器的参数自整定方法[J]. 动力学与控制学报, 2023, 21(7): 77—88.
- LI Y G, ZHANG X, XU Y. Parameter self-tuning method of fractional order PID controller for the aero engine [J]. Journal of Dynamics and Control, 2023, 21(7): 77—88. (in Chinese)
- [9] 李伟, 张美婷, 赵俊锋, 等. 含有分数阶 PID 控制器的随机动力系统可靠性分析[J]. 动力学与控制学报, 2019, 17(1): 65—72.
- LI W, ZHANG M T, ZHAO J F, et al. Reliability analysis of stochastic dynamical systems with fractional order PID controller [J]. Journal of Dynamics and Control, 2019, 17(1): 65—72. (in Chinese)
- [10] 刘天舒. BP 神经网络的改进研究及应用[D]. 哈尔滨: 东北农业大学, 2011.
- LIU T S. The research and application on BP neural network improvement [D]. Harbin: Northeast Agricultural University, 2011. (in Chinese)
- [11] 刘益民. 基于改进 BP 神经网络的 PID 控制方法的研究[D]. 西安: 中国科学院研究生院(西安光学精密机械研究所), 2007.
- LIU Y M. Research on PID control method based on improved BP neural network [D]. Xi'an: Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, 2007. (in Chinese)
- [12] 朱喜娜, 陆达, 范汉青. 基于 BP 算法 PID 控制器的研究[J]. 计算机技术与发展, 2010, 20(5): 183—186.
- ZHU X N, LU D, FAN H Q. Research of PID controller based on BP algorithm [J]. Computer Technology and Development, 2010, 20(5): 183—186. (in Chinese)
- [13] 刘莉, 刘强, 靳鸿, 等. 引入动量项的变步长 BP 网络预测算法[J]. 探测与控制学报, 2015, 37(5): 102—105.
- LIU L, LIU Q, JIN H, et al. BP network prediction algorithm with momentum term and variable step size [J]. Journal of Detection & Control, 2015, 37(5): 102—105. (in Chinese)
- [14] 唐明峰, 秦琳琳, 刘鹏, 等. 基于改进 BP-PID 干煤粉气化炉控制系统的研究[J]. 计算机仿真, 2021, 38(9): 276—280.
- TANG M F, QIN L L, LIU P, et al. Research on control system of dry pulverized coal gasifier based on improved BP-PID [J]. Computer Simulation, 2021, 38(9): 276—280. (in Chinese)
- [15] SILAA M Y, BARAMBONES O, BENCHERIF A. A novel adaptive PID controller design for a PEM fuel cell using stochastic gradient descent with momentum enhanced by whale optimizer [J]. Electronics, 2022, 11(16): 2610.
- [16] CAO W, SUN M. Iterative learning control for discrete-time switched systems with attenuation factor [J]. Journal of Vibration and Control, 2016, 22(12): 2898—2906.
- [17] DATTA L. A survey on activation functions and their relation with Xavier and He normal initialization [EB/OL]. (2020-03-18) [2025-02-15]. <https://arxiv.org/abs/2004.06632v1>.
- [18] ZHANG J Z, HE T X, SRA S, et al. Why gradient clipping accelerates training: a theoretical justification

- tion for adaptivity [EB/OL]. (2019-05-28)[2025-02-15]. <https://arxiv.org/abs/1905.11881v2>.
- [19] BANERJEE C, MUKHERJEE T, PASILIAO E JR. An empirical study on generalizations of the ReLU activation function [C]//Proceedings of the 2019 ACM Southeast Conference. New York: ACM, 2019: 164—167.
- [20] XU J, LI Z S, DU B W, et al. Reluplex made more practical: leaky ReLU [C]//2020 IEEE Symposium on Computers and Communications (ISCC). New York: IEEE, 2020: 1—7.
- [21] 史加荣, 王丹, 尚凡华, 等. 随机梯度下降算法研究进展[J]. 自动化学报, 2021, 47(9): 2103—2119.
- SHI J R, WANG D, SHANG F H, et al. Research advances on stochastic gradient descent algorithms [J]. Acta Automatica Sinica, 2021, 47(9): 2103—2119. (in Chinese)
- [22] RUAN S L, XIE H Y, JIANG S. Integrated proactive control model for energy efficiency processes in facilities management: applying dynamic exponential smoothing optimization [J]. Sustainability, 2017, 9(9): 1597.
- [23] 孙蓉, 李强, 罗海峰, 等. 基于自适应移动平滑与时间卷积网络误差修正的风电功率预测[J]. 全球能源互联网, 2022(1): 11—22.
- SUN R, LI Q, LUO H F, et al. Wind power forecasting based on error correction using adaptive moving smoothing and time convolution network [J]. Journal of Global Energy Interconnection, 2022(1): 11—22. (in Chinese)
- [24] KAWALA-STERNIUK A. Implementation of smoothing filtering methods for the purpose of improvement inverted pendulum's trajectory [J]. Przegląd Elektrotechniczny, 2020, 1(4): 112—115.
- [25] KOSZEWNIA A, TROCKI, SŁOWIK M. PID controllers design applied to positioning of ball on the Stewart platform [J]. Acta Mechanica et Automatica, 2014, 8(4): 214—218.
- [26] 聂小博, 熊玥, 潘勇军. 基于模糊PID算法的车身稳定控制策略与多工况联合仿真[J]. 动力学与控制学报, 2021, 19(3): 46—52.
- NIE X B, XIONG Y, PAN Y J. Multi-condition co-simulations of vehicle stability control via fuzzy PID algorithm [J]. Journal of Dynamics and Control, 2021, 19(3): 46—52. (in Chinese)
- [27] IBRAHIM D. Engineering simulation with MATLAB: improving teaching and learning effectiveness [J]. Procedia Computer Science, 2011, 3: 853—858.
- [28] SRIVASTAVA N. Matlab simulation of temperature control of heat exchanger using different controllers [J]. Automation, Control and Intelligent Systems, 2014, 2(1): 1.
- [29] 胡黄水, 赵思远, 刘清雪, 等. 基于动量因子优化学习率的BP神经网络PID参数整定算法[J]. 吉林大学学报(理学版), 2020, 58(6): 1415—1420.
- HU H S, ZHAO S Y, LIU Q X, et al. BP neural network PID parameter tuning algorithm based on momentum factor optimized learning rate [J]. Journal of Jilin University (Science Edition), 2020, 58(6): 1415—1420. (in Chinese)
- [30] SKOCZOWSKI S, DOMEK S, PIETRUSEWICZ K, et al. A method for improving the robustness of PID control [J]. IEEE Transactions on Industrial Electronics, 2005, 52(6): 1669—1676.
- [31] BRENDEN W, RAUBER J, KÜMMERER M, et al. Accurate, reliable and fast robustness evaluation [C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, 2019: 12861—12871.
- [32] GUO J, BAO W, WANG J K, et al. A comprehensive evaluation framework for deep model robustness [J]. Pattern Recognition, 2023, 137: 109308.
- [33] SHAHROKNI A, FELDT R. A systematic review of software robustness [J]. Information and Software Technology, 2013, 55(1): 1—17.
- [34] 侯彦东, 方惠敏, 杨国胜, 等. 一种改进的可变学习速率的BP神经网络算法[J]. 河南大学学报(自然科学版), 2008, 38(3): 309—312.
- HOU Y D, FANG H M, YANG G S, et al. An improvement of variable learning rate BP algorithm [J]. Journal of Henan University (Natural Science), 2008, 38(3): 309—312. (in Chinese)