浙江理工大学学报，2020，43(5)：647-652
Journal of Zhejiang Sci-Tech University
DOI:10.3969/j.issn.1673－3851(n).2020.05.009

基于深度强化学习的自适应增益控制算法

姚　杰,柯鼐挺,任　佳
(浙江理工大学机械与自动控制学院,杭州 310018)

摘　要：经典比例-积分-微分(Proportional-Integral-Derivative，PID)控制器的参数整定过程繁琐,且随着被控对象模型的改变需要重新整定参数,针对该问题提出了一种基于深度强化学习的自适应增益控制算法。该算法在经典PID控制器的比例环节引入深度Q学习网络(Deep Q-network，DQN)模型,对增益进行自适应调整,同时为简化控制器结构,去除了经典PID控制器中的微分环节和积分环节。以双容水箱为研究对象,对该算法进行仿真实验。结果表明:该算法在满足定值控制任务的前提下,相比于经典PID控制器,其超调量及稳态误差更小,且在模型对象改变后能够通过自学习得到相应的控制策略,从而避免了繁琐的参数整定过程。

关键词：深度强化学习;PID;自适应增益;定值控制
中图分类号：TP29　　**文献标志码**：A　　**文章编号**：1673-3851 (2020) 05-0647-06

Adaptive gain control algorithm based on deep reinforcement learning

YAO Jie, *KE Liuting*, *REN Jia*
(Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: The parameter setting process of the classic PID (Proportional-Integral-Derivative) controller is cumbersome, and the parameters need to be set again as the controlled object model changes. Aiming at this problem, we proposed an adaptive gain control algorithm based on deep reinforcement learning. The algorithm introduces deep DQN (Deep Q-network) model in the proportional element of classic PID controller for adaptive to adaptively adjust the gain. Besides, to simplify the structure of the controller, the differentiation element and integration element in the classic PID controller were removed. The simulation experiment was performed on the dual-capacity water tank. The experimental results show that compared with the classic PID controller, the algorithm has smaller overshoot and steady-state error under the precondition of meeting the fixed-value control task, and it can obtain corresponding control strategy through self-learning after the model is changed, thereby avoiding the tedious parameter tuning process.

Key words: deep reinforcement learning; PID; adaptive gain; fixed value control

0　引　言

PID(Proportional-Integral-Derivative，比例-积分-微分)控制器具有鲁棒性高且易操作等优点,被广泛应用于速度和位置控制任务[1]。然而PID控制器的参数整定需要通过观察被控对象的响应逐步调整,过程繁琐,在一定程度上依赖工程师的经验[2]。而且,PID控制器的参数在整定后就不会改变,在被控对象模型改变后需要重新整定,无法满足时变系统的控制要求。

收稿日期:2020－01－09　　网络出版日期:2020－05－08
基金项目:浙江省公益技术研究项目(GG20F030031);浙江省自然科学基金项目(LY17F030024)
作者简介:姚　杰(1996－　),男,浙江温州人,硕士研究生,主要从事深度强化学习方面的研究。
通信作者:任　佳,E-mail:jren@zstu.edu.cn

针对这些问题,众多学者提出了多种自适应PID控制器。在现有的自适应PID控制方法中,模糊PID自适应控制[3-4]具有简单且容易实现的特点。模糊PID自适应控制法首先建立误差、误差变化率与PID控制器参数之间的模糊规则,再通过隶属度函数进行模糊推理,以达到PID参数自适应调整的目的。基于该方法的控制器能够应用于非线性与不确定系统,但是,模糊规则的建立与隶属度函数的选择仍然依赖于人工经验。

为了减少对人工经验的依赖,学者们提出了多种基于进化算法的自适应PID控制器与神经网络自适应PID控制器。Bhatt等[5]采用鲸鱼优化算法(Whale optimizer algorithm, WOA)对PID参数进行整定,并应用于两区互联非再热热力系统负载频率的控制。袁春元等[6]采用蚁群算法(Ant colony optimization, ACO)作为PID参数的整定方法,该方法先利用Ziegler-Nichol法确定参数的搜索范围,再通过蚁群算法的寻优能力对PID参数进行整定,但是该方法无法在线调整PID参数。基于进化算法的自适应PID控制器能达到良好的控制效果,但无法在线调整PID参数。Du等[7]利用径向基神经网络算法(Radial basis function neural network, RBF)的强大学习能力和自适应能力,实现了PID参数的自适应调整。霍召晗等[8]研究的控制器则采用小波神经网络算法(Wavelet neural network, WNN),通过前馈式神经网络与梯度下降纠正误差法,以达到在线实时更新PID参数的目的,相比于经典的PID控制器,该控制器具有更好的动态性能和抗干扰性。

近年来,随着深度强化学习的发展,深度强化学习被广泛应用于交通指挥[9-10]、资源分配[11]、视频游戏[12]、机器人控制[13]等领域。深度强化学习兼具深度学习的特征提取能力和强化学习从采样数据中学习的能力,能够通过试错,在给定的环境中解决任务。由于深度强化学习的优点,有学者尝试在PID参数整定中引入深度强化学习[14-16]。如孙歧峰等[16]提出一种基于异步优势执行器评价器(Asynchronous advantage actor-critic, A3C)的自适应PID控制,达到了在线调整PID参数的目的,但是该算法训练量大,需耗费较多计算机资源。综上所述,已有自适应PID算法大多是通过引入各种智能算法对比例环节、积分环节和微分环节的三个参数进行自动整定或学习。本文针对定值控制问题,为提高控制器的自学习能力,引入了深度强化学习中的深度Q网络(Deep Q-networks, DQN),提出了一种基于DQN的自适应增益控制算法(Adaptive gain control based on DQN, DQN-G)。由于经典PID控制器参数整定繁琐,为简化控制器结构,本文去除了经典PID控制器中的积分环节和微分环节,并以双容水箱为研究对象进行仿真实验。该算法有望应用在汽车悬挂系统、无人机悬停等定值控制问题。

1 增量式PID控制器结构

增量式PID控制器的结构如图1所示,其中:$y'(t)$表示被控量目标值,$y(t)$表示被控量当前值,$e(t)$表示t时刻被控量目标值与被控量当前值之间的差值,$\Delta u(t)$为输入量增量。被控对象的输入量$u(t)$为:

$$u(t)=u(t-1)+\Delta u(t)=u(t-1)+k_{\mathrm{P}}e(t)+k_{\mathrm{I}}(e(t)-e(t-1))+k_{\mathrm{D}}(e(t-1)-e(t-2)) \tag{1}$$

其中:k_{P}、k_{I}、k_{D}分别为PID控制器中比例环节、积分环节和微分环节参数。

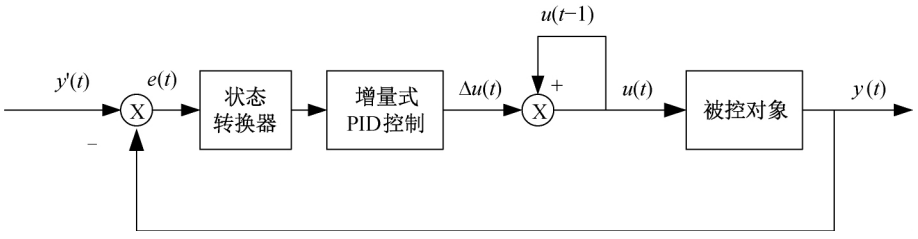

图1 增量PID控制结构

2 基于DQN的自适应增益控制

DQN算法是一种深度强化学习算法,该算法在Q学习和深度学习结合的基础上引入经验回放机制,并且设置了目标网络来单独处理时间差分算法中的TD偏差。DQN算法的网络结构由目标网络和评估网络构成,分别用于逼近目标值函数和评估值函数,更详细的信息可参阅文献[10]。

2.1 DQN-G算法结构

基于DQN的自适应增益控制器的设计思路是

将定值控制过程离散化，从而减小动作空间，由 DQN 根据被控量的目标值与当前值之间的差值来选择增大控制量、减小控制量或控制量不变的操作动作，控制量的改变幅度由闭环负反馈提供。

DQN-G 算法由 DQN 算法与负反馈闭环控制系统构成，DQN-G 控制系统框图如图 2 所示。将被控对象系统的状态信息构成的被控系统状态向量 $s(t)$ 传入评估网络，通过评估网络得到在该被控系统状态下各个动作的价值 $Q(\boldsymbol{s}(t),a_i)$，选择价值最大的动作作为控制动作 $a(t)$，$a(t)$ 与 $e(t)$ 相乘得到被控对象的输入增量 $\Delta u(t)$，则被控对象的输入量如式(2)所示：

$$u(t+1)=u(t)+\Delta u(t) \tag{2}$$

其中的 $\Delta u(t)$ 可进一步表示为：

$$\Delta u(t)=a(t)e(t) \tag{3}$$

当 $u(t+1)$ 作用于被控对象后，被控对象返回新的系统状态向量 $\boldsymbol{s}(t+1)$ 和奖励值 $r(t+1)$，完成一步采样。每完成一步采样后将经验序列 $(\boldsymbol{s}(t),a(t),r(t+1),\boldsymbol{s}(t+1))$ 储存到经验回放池，当经验序列数量达到经验回放池容量上限后，新的经验序列将依次覆盖旧的经验序列。经验回放池中的经验序列用于训练或离线学习。

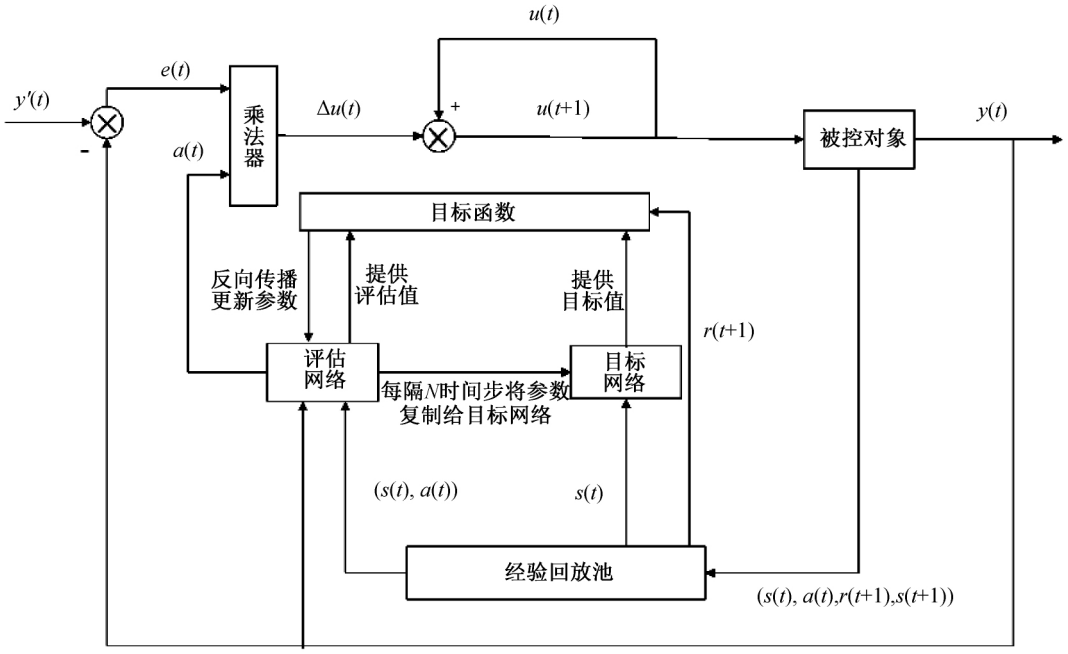

图 2　DQN-G 控制系统框图

在本文中，训练开始时经验回放池为空，DQN 采用随机策略作为行动策略进行探索和收集经验，待经验填满经验池后，算法开始更新迭代，并采用 ε -greedy 策略作为行动策略，其中 $\varepsilon\in[0,1]$，该策略使 DQN 有 $1-\varepsilon$ 的概率随机选择动作，有 ε 的概率选择价值最大的动作。迭代时评估网络从经验回放池随机批量采样，从而打破样本之间的相关性。假设采样到的经验序列为 $(\boldsymbol{s},a,r,\boldsymbol{s}')$，其中状态 $\boldsymbol{s}$ 和动作 a 作为评估网络的输入，通过评估网络逼近评估值函数 $Q(\boldsymbol{s},a\mid\theta_i)$，$\boldsymbol{s}'$ 作为目标网络的输入，通过目标网络逼近目标值函数 $r+\max\limits_{a'}Q(\boldsymbol{s}',a'\mid\theta_{i-1})$，评估值函数与目标值函数构成目标函数，如式(4)所示：

$$L_i(\theta_i)=E_{s,a,r,s'}[(r+\gamma\max_{a'}Q(\boldsymbol{s}',a'\mid\theta_{i-1})-Q(\boldsymbol{s},a\mid\theta_i))^2] \tag{4}$$

其中：γ 为折扣因子，i 为迭代次数。目标函数对 θ_i 求偏导得：

$$\nabla_{\theta_i}L_i(\theta_i)=[(r+\gamma\max_{a'}Q(\boldsymbol{s}',a'\mid\theta_{i-1}))-Q(\boldsymbol{s},a\mid\theta_i)]\nabla_{\theta_i}Q(\boldsymbol{s},a\mid\theta_i)] \tag{5}$$

2.2　DQN-G 网络结构

DQN 算法首先在视频游戏领域中提出，网络结构中存在用于处理视频游戏图像的卷积神经网络。本文无须对图像进行处理，因此将 DQN 中的卷积网络替换成 BP 神经网络，评估网络与目标网络结构相同。评估网络与目标网络的网络结构如图 3 所示，网络结构共有三层。

第 1 层为输入层，输入向量为被控对象当前状态构成的状态向量，可表示为 $\boldsymbol{s}(t)=(x_1(t),x_2(t),\cdots,x_n(t))^{\mathrm{T}}$。

第 2 层为隐藏层，隐藏层的输入为：

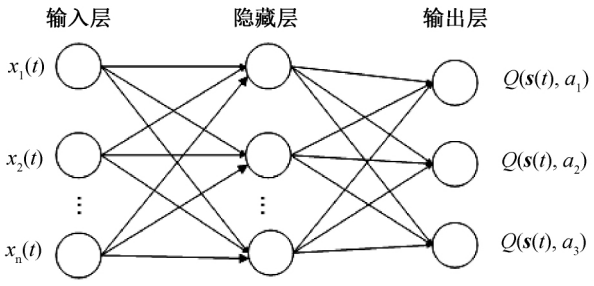

图3 评估网络与目标网络的网络结构

$$h_{\text{in}i}(t)=\boldsymbol{w}_{1i}^{\text{T}}\boldsymbol{s}(t)+b_{1i} \tag{6}$$

其中：i 表示隐藏层第 i 个神经元，$i=1,2,3,\cdots,n$；$\boldsymbol{w}_{1i}^{\text{T}}$ 和 b_{1i} 分别表示输入层对应隐藏层第 i 个神经元的权重系数行向量和截距。隐藏层经过激活函数 ReLu，隐藏层的输出为：

$$h_{\text{out}i}(t)=\max(h_{\text{in}i}(t),0) \tag{7}$$

其中：$h_{\text{out}i}$ 为隐藏层第 i 个神经元的输出。

第3层为输出层，隐藏层的输出为该层的输入，输出为：

$$y_{\text{out}i}(t)=\boldsymbol{w}_{2j}^{\text{T}}\boldsymbol{h}_{\text{out}}(t)+b_{2j} \tag{8}$$

其中：$y_{\text{out}i}$ 为输出层第 i 个神经元的输出，$\boldsymbol{w}_{2j}^{\text{T}}$ 和 b_{2j} 分别表示隐藏层对应输出层第 j 个神经元的权重系数行向量和截距，$j=1,2,3$。式(8)的输出即各个动作对应的值，因此式(8)的结果即为：

$$\boldsymbol{y}_{\text{out}}(t)=(Q(\boldsymbol{s}(t),a_1),Q(\boldsymbol{s}(t),a_2),Q(\boldsymbol{s}(t),a_3)) \tag{9}$$

3 算法测试及讨论

3.1 对象描述及参数设置

为进一步验证算法效果，本文采用 Python 构建双容水箱模型，在该模型上评估 PID、DQN 和 DQN-G 三种算法的表现。水箱对象模型如图4所示，上水箱为水箱1，下水箱为水箱2。

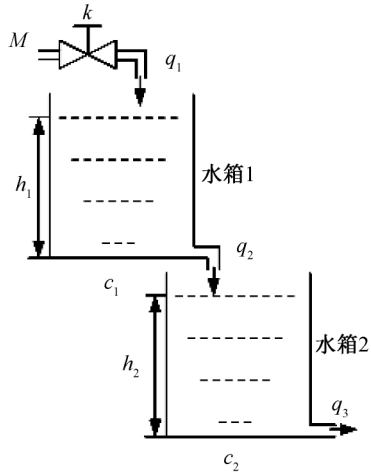

图4 双容水箱液位模型

设 M 为水箱1的单位时间最大流入量，其中 $k(t)$ 为 t 时刻阀门开度，则 t 时刻水箱1的流入量 $q_1(t)$：

$$q_1(t)=k(t)\times M \tag{10}$$

t 时刻水箱1单位时间流出量 $q_2(t)$ 为：

$$q_2(t)=\sqrt{2\times g\times h_1(t)}\times c_3 \tag{11}$$

其中：g 为重力加速度；c_3 为水箱1出口的横截面积；$h_1(t)$ 为水箱1的液位，其表达式为：

$$h_1(t)=h_1(t-1)+\frac{q_1(t-1)-q_2(t-1)}{c_1} \tag{12}$$

其中：c_1 为水箱1横截面积。t 时刻水箱2单位时间流出量 $q_3(t)$ 为：

$$q_3(t)=\sqrt{2\times g\times h_2(t)}\times c_4 \tag{13}$$

其中：c_4 为水箱2出口的横截面积；$h_2(t)$ 为水箱2液位，其表达式为：

$$h_2(t)=h_2(t-1)+\frac{q_2(t-1)-q_3(t-1)}{c_2} \tag{14}$$

其中：c_2 为水箱2横截面积。

本实验中的对象及控制器参数设置如下。水箱对象参数：水箱横截面积 $c_1=c_2=0.5\ \text{m}^2$，水箱出口横截面积 $c_3=c_4=0.03\ \text{m}^2$；控制动作：$a_1=+1$，$a_2=0$，$a_3=-1$；PID 参数采用经验法整定得到：$k_{\text{P}}=2$，$k_{\text{I}}=0.05$，$k_{\text{D}}=0$；DQN 参数设置：隐藏层神经元个数 $n=50$，经验回放池大小 $capacity=2000$，每次经验回放数量 $batch\_size=32$，贪心率 $\varepsilon=0.9$，学习率 $\alpha=0.01$，折扣因子 $\gamma=0.9$，目标网络参数更新频率 $iteration=32$，奖励设置

$$r=\begin{cases}+1, & \text{若}\ |h_2-\text{设定值}|<\text{允许稳态误差范围}\\ -5, & \text{若}\ h_1\ \text{或}\ h_2>\text{限制最大水位}\\ -1, & \text{其他}\end{cases}。$$

3.2 结果分析

本文从三个方面对算法性能进行了对比测试，分别是无扰动情况下的性能对比、抗干扰性能对比和参数发生变化时算法的鲁棒性能对比。

a)无扰动情况下的性能对比。被控对象模型参数相同且不对控制过程进行干扰的情况下，三种算法的性能对比如图5和表1所示。从图5可以看出，PID 算法有振荡趋向目标值的过程，DQN 算法与 DQN-G 算法趋向目标值的过程更加平滑。由表1可知，本文提出的 DQN-G 算法上升时间比 PID 算法多了 1.9785 s，但超调量比 DQN 算法低了 1.07%，比 PID 算法低了 17.37%；稳态误差比 DQN 算法小了 1.35%，比 PID 算法低了 0.34%。

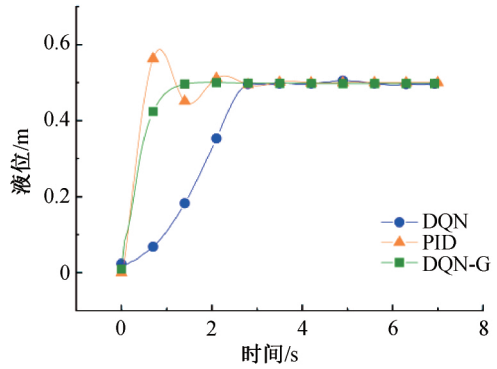

图 5　无扰动情况下输出响应对比曲线

表 1　无扰动情况下性能对比数据

算法	性能指标		
	超调量/%	上升时间/s	稳态误差/%
PID	17.41	0.5719	±0.83
DQN	1.11	5.7810	±1.84
DQN-G	0.04	2.5504	±0.49

b)抗干扰性能比较。被控对象模型参数相同，在系统稳定后加入相同扰动的情况下，三种算法的性能对比如图 6 和表 2 所示。从图 6 可以看出，在系统稳定后加入相同扰动，PID、DQN 和 DQN-G 算法将被控量恢复到目标值的过程曲线与无扰动情况下的曲线基本一致。由表 2 可知，DQN-G 算法的上升时间比 PID 算法多了 1.2895 s，但超调量比 DQN 算法低了 0.25%，比 PID 算法低了 19.59%；稳态误差比 DQN 算法小了 1.01%，比 PID 算法低了 0.48%。

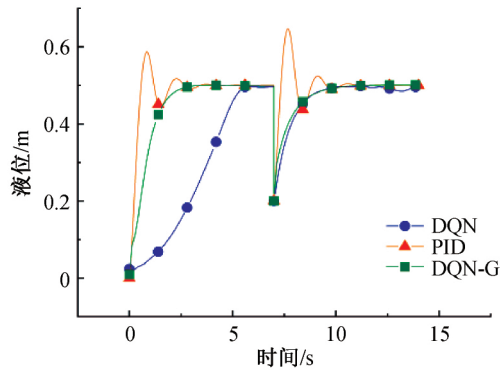

图 6　加入扰动后输出响应对比曲线

表 2　加入扰动后恢复到目标状态的性能对比数据

算法	性能指标		
	超调量/%	上升时间/s	稳态误差/%
PID	19.76	0.3337	±0.83
DQN	0.42	2.8749	±1.36
DQN-G	0.17	1.6232	±0.35

c)鲁棒性能对比。被控对象模型参数改变，不在控制过程中加入扰动的情况下，三种算法性能对比如图 7 和表 3 所示。从图 7 可以看出，控制算法在不重新调参的情况下，由于 PID 算法无法自学习，被控对象模型参数的更改对 PID 算法的控制效果造成了一定的影响，而 DQN 和 DQN-G 算法能够通过自学习得到相应的控制策略。由表 3 可知，被控模型参数更改后，虽然 PID 算法的上升时间仅 0.1872 s，但超调量高达 42.37%。本文提出的 DQN-G 通过自学习后稳态误差性能及超调量性能依然比 PID 算法和 DQN 算法更优。

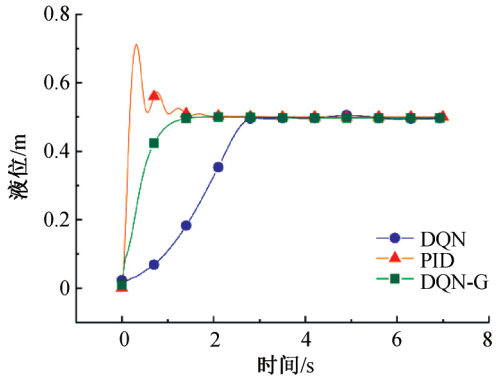

图 7　被控对象模型参数发生改变时算法鲁棒性能对比曲线

表 3　被控对象模型参数发生改变时算法鲁棒性能对比数据

算法	性能指标		
	超调量/%	上升时间/s	稳态误差/%
PID	42.37	0.1827	±0.73
DQN	1.11	4.7616	±1.84
DQN-G	0.04	1.8206	±0.49

综上所述，与经典 DQN 算法比较，本文提出的 DQN-G 算法在 DQN 算法的基础上结合了负反馈机制，当被控对象状态与目标状态之间相差较大时，负反馈机制能够为 DQN 模块提供一个较大的系数，因此相对 DQN 算法具有更短的上升时间；当被控对象状态非常接近目标状态时，负反馈机制为 DQN 模块提供一个非常小的系数，因此 DQN-G 算法具有更低的稳态误差。与经典 PID 算法比较，DQN-G 算法在被控对象状态与目标状态之间相差较大时，由于 DQN 模块在开始时为比例环节提供了一个绝对值小于 1 的系数，使得上升时间比 PID 算法慢；当被控对象状态在目标状态附近时，负反馈机制为 DQN 模块提供一个非常小的系数，使得 DQN-G 算法能够对被控对象输入量进行细微的增量式调节，因此 DQN-G 具有更低的超调量和更低的稳态误差，并且 DQN-G 算法属于无模型的算法，在被控对象模型改变的情况下能够通过学习来适应新的被控对象。

4 结 论

针对经典PID算法参数整定繁琐、超调量过大等问题,本文采用DQN算法对比例环节的增益进行自适应调整,提出了一种基于深度强化学习的自适应增益控制算法,即DQN-G算法。该算法去除了经典PID控制器中的微分环节和积分环节,仅通过自适应调节比例环节的增益达到良好的定值控制效果;被控对象模型改变后无须重新整定参数,能够通过自学习得到相应的比例环节整定策略。仿真实验结果表明,针对定值控制问题,DQN-G算法与经典PID算法相比有明显优势,虽然上升时间比PID算法久,但稳态误差性能和超调量性能优于PID算法,能够作为定值控制任务的有效方法。

参考文献:

[1] Adel T, Abdelkader C. A Particle swarm optimization approach for optimum design of PID controller for nonlinear systems[C]//2013 International Conference on Electrical Engineering and Software Applications. Hammamet. Tunisia: IEEE, 2013: 682-685.

[2] 甄岩, 郝明瑞. 基于深度强化学习的智能PID控制方法研究[J]. 战术导弹技术, 2019(5): 37-43.

[3] Qureshi M S, Swarnkar P, Gupta S. Fuzzy PID sliding mode control for robotics: An application to surgical robot[J]. Recent Advances in Electrical & Electronic Engineering, 2019, 12(2): 118-129.

[4] Osinski C, Leandro G V, Da Costa Oliveira G H. Fuzzy PID controller design for LFC in electric power systems [J]. IEEE Latin America Transactions, 2019, 17(1): 147-154.

[5] Bhatt R, Parmar G, Gupta R. Whale optimized PID controllers for LFC of two area interconnected thermal power plants[J]. ICTACT Journal on Microelectronics, 2018, 3(4): 467-472.

[6] 袁春元, 蔡锦康, 王新彦. 基于粒子群算法的车辆悬架PID控制器研究[J]. 中国农机化学报, 2019, 40(5): 91-97.

[7] Du X J, Wang J L, Jegatheesan V, et al. Dissolved oxygen control in activated sludge process using a neural network-based adaptive PID algorithm [J]. Applied Sciences, 2018, 8(2): 261.

[8] 霍召晗, 许鸣珠. 基于小波神经网络PID的永磁同步电机转速控制[J]. 电机与控制应用, 2019, 46(11): 1-6.

[9] Liu X Y, Ding Z, Borst S, et al. Deep reinforcement learning for intelligent transportation systems[EB/OL]. (2018-12-03) [2020-03-02]. https://arxiv. org/abs/1812. 00979.

[10] Zheng G, Zang X, Xu N, et al. Diagnosing reinforcement learning for traffic signal control[EB/OL]. (2019-05-12)[2020-03-02]. https://arxiv.org/abs/1905. 04716.

[11] 程超, 滕俊杰, 赵艳领, 等. 一种基于多智能体强化学习的流量分配算法[J]. 北京邮电大学学报, 2019, 42(6): 1-7.

[12] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.

[13] Carlucho I, De Paula M, Acosta G G. Double Q-PID algorithm for mobile robot control[J]. Expert Systems with Applications, 2019, 137: 292-307.

[14] 刘俊杰, 郝明瑞, 孙明玮, 等. 基于强化学习的飞航导弹姿态控制PID参数调节方法[J]. 战术导弹技术, 2019(5): 58-63.

[15] 张佳慧. 基于Actor-Critic学习的自适应PID控制策略研究[D]. 秦皇岛: 燕山大学, 2018: 19-30.

[16] 孙歧峰, 任辉, 段友祥. 基于异步优势执行器评价器学习的自适应PID控制设计[J]. 信息与控制, 2019, 48(3): 323-328.

(责任编辑:康 锋)