文章编号:1671-4598(2019)02-0070-04 DOI:10.16526/j.cnki.11—4762/tp.2019.02.016 中图分类号:TP27 文献标识码:A

基于异步优势执行器评价器的自适应 PID 控制

段友祥，任 晖，孙岐峰，闫亚男
（中国石油大学（华东）计算机与通信工程学院，山东 青岛 266580）

摘要：自适应 PID 较好地解决了传统 PID 无法自整定参数的问题，已成为控制领域内的研究热点；研究基于异步优势执行器评价器（Asynchronous Advantage Actor—Critic，A3C）算法设计了一种新的自适应 PID 控制器；该控制器利用 A3C 结构的多线程异步学习特性，并行训练多个执行器评价器（Actor—Critic，AC）结构的智能体，每个智能体采用多层前馈神经网络逼近策略函数和值函数实现在连续动作空间中搜索最优的参数整定策略，以达到最佳的控制效果；与已有的多种自适应 PID 控制器性能对比分析结果表明该方法具有收敛速度快，自适应能力强的特点。

关键词：深度强化学习；异步优势执行器评价器；自适应 PID

Adaptive PID Controller Based on Asynchronous Advantage Actor—Critic Learning

Duan Youxiang, Ren Hui, Sun Qifeng, Yan Yanan
(College of Computer & Communication Engineering, China University of Petroleum, Qingdao 266580, China)

Abstract: Self—adaptive PID has become a hotspot in the field of control, it can solve the problem that traditional PID can't turning parameters. This paper proposed a new adaptive PID controller based on the Asynchronous Advantage Actor—Critic (A3C) algorithm. It used the multi—threaded and asynchronous learning style to train multiple agents of Actor—Critic (AC) structures in parallel. In order to achieve the best effect, each agent adopts a multilayer feedforward neural network to approximate strategy function and value function. In this way, they can search for the best parameter turning strategies in continuous motion space. Compared with the performance of others adaptive PID controllers, the results show that this method has the advantage of fast convergence and strong self—adaptability.

Keywords: deep reinforcement learning; asynchronous advantage actor—critic; adaptive PID control

0 引言

PID 控制具有鲁棒性高且易于操作等优点，是现代工业中广泛应用的一种控制方法[1]。然而，传统 PID 的参数一旦确定就无法在线调整，难以满足时变系统的控制要求。自适应 PID 在传统 PID 的基础上引入了在线调参的思想，使其能够根据系统状态的变化调整 PID 参数，提高了系统的响应速度。目前应用比较广泛的自适应 PID 控制器有：模糊自适应 PID 控制器[2]，它以误差和误差变化率作为输入，通过查询模糊矩阵表进行参数调整，从而满足 PID 参数自整定的要求，但这种设计方法需要较多的先验知识，存在大量的参数优化问题[3]；基于神经网络的自适应 PID 控制[4]，利用神经网络对于非线性结构的良好逼近能力，无需辨识复杂的非线性被控对象就能够达到有效的控制，但是获取监督学习中的教师信号仍然存在困难[5]；进化算法自适应 PID 控制器[6]，虽然对于先验知识要求较少，但是在实际工程中难以实现实时控制；强化学习自适应 PID 控制器[7]，利用强化学习非监督特性解决了教师信号难以获取的问题，这类控制器需要较少的先验知识而且控制过程无需复杂的参数优化。其中执行器－评价器（Actor—Critic，AC）自适应 PID[8]是应用最为广泛的强化学习控制器，该控制器提出了一种结合 AC 结构实现在线调整参数并采用神经网络逼近马氏决策过程中的值函数和决策函数的设计思路，但由于 AC 算法中前后学习数据的相互关联性，影响了控制器的收敛速度[9]。

Google 的 DeepMind 团队提出的异步优势执行器评价器（Asynchronous Advantage Actor—Critic，A3C）学习算法[10]利用 CPU 多线程并行的特性，在 CPU 多线程上异步地训练多个智能体（agent），并行中的 agent 会经历不同的学习状态，从而打破了学习样本的相关性[11]，这种高效的异步结构执行方式已经应用到多个领域[12]。本文结合 A3C 结构多线程异步训练的方式以及强化学习的无模型在线学习能力，使用 BP 神经网络作为函数逼近器，最终研究提出了一种基于异步优势执行器评价器的自适应 PID 控制器设计方法，并在仿真实验中验证了该方法的优越性和有效性。

1 系统结构及原理

1.1 增量式 PID 控制原理

数字 PID 控制可分为两类，其中包括位置式 PID 与增

收稿日期:2018-07-23； **修回日期**:2018-08-18。
基金项目:“十三五”重大专项（2017ZX05009—001，2016ZX05011—002）;中央高校基本科研业务费(18CX02020A)。
作者简介:段友祥(1964-) 男,山东省东营市人,博士 教授,主要从事智能控制与人工智能方向的研究。

量式 PID[13]。增量式 PID 是一种通过对控制量的增量进行 PID 控制的算法，其计算公式见式（1）：

$$u(t)=u(t-1)+\Delta u(t)=u(t-1)+K_i(t)e(t)+K_p(t)\Delta e(t)+K_d(t)\Delta^2 e(t) \tag{1}$$

其中：

$$e(t)=y'(t)-y(t),$$
$$\Delta e(t)=e(t)-e(t-1),$$
$$\Delta^2 e(t)=e(t)-2*e(t-1)+e(t-2)$$

$y'(t)$ 表示当前的实际信号值，$y(t)$ 表示当前系统的输出值，$e(t)$ 表示当前误差，$\Delta e(t)$ 为一次误差，$\Delta^2 e(t)$ 为二次误差。k_P 为比例系数，决定了控制程度的强弱，比例系数越大，系统的响应速度就越快，但是容易使系统发生震荡和超调。k_I 为积分系数可以消除系统的静态误差。k_D 为微分系数有助于减少系统超调量，提高系统的控制精度。不同的 PID 参数造成了控制系统的差异，控制过程中，根据系统的动态特性从而调整 PID 参数，往往会得到满意的控制效果。

在控制结构上，增量式 PID 控制较位置式 PID 控制取消了 PID 控制中积分环节的累计求和，节省了大量的计算性能和储存空间，为 A3C 算法的学习速率和学习样本的存储提供了保障。此外，增量式 PID 的每一次的输出为控制量的增量，在系统发生故障时对系统的影响程度较小，使得环境奖励更加稳定，保证了算法学习的收敛速度。

1.2　A3C 学习算法结构

A3C 算法是一种深度强化学习算法，该算法在 Actor－Critic 框架基础上引入了异步训练的思想，在提升控制性能的同时大大加快了训练速度。A3C 学习框架由一个中央网络（Global Net）和多个 Actor－Critic 结构以及仿真环境组成，如图 1 所示。

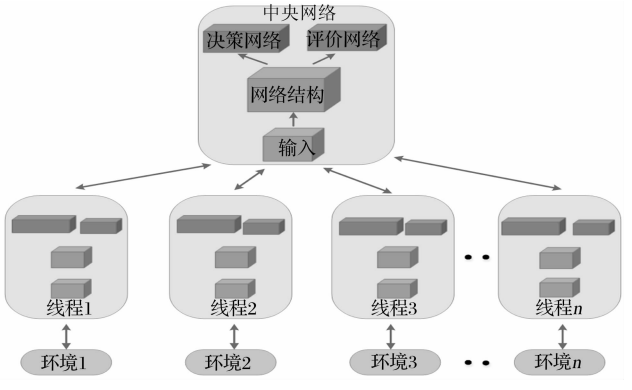

图 1　A3C 算法结构图

由图 1 可见，A3C 算法创建了多个 agent，每个 agent 即为一个 AC 结构，包括一个决策网络和评价网络，算法将 agent 放置在相同的环境实例中并行执行和学习，提高了每个 agent 的学习速率。此外，A3C 采用了中央网络的学习机制，打破了 agent 学习样本的相关性，其主要作用为更新和存储 AC 结构中决策网络和评价网络的参数，不同 agent 将自身的学习数据传递给中央网络用以更新自身参数，从而提高了收敛速率。其中，决策网络即 Actor 网络，负责学习最优策略，使得 agent 可以针对不同环境状态选择最优的决策，而评价网络即 Critic 网络，负责拟合价值函数，增强了 agent 对于环境的奖励感知能力。决策网络和评价网络的组合使用保证了学习算法的有效性和鲁棒性。

2　A3C－PID 控制器设计

2.1　A3C－PID 控制器结构

基于 A3C 学习的自适应 PID 控制器的设计思路就是在增量式 PID 控制器的基础上结合了 A3C 异步学习结构，其结构设计如图 2 所示。

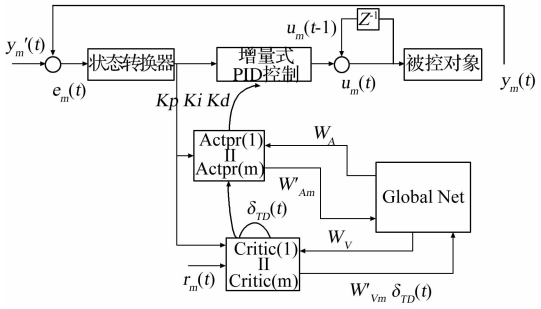

图 2　A3C－PID 控制结构图

在图 2 中，对于每一个 agent，初始化误差 $e_m(t)$ 经状态转换器计算出 $\Delta e_m(t)\Delta^2 e_m(t)$ 以生成状态向量 $S_m(t)=[e_m(t),\Delta e_m(t),\Delta^2 e_m(t)]^T$ 用于表示不同时刻的系统状态特征。Actor（m）表示 m 个并行训练的 agent 中的 Actor 网络，其通过神经网络将状态向量 $S_m(t)$ 映射成 PID 控制器的 k_P、k_I、k_D 三个参数值，新的参数作用于增量式 PID 控制器从而产生控制量增量，系统计算下一时刻的误差 $e_m(t+1)$ 并根据式（2）计算奖励值 $r_m(t)$，以此完成一步样本采样。待系统完成 n 步采样后，Critic（m）输出状态的估计值 $V(S_{t+n},W'_v)$ 并计算 n 步 TD 误差值 δ_{TD}，其中 $V(S_{t+n},W'_v)$ 和 δ_{TD} 都是作为评判 Actor 网络在 t 时刻决策优劣程度的重要依据。此后，Global Net 将 Actor（m）网络参数 W'_{am}、Critic（m）网络参数 W'_{vm} 以及 δ_{TD} 作为学习样本并按照策略梯度和梯度下降的方式分别更新自身的 W_a、W_v 参数。参数更新后，Global Net 将 W_a 和 W_v 传递给 Actor（m）和 Critic（m）从而实现了 AC 网络的异步更新。

$$r_m(t)=\alpha_1 r_1(t)+\alpha_2 r_2(t) \tag{2}$$
$$r_1(t)=\begin{cases}0, & |e_m(t)|<\varepsilon\\ \varepsilon-e_m(t), & \text{其他}\end{cases}$$
$$r_2(t)=\begin{cases}0, & |e_m(t)|\leqslant|e_m(t-1)|\\ |e_m(t)|-|e_m(t-1)|, & \text{其他}\end{cases}$$

2.2　基于多层前馈神经网络的 A3C－PID 学习

多层前馈神经网络[14]又称 BP 神经网，是一种多层前向网络的反向传播算法，具有较强的非线性映射能力，适合于求解内部机制复杂的问题。因此，本文使用两个 BP 神经网络分别实现 Actor 策略函数和 Critic 值函数的学习，其网络结构如图 3、图 4 所示。

如图 3 所示，Actor 网络共有 3 层：

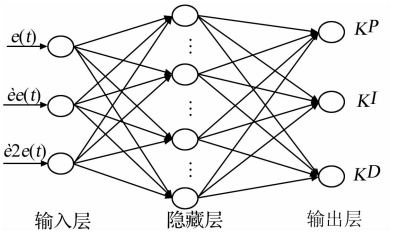

图 3 Actor 网络结构图

第 1 层为输入层共有三个结点，输入向量 $S=[e_m(t),\Delta e_m(t),\Delta^2 e_m(t)]^T$ 代表状态向量。

第 2 层为隐藏层设有 20 个结点，隐藏层与输入层之间没有设置激活函数，其隐藏层的输入为对输入层直接加权求和，如公式（3）：

$$hi_k(t)=\sum_{i=1}^{n} w_{ik}x_i(t)-b_k \quad k=1,2,3\cdots 20 \tag{3}$$

其中：k 表示隐藏层神经元的个数。隐藏层的输出使用了 Relu6 激活函数，其输出公式见式（4）：

$$ho_k(t)=\min(\max(hi_k(t),0),6) \quad k=1,2,3\cdots 20 \tag{4}$$

第 3 层为输出层设有三个结点，输出层的输入直接对隐藏层的输出进行加权求和，输出公式如式（5）所示：

$$yi_o(t)=\sum_{j=1}^{k} w_{ho}ho_j-b_o \quad o=1,2,3 \tag{5}$$

输出层的输出使用 softplus 激活函数，其输出公式如式（6）：

$$yo_o(t)=\log(1+e^{yi_o(t)}) \quad o=1,2,3 \tag{6}$$

Actor 网络并不是直接输出 k_P、k_I、k_D 值而是输出这三个参数的均值和方差，最终通过高斯分布估计出 k_P、k_I、k_D 的实际值。Critic 网络结构同样采用三层 BP 神经网表示，结构如图 4 所示，前两层的结构与 Actor 网络前两层结构相同，不同在于 Critic 网络的输出层只有一个结点，输出了不同时刻状态的值函数 $V(S_t,W'_v)$。

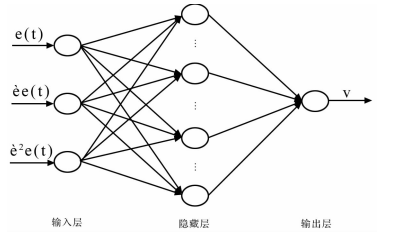

图 4 Critic 网络结构

强化学习中包括蒙特卡洛学习和 TD 学习方法，其二者区别在于蒙特卡洛方法需要统计完成一次控制的每一个采样点的状态值，并且在此之后再对起始状态的决策进行评价，具有较长的学习周期，不适用于实时控制。然而 TD 学习方法可以在 n 步采样后即可评价起始状态的决策，从而完成一次模型学习，二者收敛性能相近，但 TD 学习算法具有实时性，更加适用于实时控制。基于此，在 A3C－PID 结构中，Actor 与 Critic 网络均采用 n 步 TD 误差的方法[15]来学习动作概率函数和值函数。在 n 步 TD 学习算法中，TD 误差 δ_{TD} 的计算由起始状态 S_t 的状态估计值 $V(S_{t+n},W'_v)$ 与 n 步后样本的估计值 q_t 的差分实现，如公式（7）：

$$\delta_{TD}=q_t-V(S_t,W'_v) \tag{7}$$

$$q_t=r_{t+1}+\gamma r_{t+2}+\cdots+\gamma^{n-1}r_{t+n}+\gamma^n V(S_{t+n},W'_v)$$

其中：$0<\gamma<1$，表示折扣因子，用来确定延迟回报与立即回报的比例，W'_v 为 Critic 网络权值。误差 δ_{TD} 反应了 Actor 网络所选动作的优劣程度，系统学习的性能指标为公式（8）。

$$E(t)=\frac{1}{2}\delta_{TD}^2(t) \tag{8}$$

在计算出 TD 误差后，为打破学习样本的关联性，A3C 结构中的每个 Actor－Critic 网络并不会直接更新自身的网络权值，而是用自身的梯度去更新中央网络存储的 Actor－Critic 网络参数，更新公式见公式（9）、公式（10）：

$$W_a=W_a+\alpha_a(dW_a+\nabla_{w'a}\log\pi(a\mid s;W'_a)\delta_{TD}) \tag{9}$$

$$W_v=W_v+\alpha_c(dW_v+\frac{\partial\,\delta_{TD}^2}{W'_v}) \tag{10}$$

其中：W_a 为中央网络存储的 Actor 网络权值，W'_a 表示每个 AC 结构的 Actor 网络的权值，W_v 为中央网络存储的 Critic 网络权值，W'_v 表示每个 AC 结构的 Critic 网络权值，α_a 为 Actor 的学习率，α_c 为 Critic 的学习率。

2.3 A3C－PID 控制器网络初始化

网络的初始参数直接影响了闭环控制系统的稳定性，神经网络 PID 控制由于教师信号难以获取，需要按照经验或人工试凑确定网络参数。强化学习的非监督学习特性使得控制器通过 K 次迭代学习便可获取最优的网络初始参数。然而，AC－PID 控制器由于 AC 算法获取的学习样本具有前后关联性，导致了较慢的收敛速度。相比之下，A3C－PID 在 CPU 的多线程中异步学习网络参数，破坏了样本关联性，提高了收敛速率。A3C－PID 网络参数学习过程与 2.1 节中叙述相似，但不同在于 A3C－PID 在迭代学习时设置 m 值为计算机 CPU 核心线程数，而当 A3C－PID 在线控制时 m 值设置为 1。

2.4 A3C－PID 控制器设计流程

基于 A3C 并行学习的体系结构和以 n 步 TD 误差为性能指标的网络学习方式，归纳出 A3C－PID 控制器的设计流程如下：

a）设置采样周期 ts，A3C 算法的线程个数 m，更新周期 n，通过 K 次迭代学习，初始化每个 AC 结构的网络参数；

b）计算系统误差 $e_m(t)$，构造出系统状态向量 $S_m(t)$，作为 Actor（m）和 Critic（m）的输入；

c）Critic（m）输出 $V(S_t,W'_v)$；

d）Actor（m）输出 k_P、k_I、k_D 值，根据式（1）计算系统输出 $u_m(t)$，并观测下一采样时间系统误差 $e_m(t+1)$，根据式（2）计算奖励值函数 $r_m(t)$；

e）判断是否更新 Actor，Critic 参数，若达到更新条件，Critic 输出状态估计值 $V(S_{t+n},W'_v)$ 根据式（9）和式（10）更新 Global Net 参数 W_a 和 W_c，否则更新 $S_m(t)$ 并返

回步骤 d)；

f) Global Net 传递 Actor（m）和 Critic（m）新的参数值 W'_{am} 和 W'_{cm}；

g) 判断是否满足控制结束条件，若满足结束条件，退出控制，否则更新 $S_m(t)$ 并返回步骤 c)。

3　仿真实验

阶跃响应能够很大程度上反应系统的动态特性，是分析系统性能的重要手段。因此，为测试控制算法能够在系统特性动态改变的同时自适应地调整 PID 参数，进行了阶跃信号控制实验。被控制对象选为：

$$y(t) = 1.2(1-0.8e^{-0.1\cdot k})y(t-1)+u(t-1) \quad (11)$$

被控对象的初始状态取 [0，0]，设采样时间为 1 ms，A3C 学习自适应 PID 控制的各个参数为：

$$m = 4, \alpha_a = 0.001, \alpha_c = 0.01, \varepsilon = 0.001$$

$$\gamma = 0.9, n = 30, K = 3000$$

仿真结果见图 5～8 及表 1。

表 1　控制器性能对比

控制器	超调量/%	上升时间/ms	稳态误差	调节时间/ms
A3C－PID	0.157 1	18	0	33
AC－PID	0.102 1	21	0	48
BP－PID	2.170 5	12	0	32

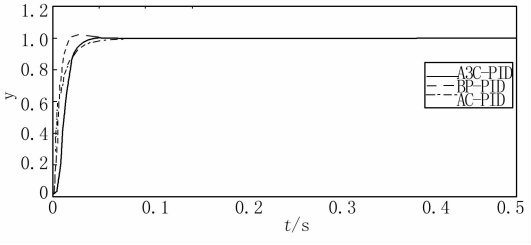

图 5　位置跟踪

图 5 为 A3C，BP，AC 自适应 PID 控制器对于参考模型的位置跟踪结果；表 1 为 A3C，BP，AC 自适应 PID 控制器性能对比。从表 1 可以看出，三个控制器都有着较好的控制精度，即稳态误差都为 0。在动态性能方面，见图 5，在仿真初期（大约 20 个仿真周期内），BP－PID 控制器有着更快的响应速度，上升时间更短，为 12 ms，但是 BP－PID 具有 2.1705% 的较高系统超调量。相反 AC－PID 和 A3C－PID 都具有较小的 0.1571% 和 0.1021% 的系统超调量，但是 AC－PID 的调节时间较长，为 48 ms，上升时间 21 ms。相比之下，A3C－PID 控制器有着更好的控制稳定性和快速性。

图 6 和图 7 分别为 A3C－PID 的跟踪误差以及 PID 控制器的参数自适应变换的过程。由图 6～图 7 可以看出，A3C－PID 控制器能够根据不同周期内的误差自适应调整 PID 参数值。在仿真开始阶段，由于系统跟踪误差较大，为保证系统有较快的响应速度，k_P 不断增大，k_D 逐渐减小，同时为避免系统出现较高的超调量，限制了 k_I 的增加；随

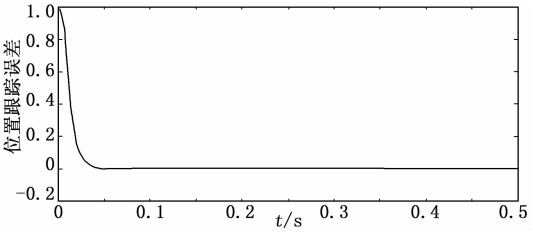

图 6　位置跟踪误差

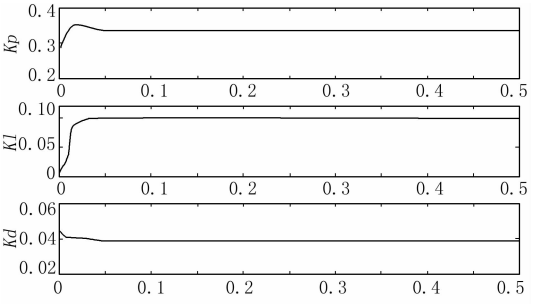

图 7　控制器参数整定结果

着误差不断减小，k_P 开始减小，为消除累计误差 k_I 值逐渐增加，但同时造成了少量的超调，由于此阶段 k_D 值于系统影响较大，所以趋于稳定；最终跟踪误差为 0，k_P、k_I、k_D 值达到稳定状态。仿真结果可以看出，A3C－PID 控制器有着良好的自适应能力。

强化学习的目标是学习最优策略从而最大化由起始状态到终止状态的折扣回报值 U，计算公式见式（12）：

$$U = E[\sum_{t=0}^{end}\gamma R(S_t)] \quad (12)$$

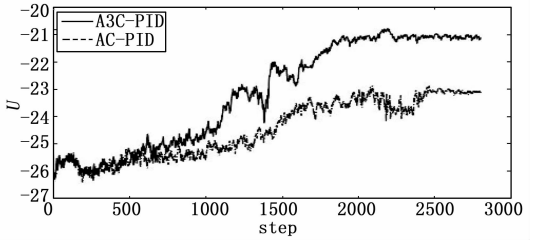

图 8　强化学习折扣回报值曲线

图 8 为 AC－PID 与 A3C－PID 折扣回报值曲线。从图 8 可以看出，在进行 3000 次迭代训练后，A3C－PID 相比 AC－PID 获得了更高的回报值。除此之外，A3C－PID 在约 1800 次迭代训练后渐渐趋于稳定状态，而 AC－PID 在 2500 次迭代后才出现收敛的趋势。由此可得，A3C－PID 相比 AC－PID 有着更快的收敛速度。

4　结束语

本文详细分析了多种自适应 PID 控制器，并在增量式 PID 控制器的基础上引入了 A3C 的异步学习体系。结合该体系，把多个 AC 结构置于相同环境中并行进行学习，为使控制器搜索到最优整定策略，使用 BP 神经网逼近每个 AC

（下转第 78 页）