doi:10.3969/j.issn.1005-3026.2015.03.004

模型未知非零和博弈问题的策略迭代算法

杨 明[1], 罗艳红[1], 王义贺[2]
(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 国网辽宁省电力有限公司 经济技术研究院, 辽宁 沈阳 110000)

摘 要: 提出了一种在线积分策略迭代算法,用来求解内部非线性动力模型未知的双人非零和博弈问题.通过在控制策略和干扰策略中引入探测信号,从而避开了系统的模型信息,得到了一个求解非零和博弈的无模型的近似动态规划算法.该算法同步更新值函数、控制策略、扰动策略,并且最终得到收敛的策略权值.在算法实现过程中,使用4个神经网络分别近似两个值函数、控制策略和扰动策略,使用最小二乘法估计神经网络的未知参数.最后仿真结果验证了算法的有效性.

关 键 词: 自适应动态规划;非零和博弈;策略迭代;神经网络;最优控制
中图分类号: TP 183 **文献标志码**: A **文章编号**: 1005-3026(2015)03-0318-05

Policy Iteration Algorithm for Nonzero-Sum Games with Unknown Models

YANG Ming[1], *LUO Yan-hong*[1], *WANG Yi-he*[2]
(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. Economic Technology Institute, Nation State Liaoning Province Power Co., Ltd., Shenyang 110000, China. Corresponding author: YANG Ming, E-mail: yangming20060916@126.com)

Abstract: An online integral policy iteration algorithm was proposed to find the solution of two-player nonzero-sum differential games with completely unknown nonlinear continuous-time dynamics. Exploration signals can be added into the control and disturbance policies, rather than having to find the model information. An approximate dynamic programming (ADP) of model-free approach can be constructed, and the nonzero-sum games can be solved. The value function, control and disturbance policies simultaneously can be updated by the proposed algorithm, and converged policy weight parameters are obtained. To implement the algorithm, four neural networks are used respectively to approximate the two game value functions, the control policy and the disturbance policy. The least squares method is used to estimate the unknown parameters of the neural networks. The effectiveness of the developed scheme is demonstrated by a simulation example.

Key words: adaptive dynamic programming; nonzero-sum games; policy iteration; neural networks; optimal control

博弈论考虑游戏中的个体的预测行为和实际行为,并研究它们的优化策略.零和博弈表示所有博弈方的利益之和为零或一个常数,即一方有所得,其他方必有所失.非零和博弈表示在不同策略组合自适应动态规划(adaptive dynamic programming,ADP)理论融合了动态规划、增强学习以及函数近似等方法,在处理非线性系统的最优控制问题中有着独特优势[1-2].ADP方法已经应用到求解零和博弈问题和非零和博弈中[3-10],根据是否需要内部系统模型信息,算法可分为无模型算法和有模型算法.文献[3]提出一种基于策略迭代增强学习技术的自适应控制算法解决非零和博弈问题;文献[7]采用积分RL(reinforcement learning)方法在线求解线性连续

收稿日期: 2014-01-08
基金项目: 国家自然科学基金资助项目(61104010);高等学校博士学科点专项科研基金资助项目(20110042120032).
作者简介: 杨 明(1987-),男,山东临沂人,东北大学博士研究生.

系统的非零和博弈纳什均衡解；文献[8]采用基于单网络的 ADP 方法来求解非线性连续系统的非零和博弈平衡点. 以上文献提出的算法均需要知道系统模型信息. 文献[9]提出了一种无模型策略迭代算法求解线性二次最优问题；文献[10]提出了一种无模型策略迭代算法求解零和博弈问题，其中值函数、控制策略和干扰策略同时更新. 目前还没有对内部系统模型未知的非零和博弈问题的求解.

本文提出一种在线积分策略迭代算法，该算法不需要知道系统的模型信息，利用输入输出数据求解非零和博弈问题的纳什平衡解.

1　问题描述

考虑一类非线性连续时间动态系统：

$$\dot{x} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}. \tag{1}$$

其中：$\boldsymbol{x} \in \mathbf{R}^n$ 是系统状态；$\boldsymbol{u} \in \mathbf{R}^m$ 是控制输入；$\boldsymbol{w} \in \mathbf{R}^q$ 是外部扰动输入；$\boldsymbol{f}(\boldsymbol{x}) \in \mathbf{R}^n$；$\boldsymbol{g}(\boldsymbol{x}) \in \mathbf{R}^{n\times m}$ 和 $\boldsymbol{k}(\boldsymbol{x}) \in \mathbf{R}^{n\times q}$ 是内部系统模型. 假设 $\boldsymbol{f}(\boldsymbol{x}) \in \mathbf{R}^n$，$\boldsymbol{g}(\boldsymbol{x}) \in \mathbf{R}^{n\times m}$ 和 $\boldsymbol{k}(\boldsymbol{x}) \in \mathbf{R}^{n\times q}$ 是未知的，$\boldsymbol{x} = 0$ 是系统(1)的一个平衡点.

仿照文献[8]定义两个性能指标：

$$J_1(\boldsymbol{x}_0,\boldsymbol{u},\boldsymbol{w}) = \int_0^\infty (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_1\boldsymbol{x} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{11}\boldsymbol{u} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{12}\boldsymbol{w})\mathrm{d}\tau \triangleq \int_0^\infty r_1(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})\mathrm{d}\tau; \tag{2}$$

$$J_2(\boldsymbol{x}_0,\boldsymbol{u},\boldsymbol{w}) = \int_0^\infty (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_2\boldsymbol{x} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{21}\boldsymbol{u} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{w})\mathrm{d}\tau \triangleq \int_0^\infty r_2(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w})\mathrm{d}\tau. \tag{3}$$

其中：$\boldsymbol{Q}_1 \geqslant 0$；$\boldsymbol{Q}_2 \geqslant 0$；$\boldsymbol{R}_{11} > 0$；$\boldsymbol{R}_{12} \geqslant 0$；$\boldsymbol{R}_{21} \geqslant 0$；$\boldsymbol{R}_{22} > 0$.

对于控制策略 $\boldsymbol{u}$ 和干扰策略 $\boldsymbol{w}$，定义值函数：

$$V_1(\boldsymbol{x}_t,\boldsymbol{u},\boldsymbol{w}) = \int_t^\infty (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_1\boldsymbol{x} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{11}\boldsymbol{u} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{12}\boldsymbol{w})\mathrm{d}\tau; \tag{4}$$

$$V_2(\boldsymbol{x}_t,\boldsymbol{u},\boldsymbol{w}) = \int_t^\infty (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}_2\boldsymbol{x} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{21}\boldsymbol{u} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{w})\mathrm{d}\tau. \tag{5}$$

定义非零和博弈问题为

$$V_1^*(\boldsymbol{x}_0) = \min_{\boldsymbol{u}}\max_{\boldsymbol{d}} V_1(\boldsymbol{x}_0,\boldsymbol{u},\boldsymbol{w}), \tag{6}$$

$$V_2^*(\boldsymbol{x}_0) = \min_{\boldsymbol{d}}\max_{\boldsymbol{u}} V_2(\boldsymbol{x}_0,\boldsymbol{u},\boldsymbol{w}). \tag{7}$$

问题目标是找到满足以下不等式的纳什平衡策略$(\boldsymbol{u}^*,\boldsymbol{w}^*)$：

$$\left.\begin{aligned} V_1(\boldsymbol{x}_t,\boldsymbol{u}^*,\boldsymbol{w}^*) \leqslant V_1(\boldsymbol{x}_t,\boldsymbol{u},\boldsymbol{w}^*),\\ V_2(\boldsymbol{x}_t,\boldsymbol{u}^*,\boldsymbol{w}^*) \leqslant V_2(\boldsymbol{x}_t,\boldsymbol{u}^*,\boldsymbol{w}). \end{aligned}\right\} \tag{8}$$

对值函数微分得到非线性 Lyapunov 方程：

$$0 = r_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}) + (\nabla V_i)^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w})\ (i = 1,2). \tag{9}$$

其中 $\nabla V_i = \dfrac{\partial V_i}{\partial \boldsymbol{x}}$.

定义 Hamilton 函数：

$$H_i(\boldsymbol{x},\nabla V_i,\boldsymbol{u},\boldsymbol{w}) = r_i(\boldsymbol{x},\boldsymbol{u},\boldsymbol{w}) + (\nabla V_i)^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w})\ (i = 1,2).$$

由 $\dfrac{\partial H_i}{\partial \boldsymbol{u}} = 0$，$\dfrac{\partial H_i}{\partial \boldsymbol{w}} = 0$ 得到

$$\left.\begin{aligned} \boldsymbol{u} &= -\frac{1}{2}\boldsymbol{R}_{11}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\nabla V_1,\\ \boldsymbol{w} &= -\frac{1}{2}\boldsymbol{R}_{22}^{-1}\boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\nabla V_2. \end{aligned}\right\} \tag{10}$$

将式(10)代入式(9)得到两个耦合的 HJ 方程：

$$\boldsymbol{Q}_1(\boldsymbol{x}) - \frac{1}{4}(\nabla V_1)^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}_{11}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\nabla V_1 + (\nabla V_1)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) + \frac{1}{4}(\nabla V_2)^{\mathrm{T}}\boldsymbol{k}(\boldsymbol{x})\boldsymbol{R}_{22}^{-1}\boldsymbol{R}_{12}\boldsymbol{R}_{22}^{-1}\boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\nabla V_2 - \frac{1}{2}(\nabla V_1)^{\mathrm{T}}\boldsymbol{k}(\boldsymbol{x})\boldsymbol{R}_{22}^{-1}\boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\nabla V_2 = 0,$$

$$Q_2(x) - \frac{1}{4}(\nabla V_2)^{\mathrm{T}}\boldsymbol{k}(\boldsymbol{x})\boldsymbol{R}_{22}^{-1}\boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\nabla V_2 + (\nabla V_2)^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) + \frac{1}{4}(\nabla V_1)^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{21}\boldsymbol{R}_{11}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\nabla V_1 - \frac{1}{2}(\nabla V_2)^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}_{11}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\nabla V_1 = 0.$$

这是两个非线性偏微分方程，很难得到解析解.

2　无模型积分策略迭代算法

为了分析无模型积分策略迭代(policy iteration，PI)算法，首先给出一个已知模型非零和博弈的 PI 算法.

算法 1 的步骤如下：

步骤 1　给定初始稳定控制策略 $\boldsymbol{u}_1$ 和干扰策略 $\boldsymbol{w}_1$，设定 $i = 1$.

步骤 2(策略评价)　根据已知 $\boldsymbol{u}_i$ 和 $\boldsymbol{w}_i$，通过下面的李雅普诺夫方程求解 V_1^i 和 V_2^i：

$$\left.\begin{aligned} &0 = r_1(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{d}_i) + (\nabla V_1^i)^{\mathrm{T}}(\boldsymbol{x})(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}_i(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}_i(\boldsymbol{x}))\ ,\\ &0 = r_2(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{d}_i) + (\nabla V_2^i)^{\mathrm{T}}(\boldsymbol{x})(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}_i(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}_i(\boldsymbol{x}))\ . \end{aligned}\right\} \tag{11}$$

步骤 3(策略提高)　更新控制和干扰策略：

$$\left.\begin{aligned}\boldsymbol{u}_{i+1}(\boldsymbol{x}) &= -\frac{1}{2}\boldsymbol{R}_{11}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\nabla V_1^i(x)\ ,\\ \boldsymbol{w}_{i+1}(\boldsymbol{x}) &= -\frac{1}{2}\boldsymbol{R}_{22}^{-1}\boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\nabla V_2^i(\boldsymbol{x})\ .\end{aligned}\right\}\tag{12}$$

步骤 4 如果 $\| V_j^i - V_j^{i-1} \| \leqslant \varepsilon (j=1,2,\varepsilon$ 是预设的正实数),停止算法并输出 V_1^i, V_2^i;否则,设 $i=i+1$,并转到步骤 2.

可以证明[11] 当 $i\to\infty$ 时,算法 1 得到的 V_j^i $(j=1,2)$,$\boldsymbol{u}_i, \boldsymbol{w}_i$ 分别收敛到 HJ 方程的最优解 V_j^*,最优纳什平衡策略 $\boldsymbol{u}^*$ 和 $\boldsymbol{w}^*$.

接下来给出在线无模型积分 PI 算法. 为了规避系统模型的信息,将探测信号 $\boldsymbol{e}$ 先后加入到控制策略 $\boldsymbol{u}_i$ 和干扰策略 $\boldsymbol{w}_i$ 中,假设探测信号是预知的非零有界可测信号,$\| \boldsymbol{e} \| \leqslant \boldsymbol{e}_M$.

系统(1)变为

$$\left.\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})(\boldsymbol{u}_i + \boldsymbol{e}) + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}_i,\\ \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}_i + \boldsymbol{k}(\boldsymbol{x})(\boldsymbol{w}_i + \boldsymbol{e})\ .\end{aligned}\right\}\tag{13}$$

值函数 V_1^i, V_2^i 关于时间的导数为

$$\left.\begin{aligned}\dot{V}_1^i &= (\nabla V_1^i)^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})(\boldsymbol{u}_i + \boldsymbol{e}) + \boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}_i)\ ,\\ \dot{V}_2^i &= (\nabla V_2^i)^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}_i + \boldsymbol{k}(\boldsymbol{x})(\boldsymbol{w}_i + \boldsymbol{e}))\ .\end{aligned}\right\}\tag{14}$$

利用式(11),式(12)从 t 到 $t+T$ 积分式(14)得到

$$V_1^i(\boldsymbol{x}_{t+T}) - V_1^i(\boldsymbol{x}_t) = -\int_t^{t+T} r_1(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{w}_i)\mathrm{d}\tau - 2\int_t^{t+T}\boldsymbol{u}_{i+1}^{\mathrm{T}}\boldsymbol{R}_{11}\boldsymbol{e}\mathrm{d}\tau\ ,\tag{15}$$

$$V_2^i(\boldsymbol{x}_{t+T}) - V_2^i(\boldsymbol{x}_t) = -\int_t^{t+T} r_2(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{w}_i)\mathrm{d}\tau - 2\int_t^{t+T}\boldsymbol{w}_{i+1}^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{e}\mathrm{d}\tau\ .\tag{16}$$

其中 $\boldsymbol{x}(t)$ 和 $\boldsymbol{x}(t+T)$ 记为 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_{t+T}$.

在式(15)和式(16)中没有出现 $\boldsymbol{f}(\boldsymbol{x})$,$\boldsymbol{g}(\boldsymbol{x})$ 和 $\boldsymbol{k}(\boldsymbol{x})$ 的信息,因此得到了下面的无模型积分 PI 算法.

算法 2 的步骤如下:

步骤 1 给定初始稳定控制策略 $\boldsymbol{u}_1$ 和干扰策略 $\boldsymbol{w}_1$,设定 $i=1$.

步骤 2 已知 $\boldsymbol{u}_i, \boldsymbol{w}_i, \boldsymbol{e}$ 解下面的方程求解 $V_1^i, V_2^i, \boldsymbol{u}_{i+1}, \boldsymbol{w}_{i+1}$,

$$V_1^i(\boldsymbol{x}_t) = V_1^i(\boldsymbol{x}_{t+T}) + \int_t^{t+T} r_1(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{w}_i) + 2\int_t^{t+T}\boldsymbol{u}_{i+1}^{\mathrm{T}}\boldsymbol{R}_{11}\boldsymbol{e}\mathrm{d}\tau\ ,\tag{17}$$

$$V_2^i(\boldsymbol{x}_t) = V_2^i(\boldsymbol{x}_{t+T}) + \int_t^{t+T} r_2(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{w}_i) + 2\int_t^{t+T}\boldsymbol{w}_{i+1}^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{e}\mathrm{d}\tau\ .\tag{18}$$

步骤 3 如果 $\| V_j^i - V_j^{i-1} \| \leqslant \varepsilon (j=1,2,\varepsilon$ 是预设的正实数),停止算法并输出 V_1^i, V_2^i;否则,设 $i=i+1$,并转到步骤 2.

在算法 2 中同时更新值函数、控制策略和干扰策略.

3 算法实现

为了实现算法 2,使用 2 个评价神经网络和 2 个执行网络用来近似值函数、控制策略和干扰策略. 这里为了简化表示,记 $m=q=1$. 对式(17)和式(18),$V_j^i(\boldsymbol{x})(j=1,2)$,$\boldsymbol{u}_{i+1}(\boldsymbol{x})$,$\boldsymbol{w}_{i+1}(\boldsymbol{x})$ 可以用单层神经网络表示如下:

$$V_1^i(\boldsymbol{x}) = (\boldsymbol{W}_i^1)^{\mathrm{T}}\boldsymbol{\phi}_1(\boldsymbol{x}) + \varepsilon_i^1(\boldsymbol{x})\ ;\tag{19}$$

$$V_2^i(\boldsymbol{x}) = (\boldsymbol{W}_i^2)^{\mathrm{T}}\boldsymbol{\phi}_2(\boldsymbol{x}) + \varepsilon_i^2(\boldsymbol{x})\ ;\tag{20}$$

$$\boldsymbol{u}_{i+1}(\boldsymbol{x}) = (\boldsymbol{W}_{i+1}^3)^{\mathrm{T}}\boldsymbol{\phi}_3(\boldsymbol{x}) + \varepsilon_{i+1}^3(\boldsymbol{x})\ ;\tag{21}$$

$$\boldsymbol{w}_{i+1}(\boldsymbol{x}) = (\boldsymbol{W}_{i+1}^4)^{\mathrm{T}}\boldsymbol{\phi}_4(\boldsymbol{x}) + \varepsilon_{i+1}^4(\boldsymbol{x})\ .\tag{22}$$

其中:$\boldsymbol{W}_i^j(j=1,2,3,4)$ 是具有适当维数的未知理想权值;$\boldsymbol{\phi}_j(j=1,2,3,4)$ 是激活函数,采用多项式函数;$\varepsilon_i^j(j=1,2,3,4)$ 是有界神经网络近似误差. 当隐藏层的神经元数量趋于无穷时,近似误差一致趋于零. 因为理想的权值未知,假设评价网络和执行网络的输出为

$$\hat{V}_1^i(\boldsymbol{x}) = (\hat{\boldsymbol{W}}_i^1)^{\mathrm{T}}\boldsymbol{\phi}_1(\boldsymbol{x})\ ;\tag{23}$$

$$\hat{V}_2^i(\boldsymbol{x}) = (\hat{\boldsymbol{W}}_i^2)^{\mathrm{T}}\boldsymbol{\phi}_2(\boldsymbol{x})\ ;\tag{24}$$

$$\hat{\boldsymbol{u}}_{i+1}(\boldsymbol{x}) = (\hat{\boldsymbol{W}}_{i+1}^3)^{\mathrm{T}}\boldsymbol{\phi}_3(\boldsymbol{x})\ ;\tag{25}$$

$$\hat{\boldsymbol{w}}_{i+1}(\boldsymbol{x}) = (\hat{\boldsymbol{W}}_{i+1}^4)^{\mathrm{T}}\boldsymbol{\phi}_4(\boldsymbol{x})\ .\tag{26}$$

其中 $\hat{\boldsymbol{W}}_i^1, \hat{\boldsymbol{W}}_i^2, \hat{\boldsymbol{W}}_{i+1}^3, \hat{\boldsymbol{W}}_{i+1}^4$ 是当前估计值.

使用式(23) ~ 式(26),式(17)和式(18)可以写成下面的形式:

$$(\boldsymbol{\psi}_k^1)^{\mathrm{T}}\begin{bmatrix}\hat{\boldsymbol{W}}_i^1\\ \hat{\boldsymbol{W}}_{i+1}^3\end{bmatrix} = \theta_k^1;\tag{27}$$

$$(\boldsymbol{\psi}_k^2)^{\mathrm{T}}\begin{bmatrix}\hat{\boldsymbol{W}}_i^2\\ \hat{\boldsymbol{W}}_{i+1}^4\end{bmatrix} = \theta_k^2.\tag{28}$$

其中:$\theta_k^j = \int_{t+(k-1)T}^{t+kT} r_j(\boldsymbol{x},\boldsymbol{u}_i,\boldsymbol{w}_i)\mathrm{d}\tau (j=1,2)$;

$\boldsymbol{\psi}_k^1 = [(\boldsymbol{\phi}_1(\boldsymbol{x}_{t+(k-1)T}) - \boldsymbol{\phi}_1(\boldsymbol{x}_{t+kT}))^{\mathrm{T}}, -2\int_{t+(k-1)T}^{t+kT}\boldsymbol{R}_{11}\boldsymbol{e}\boldsymbol{\phi}_3^{\mathrm{T}}(\boldsymbol{x})\mathrm{d}\tau]$;

$\boldsymbol{\psi}_k^2 = [(\boldsymbol{\phi}_2(\boldsymbol{x}_{t+(k-1)T}) - \boldsymbol{\phi}_2(\boldsymbol{x}_{t+kT}))^{\mathrm{T}}, -2\int_{t+(k-1)T}^{t+kT}\boldsymbol{R}_{22}\boldsymbol{e}\boldsymbol{\phi}_4^{\mathrm{T}}(\boldsymbol{x})\mathrm{d}\tau]$.

因为式(27)和式(28)是一维方程,不能保证解的唯一性. 使用最小二乘法计算参数向量:

$$(\boldsymbol{\Phi}_i^1)^{\mathrm{T}}\begin{bmatrix}\hat{\boldsymbol{W}}_i^1\\ \hat{\boldsymbol{W}}_{i+1}^3\end{bmatrix} = \boldsymbol{\Theta}_i^1, (\boldsymbol{\Phi}_i^2)^{\mathrm{T}}\begin{bmatrix}\hat{\boldsymbol{W}}_i^2\\ \hat{\boldsymbol{W}}_{i+1}^4\end{bmatrix} = \boldsymbol{\Theta}_i^2.$$

其中:

$$\boldsymbol{\Phi}_i^j = [\boldsymbol{\psi}_1^j, \cdots, \boldsymbol{\psi}_K^j](j=1,2)\ ;$$

$$\boldsymbol{\Theta}_i^j = [\boldsymbol{\theta}_1^j, \cdots, \boldsymbol{\theta}_K^j](j=1,2)\ .$$

如果 $\boldsymbol{\Phi}_i^j$ 列满秩，则：

$$\begin{bmatrix}\hat{\boldsymbol{W}}_i^1\\ \hat{\boldsymbol{W}}_{i+1}^3\end{bmatrix}=(\boldsymbol{\Phi}_i^1(\boldsymbol{\Phi}_i^1)^{\mathrm{T}})^{-1}\boldsymbol{\Phi}_i^1\boldsymbol{\Theta}_i^1;$$

$$\begin{bmatrix}\hat{\boldsymbol{W}}_i^2\\ \hat{\boldsymbol{W}}_{i+1}^4\end{bmatrix}=(\boldsymbol{\Phi}_i^2(\boldsymbol{\Phi}_i^2)^{\mathrm{T}})^{-1}\boldsymbol{\Phi}_i^2\boldsymbol{\Theta}_i^2.$$

因此采样数据 K 要取得足够大，$K_{\min}=\mathrm{rank}(\boldsymbol{\Phi}_i^j)=N_1+N_2$（$N_1$，$N_2$ 分别对应 $\hat{\boldsymbol{W}}_i^1$ 和 $\hat{\boldsymbol{W}}_{i+1}^3$ 的维数），以保证 $(\boldsymbol{\Phi}_i^1(\boldsymbol{\Phi}_i^1)^{\mathrm{T}})^{-1}$ 存在. 最小二乘问题可以通过采集系统数据实时求解.

4　仿真例子

为了验证算法的有效性，本文利用 Matlab 软件对算法 2 进行了仿真实验.

考虑下面的非线性系统：

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}+\boldsymbol{k}(\boldsymbol{x})\boldsymbol{w}.$$

其中：

$$\boldsymbol{f}(\boldsymbol{x})=\begin{bmatrix}x_2-2x_1\\ -x_2-0.5x_1+0.25x_2(\cos(2x_1+2))^2+\\ 0.25x_2(\sin(4x_1^2)+2)^2\end{bmatrix};$$

$$\boldsymbol{g}(\boldsymbol{x})=\begin{bmatrix}0\\ \cos(2x_1+2)\end{bmatrix};\boldsymbol{k}(\boldsymbol{x})=\begin{bmatrix}0\\ \sin(4x_1^2)+2\end{bmatrix}.$$

性能指标（2），（3）分别定义为 $\boldsymbol{Q}_1(\boldsymbol{x})=2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$，$\boldsymbol{R}_{11}=\boldsymbol{R}_{12}=2\boldsymbol{I}$，$\boldsymbol{Q}_2(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$，$\boldsymbol{R}_{21}=\boldsymbol{R}_{22}=\boldsymbol{I}$，其中 $\boldsymbol{I}$ 代表单位矩阵.

设置初始状态 $\boldsymbol{x}_0=[1\quad 2]^{\mathrm{T}}$. 神经网络激活函数 $\boldsymbol{\phi}_1(\boldsymbol{x})=\boldsymbol{\phi}_2(\boldsymbol{x})=\boldsymbol{\phi}_3(\boldsymbol{x})=\boldsymbol{\phi}_4(\boldsymbol{x})=[x_1^2\quad x_1x_2\quad x_2^2]^{\mathrm{T}}$，其初始参数均设置为零. 探测信号选择 $e=\sin(t)$，算法在线执行，系统响应数据采集周期 $T=0.04$，最小二乘法需要 20 组数据，所以每0.8 s更新一次激活函数参数. 图1表示 V_1 的参数对更新次数的变化，图 2 表示 V_2 的参数对更新次数的变化，可以看出在 3 次更新以后参数基本保持不变. 图 3 表示在得到控制器和干扰器作用下的系统响应图.

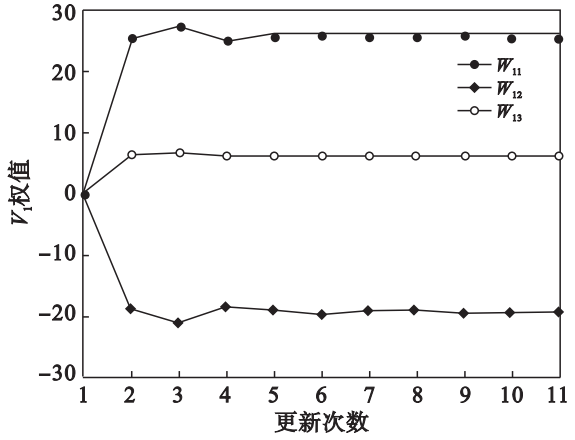

图 1　V_1 参数更新

Fig. 1　V_1 parameter update

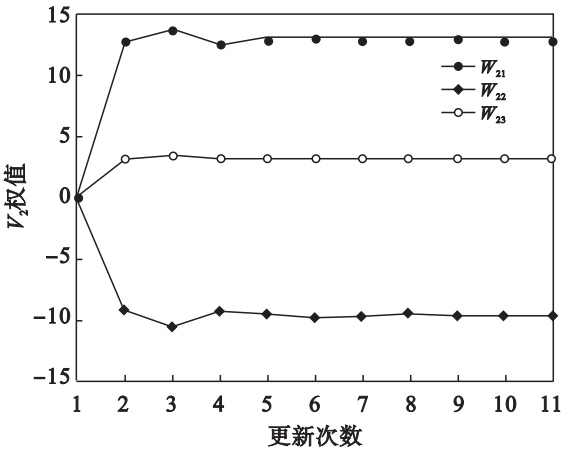

图 2　V_2 参数更新

Fig. 2　V_2 parameter update

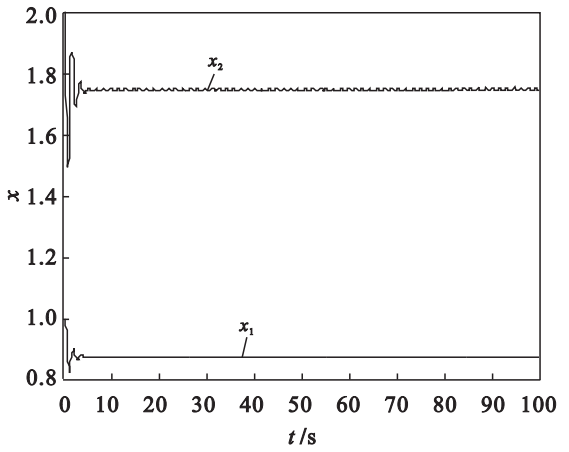

图 3　系统状态响应

Fig. 3　System state response

从图 1 和图 2 可以看出，值函数的权重 $\boldsymbol{W}$ 很快就收敛到了稳定值. 从图 3 可以看出得到的控制器可以保证系统一致最终有界的.

5　结　　语

本文提出一种无模型积分策略迭代算法求解非零和博弈问题，通过在控制策略和干扰策略中加入已知噪声，消除算法迭代对系统信息模型的依赖. 值函数、控制策略和干扰策略同时更新，加快了算法收敛速度. 最后仿真算例验证了算法的有效性.

（下转第 326 页）