Xinxing LI, Lele XI, Wenzhong ZHA, Zhihong PENG, 2022. Minimax *Q*-learning design for H_∞ control of linear discrete-time systems. *Frontiers of Information Technology & Electronic Engineering,* 23(3):438-451.
https://doi.org/10.1631/FITEE.2000446

Minimax *Q*-learning design for H_∞ control of linear discrete-time systems

Key words: H_∞ control; Zero-sum dynamic game; Reinforcement learning; Adaptive dynamic programming; Minimax *Q*-learning; Policy iteration

Corresponding author: Wenzhong ZHA
E-mail: zhawenzhong@126.com
ORCID: https://orcid.org/0000-0003-2718-5052

Motivation

1. Due to the uncertainty caused by the environment, most practical systems always suffer from external disturbances. H_∞ control is one of the most effective approaches to attenuate the effect of disturbances on the system performance. However, obtaining the H_∞ controller requires solving the nonlinear Hamilton–Jacobi–Isaacs (HJI) equation.

2. Reinforcement learning (RL), or adaptive dynamic programming (ADP), is an efficient machine learning technique for dealing with H_∞ control problems, because RL overcomes the curse of dimensionality and achieves the goal of online learning controller design.

Motivation (Cont'd)

3. Over the past few years, many model-free *Q*-learning approaches have been developed for H_∞ control problems. Note that most of the existing *Q*-learning methods are based on value iteration. Meanwhile, theoretical foundations for policy-iteration-based *Q*-learning are relatively lacking in the literature. Inspired by off-policy RL and adaptive control, we develop a novel policy-iteration-based minimax *Q*-learning method for H_∞ control of linear discrete-time systems.

Main idea

1. First, we prove the convergence of the model-based offline policy iteration algorithm by proving its equivalence to Newton's method for solving the related game algebraic Riccati equation (GARE).
2. On the basis of the model-based offline policy iteration algorithm, we develop the model-free minimax *Q*-learning algorithm.

Algorithm 1 Model-based offline policy iteration algorithm

1: Start with a set of initially stabilizing feedback gains (K_1^1, K_2^1) // Initialization

2: For the given stabilizing feedback gains (K_1^l, K_2^l), solve for the corresponding value matrix P^{l+1} via the following matrix equation: // Policy evaluation

$$P^{l+1} = S + (K_1^l)^{\mathrm{T}} R K_1^l - \gamma^2 (K_2^l)^{\mathrm{T}} K_2^l + (A - BK_1^l - DK_2^l)^{\mathrm{T}} P^{l+1} (A - BK_1^l - DK_2^l) \tag{11}$$

3: Update the control policy and disturbance policy using the following equation: // Policy improvement

$$\begin{pmatrix} K_1^{l+1} \\ K_2^{l+1} \end{pmatrix} = \zeta(P^{l+1}) \begin{bmatrix} B^{\mathrm{T}} P^{l+1} A \\ D^{\mathrm{T}} P^{l+1} A \end{bmatrix}, \tag{12}$$

where $\zeta(P^{l+1})$ is defined as follows:

$$\zeta(P^{l+1}) = \begin{bmatrix} R + B^{\mathrm{T}} P^{l+1} B & B^{\mathrm{T}} P^{l+1} D \\ D^{\mathrm{T}} P^{l+1} B & D^{\mathrm{T}} P^{l+1} D - \gamma^2 I \end{bmatrix}^{-1}$$

4: Stop if $\|K_i^{l+1} - K_i^l\| \leq \varepsilon$ $(i = 1, 2)$, where ε is a threshold; otherwise, set $l = l + 1$ and go to step 2

Method

1. We define the following *Q*-function:

$$\begin{aligned} & Q^{l+1}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \\ =& \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x}_k + \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u}_k - \gamma^2 \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{w}_k + \boldsymbol{x}_{k+1}^{\mathrm{T}} \boldsymbol{P}^{l+1} \boldsymbol{x}_{k+1} \end{aligned}$$

By using the Kronecker product quadratic polynomial basis vector, we have

$$\begin{aligned} & \boldsymbol{W}_{\mathrm{c},l+1}^{\mathrm{T}} \boldsymbol{\sigma}_k \\ =& \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x}_k + \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u}_k - \gamma^2 \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{w}_k + \boldsymbol{W}_{\mathrm{c},l+1}^{\mathrm{T}} \boldsymbol{\sigma}_{k+1,l}. \end{aligned}$$

Then the policy evaluation problem is translated into the following parameter estimation problem:

$$\begin{aligned} e_k =& \boldsymbol{x}_k^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x}_k + \boldsymbol{u}_k^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u}_k - \gamma^2 \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{w}_k \\ & + \hat{\boldsymbol{W}}_{\mathrm{c},l+1}^{\mathrm{T}}(i)(\boldsymbol{\sigma}_{k+1,l} - \boldsymbol{\sigma}_k). \end{aligned}$$

Method (Cont'd)

We use the recursive least squares (RLS) method to learn the true value:

$$\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i+1)=\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i)-\frac{\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k e_k}{1+\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k}, \tag{26a}$$

$$\boldsymbol{P}_l(i+1)=\boldsymbol{P}_l(i)-\frac{\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)}{1+\bar{\boldsymbol{\sigma}}_k^{\mathrm{T}}\boldsymbol{P}_l(i)\bar{\boldsymbol{\sigma}}_k}. \tag{26b}$$

---

2. After performing policy evaluation, we carry out the policy improvement step by using the following normalized gradient method:

$$\hat{\boldsymbol{K}}_{1,l+1}(j+1)=\hat{\boldsymbol{K}}_{1,l+1}(j)-\frac{\beta}{\left(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k\right)^2}\times\frac{\partial}{\partial\hat{\boldsymbol{K}}_{1,l+1}(j)}Q^{l+1}\left(\boldsymbol{x}_k,\hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k,\hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k\right), \tag{29a}$$

$$\hat{\boldsymbol{K}}_{2,l+1}(j+1)=\hat{\boldsymbol{K}}_{2,l+1}(j)+\frac{\beta}{\left(1+\boldsymbol{x}_k^{\mathrm{T}}\boldsymbol{x}_k\right)^2}\times\frac{\partial}{\partial\hat{\boldsymbol{K}}_{2,l+1}(j)}Q^{l+1}\left(\boldsymbol{x}_k,\hat{\boldsymbol{K}}_{1,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k,\hat{\boldsymbol{K}}_{2,l+1}^{\mathrm{T}}(j)\boldsymbol{x}_k\right). \tag{29b}$$

Method (Cont'd)

Compared with Algorithm 1, Algorithm 2 is completely model-free, thus robust to the drift in system dynamics and the inaccuracy in system modeling. In addition, both policy evaluation and policy improvement are carried out in an online adaptive way using the state samples generated by the behavior polices.

Algorithm 2 Online minimax Q-learning algorithm

1: Start with a set of initially stabilizing feedback gains $(\boldsymbol{K}_1^1, \boldsymbol{K}_2^1)$ // Initialization

2: For the given stabilizing feedback gains $(\bar{\boldsymbol{K}}_1^1, \bar{\boldsymbol{K}}_2^1)$, run Eqs. (26a) and (26b) until $\hat{\boldsymbol{W}}_{\mathrm{c},l+1}(i+1)$ converges to $\boldsymbol{W}_{\mathrm{c},l+1}$ // Policy evaluation

3: Using the obtained $\boldsymbol{W}_{\mathrm{c},l+1}$, run Eqs. (29a) and (29b) simultaneously until $(\hat{\boldsymbol{K}}_{1,l+1}(j), \hat{\boldsymbol{K}}_{2,l+1}(j))$ converges to $(\bar{\boldsymbol{K}}_1^{l+1}, \bar{\boldsymbol{K}}_2^{l+1})$ // Policy improvement

4: Stop if $\left\|\boldsymbol{K}_i^{l+1} - \boldsymbol{K}_i^l\right\| \leq \varepsilon$ $(i = 1, 2)$, where ε is a threshold; otherwise, set $l = l + 1$ and go to step 2

Major results

Convergence of the minimax *Q*-learning method:

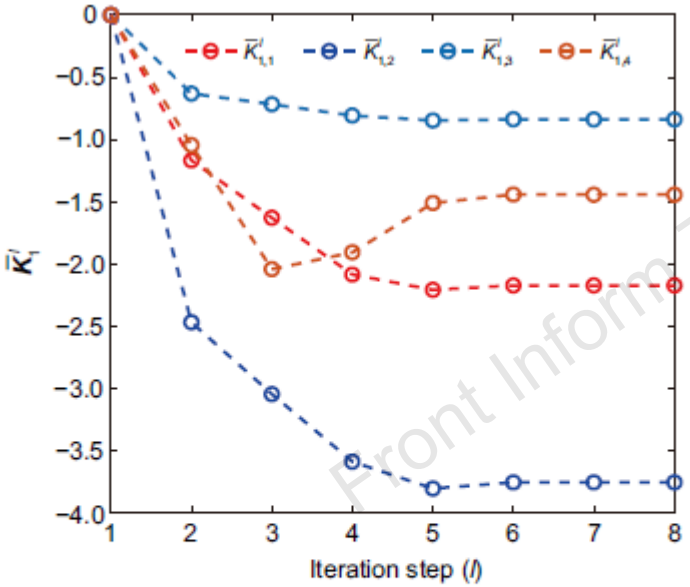

Fig. 1 Evolution of the controller feedback gain $\bar{K}_1^l$ in the policy-iteration-based minimax Q-learning method, where $\bar{K}_1^l = [\bar{K}_{1,1}^l, \bar{K}_{1,2}^l, \bar{K}_{1,3}^l, \bar{K}_{1,4}^l]^{\mathrm{T}}$

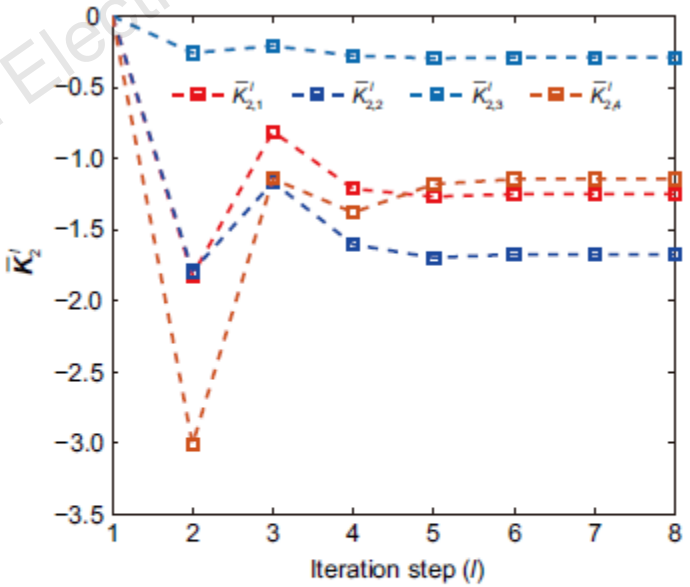

Fig. 2 Evolution of the disturbance feedback gain $\bar{K}_2^l$ in the policy-iteration-based minimax Q-learning method, where $\bar{K}_2^l = [\bar{K}_{2,1}^l, \bar{K}_{2,2}^l, \bar{K}_{2,3}^l, \bar{K}_{2,4}^l]^{\mathrm{T}}$

Major results (Cont'd)

Convergence of the value-iteration-based Q-learning method:

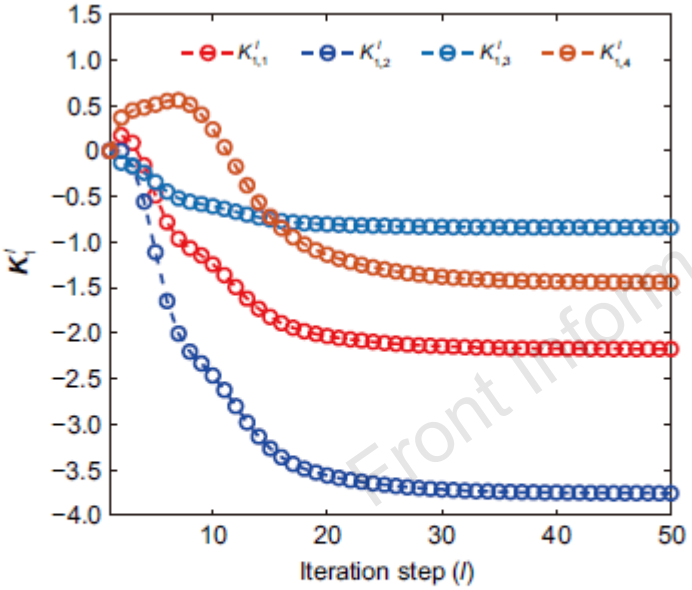

Fig. 5 Evolution of the controller feedback gain K_1^l in the value-iteration-based Q-learning method, where $K_1^l = [K_{1,1}^l, K_{1,2}^l, K_{1,3}^l, K_{1,4}^l]$

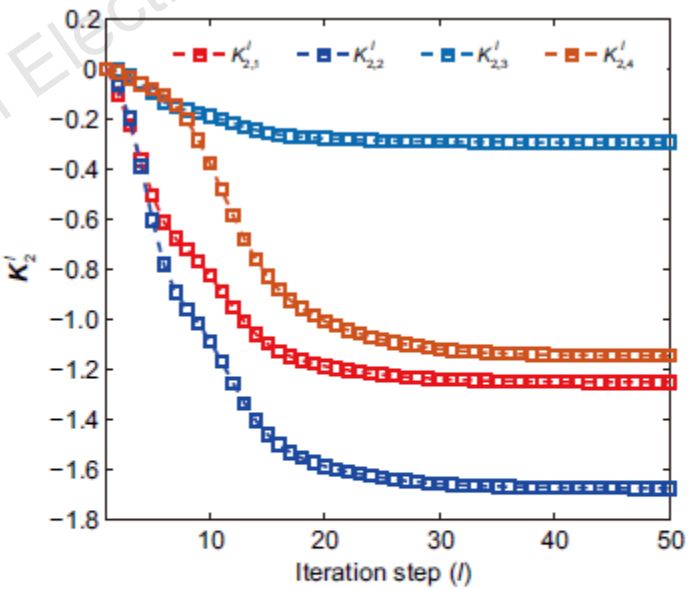

Fig. 6 Evolution of the disturbance feedback gain K_2^l in the value-iteration-based Q-learning method, where $K_2^l = [K_{2,1}^l, K_{2,2}^l, K_{2,3}^l, K_{2,4}^l]$

Major results (Cont'd)

Disturbance rejection performance of the obtained controller:

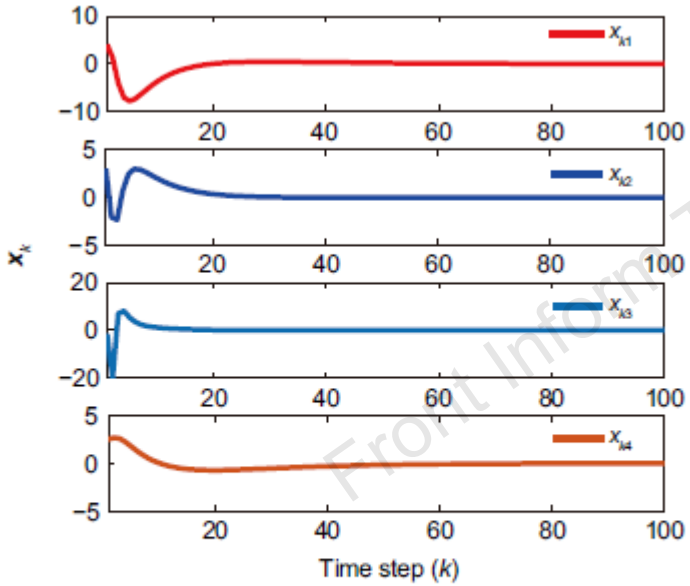

Fig. 3 State evolution of system (51) by implementing $u_k = (\bar{K}_1^8)^{\mathrm{T}} x_k$ under disturbance $w_k = 5\exp(-0.16k)$

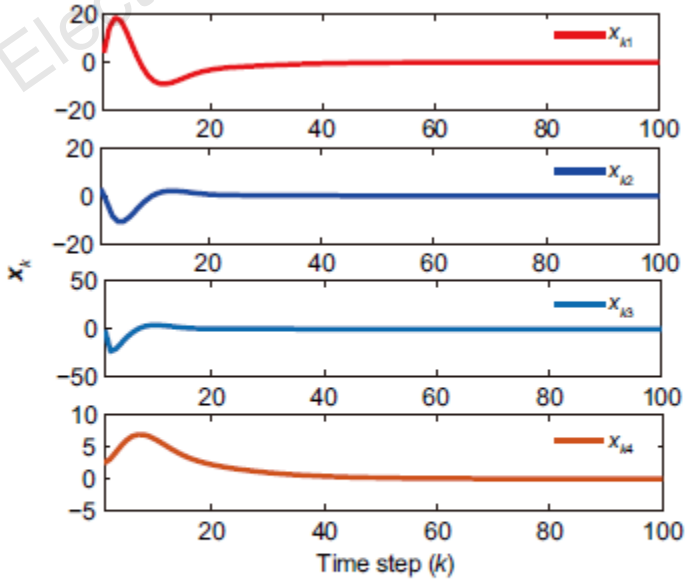

Fig. 4 State evolution of system (51) by implementing $u_k = (\bar{K}_1^8)^{\mathrm{T}} x_k$ under the worst-case disturbance $w_k = (\bar{K}_2^8)^{\mathrm{T}} x_k$

Conclusions

1. A policy-iteration-based minimax *Q*-learning method is developed for learning the H_∞ controller online using the state samples generated by the behavior policies, without querying the system model.
2. By employing a normalized gradient method, a novel policy improvement scheme is proposed.
3. The rigorous convergence analysis of the proposed minimax *Q*-learning method is established under some persistence of excitation conditions and learning rate constraints.

李新兴：中国电子科技集团公司信息科学研究院工程师。2019年毕业于北京理工大学自动化学院控制科学与工程专业，获工学博士学位。研究领域：强化学习、博弈论、最优控制、复杂系统、体系理论等。