Information Sciences 631 (2023) 396–411

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Reward inference of discrete-time expert's controllers: A complementary learning approach ☆

Adolfo Perrusquía *, Weisi Guo

School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, MK43 0AL, UK

ARTICLE INFO

Keywords:
Discrete-time linear systems
Hippocampus
Neocortex
Striatum
Complementary learning
Gradient update rule

ABSTRACT

Uncovering the reward function of optimal controllers is crucial to determine the desired performance that an expert wants to inject to a certain dynamical system. In this paper, a reward inference algorithm of discrete-time expert's controllers is proposed. The approach is inspired by the complementary mechanisms of the striatum, neocortex, and hippocampus for decision making and experience transference. These systems work together to infer the reward function associated to expert's controller using the complementary merits of data-driven and online learning methods. The proposed approach models the neocortex system as two independent learning algorithms given by a Q-learning algorithm and a gradient identification rule. The hippocampus is modelled by a least-squares update rule that extracts the relation from the states and control inputs of the expert's data. The striatum is modelled by an inverse optimal control algorithm which iteratively finds the hidden reward function. Lyapunov stability theory is used to show the stability and convergence of the proposed approach. Simulation studies are given to demonstrate the effectiveness of the proposed complementary learning algorithm.

1. Introduction

Optimal and adaptive control [1] are established control philosophies that serve as the basis for the design of modern adaptive dynamic programming (ADP) [2,3] and reinforcement learning (RL) algorithms [1,4]. The main aim of these algorithms is to find the optimal control policy that minimizes a scalar reward function (also known as utility function) in an infinite or discounted horizon. The linear quadratic regulator (LQR) is the most popular optimal control approach that serves as the basis for the design of ADP/RL algorithms [5]. In the sequel of the paper, LQR controllers and optimal controllers are used interchangeably.

There exists an extensive literature [6–8] that studies the design of model-based and model-free optimal adaptive controllers with impressive results, e.g., Lyapunov recursions of an algebraic Riccati equation [6,9], partially model-based ADP algorithms [10,11] with critic or actor-critic structures [7,12–14] and model-free RL algorithms such as Q-learning [15] and their variants [16–18]. The common term of these different learning architectures is the reward function which defines the task that the system aims to achieve.

In a control perspective [1], the reward function is a predefined function that is designed by an expert to guarantee a desired performance [19] in terms of the control input, constraints, output response, time, etc.; and hence, it is difficult or impossible

☆ This work was supported by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Research Fellowship Program.

* Corresponding author.
E-mail address: Adolfo.Perrusquia-Guzman@cranfield.ac.uk (A. Perrusquía).

https://doi.org/10.1016/j.ins.2023.02.079
Received 20 November 2022; Received in revised form 20 February 2023; Accepted 26 February 2023

Available online 2 March 2023

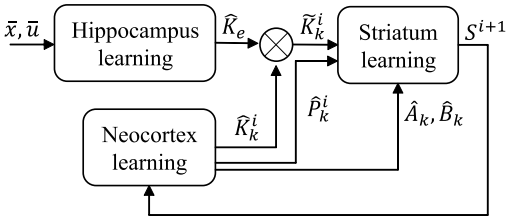


Fig. 1. Complementary learning scheme for reward inference of discrete-time linear systems.

to design the same reward function by different users. Furthermore, there exists a large amount of experts' data[1] [4] (with hidden reward function) that model a desired performance which can be exploited by complex machine learning algorithms for experience transference and imitation learning. However, the generalization of this kind of approaches is poor for different systems, environments, and tasks.

In view of the above, the main objective of this paper is to infer the hidden reward function associated to the expert's controller that ensures the same expert's desired performance.[2] Reward extraction has been addressed by inverse optimal control (IOC) and inverse reinforcement learning (IRL) algorithms [20]. Whilst the IOC algorithm is strongly dependent on dynamic knowledge [21], the IRL requires assumptions of the reward structure, e.g., a binary function [22,23]. Furthermore, constraints must be added in the IOC/IRL algorithms to avoid multiple reward solutions.

Multiple knowledge representation has been adopted to model a specific problem using different learning mechanisms, e.g., verbal, perception, and deep neural models that enable complementary learning for better learning outcomes [24,25]. A similar approach known as human-behaviour learning [26], which is inspired by the complementary learning of the hippocampus, neocortex, and striatum, has been adopted for complex decision making and experience transference. Data driven and online learning methods are used to model these complementary mechanisms [27,28].

The hippocampus is modelled as a fast learning system which is directly related to memory and experience [29–33], e.g., exploration techniques, eligibility traces [34], experience replay [4,26], and expert's data. The neocortex is a slow online learning system that have good pattern association, adaptability, and generalization [28]. In view of the above, the interplay between the hippocampus and the neocortex can be modelled by any ADP/RL algorithm [35,36] due to their high capabilities for pattern association and experience exploitation. The striatum is a learning mechanism that has complementary properties [37,38] to connect fast (hippocampus) and slow (neocortex) learning mechanisms to achieve complex behaviours and experience transference [39].

In this paper, the reward inference of expert's data is discussed. Fig. 1 shows the general scheme of the proposed approach. The scheme is based on an experience transference algorithm inspired by the complementary properties of the striatum, neocortex and hippocampus learning systems. The hippocampus is modelled as expert's data with hidden reward function and a least-squares rule that extracts the relation between the states and control inputs. The neocortex is designed as a Q-learning algorithm and a gradient identification rule. On the one hand, the Q-learning algorithm learns an optimal control policy relatively to an iterative reward function. On the other hand, the gradient identification rule estimates the parameters of the system. The striatum is designed as an IOC algorithm [23,40] that computes new improved reward functions in each iteration. These rewards feed the Q-learning algorithm to obtain an improved policy which is closer to the hippocampus policy.

The main contributions of this paper with respect to previous developments in experience transference are:

(1) A model-free experience transference algorithm that merges, in a complementary mechanism, expert's data and online data for policy generalization.
(2) The complementary algorithm combines the merits of data-driven and online learning methods to infer the reward function from expert's data using well-defined and interconnected algorithms (Gradient and least-squares identification rules, Q-learning and IOC).
(3) The reward function has a quadratic structure that avoids the binary reward function constraint of standard IRL algorithms.
(4) Rigorous stability and parameter convergence results related to the experience transference algorithm using Lyapunov stability theory are given.

2. Preliminaries

Consider the following discrete-time system

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

where $x_k \in \mathbb{R}^n$ defines the state vector, $u_k \in \mathbb{R}^m$ stands to the control input, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the dynamics of the system, $k \in \mathbb{N}$ denotes a time index. Assume that the pair (A, B) is stabilizable.

[1] Experts' data are a fixed dataset composed of the states and control inputs trajectories of a controlled system.
[2] The expert's desired performance defines a trajectory that fulfils a desired behaviour.

Classical optimal control approaches [1] seek an optimal control policy $u_k^* = -Kx_k$, where $K \in \mathbb{R}^{m\times n}$ is a stabilizing gain that minimizes a pre-defined reward/utility function. There exist several approaches that solve the optimal control problem using either off-line or on-line algorithms [8,41].

Whilst ADP and RL algorithms combine the advantages of optimal and adaptive control theories, the reward design is still handcrafted and, most of the time, it is designed to fulfil some expert's constraints [19]. Furthermore, the data science field has provided a wide amount of experts' data with hidden reward functions that show how the system must behave. Imitation learning and fuzzy systems are generally used for this kind of approaches in experience transference [42,43]. However, the system learns to imitate and improve the expert's trajectory but cannot be generalized to different trajectories or other systems and environments. In the next sections we clearly define each element of the proposed approach (see Fig. 1).

3. Hippocampus learning system

The following matrices are constructed with l data points of states $x_i^e \in \mathbb{R}^n$ and control inputs $u_i^e \in \mathbb{R}^m$ collected from the closed-loop trajectories of an expert performance: $\bar{x} = [x_0^e, \cdots, x_{l-1}^e] \in \mathbb{R}^{n\times l}$ and $\bar{u} = [u_0^e, \cdots, u_{l-1}^e] \in \mathbb{R}^{m\times l}$, with $i = 0, ..., l$. The collected data fulfils the following discrete-time dynamics

$$x_{k+1}^e = Ax_k^e + Bu_k^e. \tag{2}$$

It is assumed that the collected data are obtained from the optimization of the following value function $V_e(x_k^e)$ [6]

$$V_e(x_k^e) = \sum_{i=k}^{\infty}((x_i^e)^\top S_e x_i^e + (u_i^e)^\top R_e u_i^e) \tag{3}$$

where $S_e = S_e^\top \geq 0 \in \mathbb{R}^{n\times n}$ and $R_e = R_e^\top > 0 \in \mathbb{R}^{m\times m}$ are the unknown weight matrices of the reward function associated to the expert's trajectories.

Remark 1. The use of quadratic reward functions allows obtaining optimal solutions in the sense of the H_2 control [19]. Quadratic reward functions are convex functions and ensure that the optimum exists for the class of stabilizable discrete-time linear systems that we discuss in this paper. In addition, discontinuous control policies are avoided which prevents chattering problems or unstable closed-loop performances.

The value function (3) can be written as the following Bellman equation

$$V_e(x_k^e) = (x_k^e)^\top S_e x_k^e + (u_k^e)^\top R_e u_k^e + V_e(x_{k+1}^e). \tag{4}$$

Assume that the optimal control exists, then it is possible to write the optimal value function $V_e^*(x_k)$ in a convex form as

$$V_e^*(x_k^e) = (x_k^e)^\top P_e x_k^e, \tag{5}$$

for some kernel matrix $P_e = P_e^\top > 0 \in \mathbb{R}^{n\times n}$ which is solution of the following discrete algebraic Riccati equation (DARE)

$$P_e = A^\top P_e A + S_e - A^\top P_e B(R_e + B^\top P_e B)^{-1} B^\top P_e A. \tag{6}$$

The Hamiltonian associated to (2) and (4) is

$$\begin{aligned} H_e(x_k^e, u_k^e) &= (x_k^e)^\top S_e x_k^e + (u_k^e)^\top R_e u_k^e - (x_k^e)^\top P_e x_k^e \\ &\quad + (Ax_k^e + Bu_k^e)^\top P_e (Ax_k^e + Bu_k^e). \end{aligned} \tag{7}$$

The optimal control policy $(u_k^e)^*$ is obtained by differentiating the Hamiltonian with respect to u_k^e as

$$(u_k^e)^* = -K_e x_k^e = -(R_e + B^\top P_e B)^{-1} B^\top P_e A x_k^e, \tag{8}$$

where $K_e = (R_e + B^\top P_e B)^{-1} B^\top P_e A$. Hence, each data point on $\bar{x}$ and $\bar{u}$ satisfy (8).

The optimal control policy (8) can be written in terms of the collected data $\bar{x}$ and $\bar{u}$ to compute an estimate of the control gain $\hat{K}_e \in \mathbb{R}^{m\times n}$ as

$$\begin{aligned} \bar{u} &= -\hat{K}_e \bar{x} \\ \hat{K}_e &= -\bar{u}\bar{x}^\top(\bar{x}\bar{x}^\top)^{-1}. \end{aligned} \tag{9}$$

The LS solution (9) gives an approximation of the control gain K_e since the vectors $\bar{x}$ and $\bar{u}$ may contain noise. However, since the expert's data do not provide any constraint for the reward function then multiple and unstable solutions can be obtained [44]. To avoid this issue, the weight matrix R_e is assumed to be known.

One important technique to get experience is to satisfy a persistence of excitation (PE) condition [45] given by the next lemma.

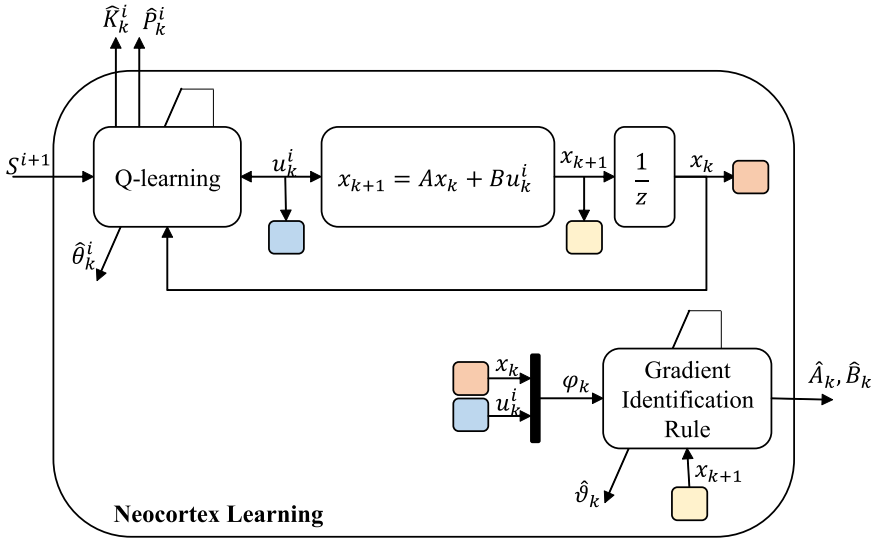


Fig. 2. Neocortex learning scheme.

Lemma 1. *[46] A vector $\Phi_k \in \mathbb{R}^t$ is said to be PE in k steps if there exist constants $\beta_1, \beta_2 \in \mathbb{R}_+$, such that*

$$\beta_1 I \leq \sum_{i=0}^{k-1} \Phi_i \Phi_i^\top \leq \beta_2 I. \tag{10}$$

The above Lemma will be used by the neocortex learning system to ensure parameter estimates convergence.

4. Neocortex learning system

The block diagram of the neocortex learning system is illustrated in Fig. 2. This system is composed of two parallel algorithms that are computed simultaneously: a Q-learning update rule and a gradient identification algorithm. The Q-learning algorithm [17] is mainly used to obtain an optimal/near optimal control policy of the control problem using the reward function extracted by the striatum. The outputs of the Q-learning algorithm are a new control gain $\widehat{K}_k^i \in \mathbb{R}^{m\times n}$ and the kernel matrix $\widehat{P}_k^i \in \mathbb{R}^{n\times n}$. The identification algorithm is used to estimate online the matrices A and B of (1) to feed the striatum learning system. Only the states and control measurements are used to feed the identification rule.

4.1. Q-learning

Define the next value function

$$V(x_k) = \sum_{j=k}^{\infty} (x_j^\top S^i x_j + u_j^{i\top} R u_j^i), \tag{11}$$

where $S^i = S^{i\top} \geq 0 \in \mathbb{R}^{n\times n}$ and $R = R^\top = R_e > 0 \in \mathbb{R}^{m\times m}$ are the weight matrices of the reward function and $u_k^i \in \mathbb{R}^m$ is the neocortex control policy of iteration i. Since the optimal control exists, then the optimal value function for the weights S^i and R verifies

$$V^*(x_k) = x_k^\top P^i x_k, \tag{12}$$

for some kernel matrix $P^i = P^{i\top} > 0 \in \mathbb{R}^{n\times n}$. The Hamiltonian associated to (1) and (11) is

$$\begin{aligned} H(x_k, u_k^i) &= x_k^\top S^i x_k + u_k^{i\top} R u_k^i - x_k^\top P^i x_k \\ &\quad + (Ax_k + Bu_k^i)^\top P^i (Ax_k + Bu_k^i). \end{aligned} \tag{13}$$

The action value function $Q(x,u)$ is used to compute the Q-learning algorithm since A and B are assumed to be unknown. The Q-function verifies

$$Q(x_k, u_k^i) = V^*(x_k) + H(x_k, u_k^i). \tag{14}$$

In addition, we have that $Q^*(x_k,u_k^*)=V^*(x_k)$. The Q-function (14) can be expressed in matrix form as [16]

$$\begin{aligned} Q(x_k,u_k^i) &= \begin{bmatrix} x_k \\ u_k^i \end{bmatrix}^\top \begin{bmatrix} A^\top P^i A + S^i & B^\top P^i A \\ A^\top P^i B & B^\top P^i B + R \end{bmatrix} \begin{bmatrix} x_k \\ u_k^i \end{bmatrix} \\ &= z_k^\top \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} z_k = z_k^\top M^i z_k, \end{aligned} \tag{15}$$

where $z_k=[x_k^\top,{u_k^i}^\top]^\top\in\mathbb{R}^q$ and $M^i\in\mathbb{R}^{q\times q}$, with $q=n+m$. Then it follows that

$$Q(x_k,u_k^i)=\theta^{i\top}(z_k\otimes z_k), \tag{16}$$

where $\theta^i=\text{vech}(M^i)\in\mathbb{R}^{\frac{1}{2}q(q+1)}$, $\text{vech}(M^i)$ denotes the half vectorization operator and $\otimes$ defines the symmetric Kronecker product.

The optimal neocortex control policy is obtained by computing the stationary condition $u_k^{i*}=\frac{\partial Q(x_k,u_k^i)}{\partial u_k^i}=0$, which yields

$$u_k^{i*}=-K^i x_k=-Q_{uu}^{-1}Q_{xu}^\top x_k, \tag{17}$$

where $K^i\equiv Q_{uu}^{-1}Q_{xu}^\top\in\mathbb{R}^{m\times n}$. The Bellman equation [35] written in terms of the Q-function is given by

$$Q(x_k,u_k^{i*})=x_k^\top S^i x_k+(u_k^{i*})^\top R u_k^{i*}+Q(x_{k+1},u_{k+1}^{i*}). \tag{18}$$

The Bellman equation can be expressed as

$$0 \quad = x_k^\top S^i x_k+{u_k^i}^\top R u_k^i+\theta^{i\top}\Phi_k \tag{19}$$

where $\Phi_k=(z_{k+1}\otimes z_{k+1}-z_k\otimes z_k)$. The optimal parameters θ^i are unknown because the parameters associated to the dynamics of the system (1) and the optimal kernel matrix P^i of (12) are unknown. Hence, an approximation is used to estimate θ^i and P^i. For this purpose, consider the following estimate of (16)

$$\widehat{Q}(x_k,u_k^i)=\widehat{\theta}_k^{i\top}(z_k\otimes z_k), \tag{20}$$

where $\widehat{\theta}_k^i\in\mathbb{R}^{\frac{1}{2}q(q+1)}$ is an estimate of θ^i. The temporal difference (TD) error associated to the approximator (20) is

$$\delta_k=x_k^\top S^i x_k+{u_k^i}^\top R u_k^i+\widehat{\theta}_k^{i\top}\Phi_k. \tag{21}$$

The Q-learning algorithm is designed to minimize the TD error, i.e., $\delta_k\to 0$ as $k\to\infty$. The following cost index is defined [41]

$$E=\frac{1}{2}\delta_k^2. \tag{22}$$

A standard gradient-descent technique is used to compute the parameter estimates vector $\widehat{\theta}^i$ that minimizes the cost index (22). The update rule is [3]

$$\widehat{\theta}_{k+1}^i=\widehat{\theta}_k^i-\alpha_1\delta_k\Phi_k, \tag{23}$$

where $\alpha_1\in(0,1]\in\mathbb{R}_+$ is the learning rate. The TD-error can also be written as

$$\delta_k=\widetilde{\theta}_k^{i\top}\Phi_k, \tag{24}$$

where $\widetilde{\theta}_k^i=\widehat{\theta}_k^i-\theta^i\in\mathbb{R}^{\frac{1}{2}q(q+1)}$ is the parametric error. Since a complete set of basis functions is used in Φ_k, then there is no residual error in the approximation. So, the update rule (23) can be rewritten in terms of the parametric error $\widetilde{\theta}_k^i$ as

$$\widetilde{\theta}_{k+1}^i=\widetilde{\theta}_k^i-\alpha_1\Phi_k\Phi_k^\top\widetilde{\theta}_k^i. \tag{25}$$

The following theorem establishes the parameter estimates convergence, that is, $\widehat{\theta}_k^i\to\theta^i$ as $k\to\infty$ which implies that $\widehat{M}_k^i\to M^i$, where $\widehat{M}_k^i\in\mathbb{R}^{(n+m)\times(n+m)}$ is an estimate of matrix M^i.

Theorem 1. *Consider the parametric error dynamics (25) of the neocortex learning system. Let the regressor fulfils the PE condition of the hippocampus learning system (10). If the learning rate α_1 satisfies*

$$\alpha_1=\alpha_0\frac{1}{\|\Phi_k\|^2+1} \tag{26}$$

for some $\alpha_0\in(0,1]$, then the parametric error $\widetilde{\theta}_k^i$ converges exponentially to zero as $k\to\infty$ and hence the TD error converges to zero $\delta_k\to 0$ and the parameter estimates converge to their real values $\widehat{\theta}_k^i\to\theta^i$.

Proof. Let write θ_k^i as θ_k. Consider the Lyapunov function

$$W_k = \alpha_1^{-1}\widetilde{\theta}_k^\top \widetilde{\theta}_k. \tag{27}$$

The time difference of the Lyapunov equation $\Delta W_k = W_{k+1} - W_k$ along the TD error trajectories (25) is

$$\begin{aligned}\Delta W_k &= \widetilde{\theta}_k^\top \alpha_1^{-1}\widetilde{\theta}_k - 2\widetilde{\theta}_k^\top \Phi_k \Phi_k^\top \widetilde{\theta}_k \\ &\quad + \alpha_1 \widetilde{\theta}_k^\top \Phi_k \Phi_k^\top \Phi_k \Phi_k^\top \widetilde{\theta}_k - \widetilde{\theta}_k^\top \alpha_1^{-1}\widetilde{\theta}_k \\ &= -\widetilde{\theta}_k^\top (2\Phi_k\Phi_k^\top - \alpha_1 \Phi_k \Phi_k^\top \Phi_k \Phi_k^\top)\widetilde{\theta}_k \\ &\leq -\lambda_{\min}(\Upsilon_k)\|\widetilde{\theta}_k\|^2,\end{aligned}$$

where $\Upsilon_k = 2\Phi_k\Phi_k^\top - \alpha_1 \Phi_k \Phi_k^\top \Phi_k \Phi_k^\top$. If the learning rate α_1 satisfies (26), then $\Upsilon_k > 0$ which guarantees the negativeness of ΔW_k and the estimation error converges asymptotically to zero, that is, $\widetilde{\theta}_k \to 0$ and hence $\widehat{\theta}_k \to \theta$. From (25) we have that

$$\begin{aligned}\widetilde{\theta}_1 &= (I - \alpha_1 \Phi_0 \Phi_0^\top)\widetilde{\theta}_0 \\ \widetilde{\theta}_2 &= (I - \alpha_1 \Phi_1 \Phi_1^\top)\widetilde{\theta}_1 = (I - \alpha_1 \Phi_1 \Phi_1^\top)(I - \alpha_1 \Phi_0 \Phi_0^\top)\widetilde{\theta}_0 \\ \vdots &= \qquad \vdots \\ \widetilde{\theta}_k &= \left[\prod_{i=0}^{k-1}(I - \alpha_1 \Phi_{k-1-i}\Phi_{k-1-i}^\top)\right]\widetilde{\theta}_0.\end{aligned}$$

Substituting the learning rate (26) in the above equality gives

$$\widetilde{\theta}_k = \left[\prod_{i=0}^{k-1}\left(I - \alpha_0 \frac{\Phi_{k-1-i}\Phi_{k-1-i}^\top}{\Phi_{k-1-i}^\top \Phi_{k-1-i} + 1}\right)\right]\widetilde{\theta}_0.$$

Notice that the term in parenthesis is always less than 1, such that we can define $\gamma_i = I - \alpha_0 \frac{\Phi_{k-1-i}\Phi_{k-1-i}^\top}{\Phi_{k-1-i}^\top \Phi_{k-1-i}+1} \leq \gamma$ for some $\gamma \in (0, 1]$ and hence

$$\widetilde{\theta}_k \leq \prod_{i=0}^{k-1}\gamma_i \widetilde{\theta}_0 \leq \gamma^k \widetilde{\theta}_0.$$

The above inequality shows that the estimation error exponentially converges to zero as $k \to \infty$. A simpler notation is given by

$$\widetilde{\theta}_k = \exp^{-\lambda k}\widetilde{\theta}_0,$$

where $\lambda = -\ln(\gamma)$. This completes the proof. □

Recall that θ^i is a vectorization of matrix M and thus, $\widehat{\theta}_k^i$ is a vectorization of the matrix $\widehat{M}_k^i$. Then, the near optimal neocortex control policy is computed with the stationary condition $\widehat{u}_k^{i*} = \frac{\partial \widehat{Q}(x_k, u_k^i)}{\partial u_k^i} = 0$, that is,

$$\widehat{u}_k^{i*} = -\widehat{K}_k^i x_k = -\widehat{Q}_{uu}^{-1}\widehat{Q}_{xu}^\top x_k, \tag{28}$$

where $\widehat{K}_k^i \in \mathbb{R}^{m\times n}$ is an estimate of the control gain K^i. The connection between $\widehat{M}_k^i$ and $\widehat{K}_k^i$ is given by [47]

$$\widehat{P}_k^i = \begin{bmatrix} I \\ -\widehat{K}_k^i \end{bmatrix}^\top \widehat{M}_k^i \begin{bmatrix} I \\ -\widehat{K}_k^i \end{bmatrix}, \tag{29}$$

where $\widehat{P}_k^i \in \mathbb{R}^{n\times n}$ is an estimate of P^i.

4.2. Gradient identification rule

The control gains computed in (8) and (17) show a nonlinear relation. So, it is mandatory to estimate the matrices of system (1). This issue is solved by using a simple gradient identification technique as

$$\hat{x}_{k+1} = \widehat{A}_k x_k + \widehat{B}_k u_k^i = \varphi_k^\top \hat{\vartheta}_k, \tag{30}$$

where $\varphi_k = \varphi(x_k, u_k^i) \in \mathbb{R}^{p\times n}$ is a matrix whose elements are function of x_k and u_k^i, and $\hat{\vartheta}_k \in \mathbb{R}^p$ is vector of parameter estimates constructed with the estimates $\widehat{A}_k \in \mathbb{R}^{n\times n}$ and $\widehat{B}_k \in \mathbb{R}^{n\times m}$ of A and B, respectively. The state approximation error is given by

$$\widetilde{x}_k = \hat{x}_{k+1} - x_{k+1}. \tag{31}$$

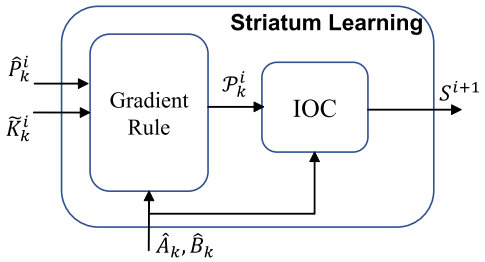


Fig. 3. Striatum learning scheme.

Here, we want to minimize the following index

$$J_1 = \frac{1}{2}\widetilde{x}_k^\top \widetilde{x}_k. \tag{32}$$

The following rule is used to estimate $\hat{\vartheta}_k$ as

$$\hat{\vartheta}_{k+1} = \hat{\vartheta}_k - \alpha_2 \varphi_k \widetilde{x}_k \tag{33}$$

where $\alpha_2 \in (0, 1]$ is a learning rate. The convergence of the rule (33) is similar to the presented in Theorem 1.

5. Striatum learning system

Fig. 3 depicts the striatum block scheme. The scheme is composed of two main systems: a gradient rule that estimates a new kernel matrix $\mathcal{P}^i$ and an inverse optimal control algorithm to estimate the weight matrix S^{i+1} of the reward function.

5.1. Gradient update rule for the kernel matrix

Define the control gain error $\widetilde{K}_k^i \in \mathbb{R}^{m\times n}$ which measures the difference between the hippocampus and neocortex gains as

$$\begin{aligned}\widetilde{K}_k^i &= \hat{K}_e - \hat{K}_k^i \\ &= -\left(\bar{u}\bar{x}^\top(\bar{x}\bar{x}^\top)^{-1} + L^{-1}\hat{B}_k^\top \hat{P}_k^i \hat{A}_k\right),\end{aligned} \tag{34}$$

where $L = L(\hat{P}_k^i) = R + \hat{B}_k^\top \hat{P}_k^i \hat{B}_k$. Here, we want to minimize the gain error to ensure that both the hippocampus and neocortex gains are approximately the same. To achieve this, define the cost index

$$E^i = \mathrm{tr}\{\widetilde{K}_k^{i\top} \widetilde{K}_k^i\}. \tag{35}$$

The following gradient descent rule

$$\mathcal{P}^i = \hat{P}_k^i - \alpha_3 \nabla_P E^i \tag{36}$$

is used to derive an improved kernel matrix $\mathcal{P}^i \in \mathbb{R}^{n\times n}$, where $\alpha_3 \in \mathbb{R}_+$ is a learning rate and $\nabla_P = \frac{\partial}{\partial \hat{P}_k^i}$ defines the gradient respect to matrix $\hat{P}_k^i$. We can observe that

$$\begin{aligned}&\nabla_P\left\{LL^{-1} = I\right\} \\ &\hat{B}_k^\top \hat{B}_k L^{-1} + L\nabla_P L^{-1} = 0 \\ &\nabla_P L^{-1} = -L^{-1}\hat{B}_k^\top \hat{B}_k L^{-1}.\end{aligned}$$

The final update rule for the kernel matrix $\mathcal{P}^i$ is

$$\begin{aligned}\mathcal{P}^i &= \hat{P}_k^i + \alpha_3\Big(\widetilde{K}_k^{i\top} L^{-1}\hat{B}_k^\top(\hat{A}_k - \hat{B}_k\hat{K}_k^i) \\ &\quad + (\hat{A}_k - \hat{B}_k\hat{K}_k^i)^\top \hat{B}_k L^{-1}\widetilde{K}_k^i\Big).\end{aligned} \tag{37}$$

Here the importance of the gradient identification rule (33) is observed since (37) requires prior information of matrices A and B (in this case $\hat{A}_k$ and $\hat{B}_k$).

5.2. Inverse optimal control

The new kernel matrix $\mathcal{P}^i$ is used with the estimates $\widehat{A}_k$ and $\widehat{B}_k$ to compute a new weight matrix S^{i+1} using a DARE [6] as

$$\begin{aligned} S^{i+1} = \ & \mathcal{P}^i - \widehat{A}_k^\top \mathcal{P}^i \widehat{A}_k \\ & + \widehat{A}_k^\top \mathcal{P}^i \widehat{B}_k (R + \widehat{B}_k^\top \mathcal{P}^i \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{P}^i \widehat{A}_k \end{aligned} \tag{38}$$

Theorem 2 provides the weight matrix S^i and gain error $\widetilde{K}_k^i$ convergence using the proposed striatum scheme.

Theorem 2. *The weight matrix S^i converges as the number of iterations i increases with a small enough learning rate α_3. Here, convergence means that $\|S^{i+1} - S^i\| \leq \varepsilon_S$ for a small threshold $\varepsilon_S \in \mathbb{R}_+$. Then, the control gain error $\widetilde{K}_k^i$ converges to zero in the limit.*

Proof. The DARE (38) of the striatum's IOC algorithm is

$$\begin{aligned} S^{i+1} = & \mathcal{P}^i - \widehat{A}_k^\top \mathcal{P}^i \widehat{A}_k \\ & + \widehat{A}_k^\top \mathcal{P}^i \widehat{B}_k (R + \widehat{B}_k^\top \mathcal{P}^i \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{P}^i \widehat{A}_k. \end{aligned} \tag{39}$$

Let $\mathcal{A} = \alpha_3 \nabla_P E^i$. Substituting (36) in (39) gives

$$\begin{aligned} S^{i+1} = & (\widehat{P}_k^i - \mathcal{A}) - \widehat{A}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{A}_k \\ & + \widehat{A}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k (R + \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k)^{-1} \\ & \times \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{A}_k. \end{aligned} \tag{40}$$

The neocortex learning system fulfils the next DARE

$$\begin{aligned} S^{i+1} = & \widehat{P}_k^{i+1} - \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k \\ & + \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k (R + \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k)^{-1} \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k. \end{aligned} \tag{41}$$

Matching (40) and (41) gives

$$\begin{aligned} & \widehat{P}_k^{i+1} - \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k \\ & \quad + \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k (R + \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k)^{-1} \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k \\ & = \widehat{P}_k^i - \widehat{A}_k^\top \widehat{P}_k^i \widehat{A}_k \\ & \quad + \widehat{A}_k^\top \widehat{P}_k^i \widehat{B}_k (R + \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k)^{-1} \widehat{B}_k^\top \widehat{P}_k^i \widehat{A}_k \\ & \quad - \Big(\widehat{A}_k^\top \mathcal{A} \widehat{B}_k (R + \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{A} \widehat{A}_k \\ & \quad + \widehat{A}_k^\top \mathcal{A} \widehat{B}_k (R + \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k)^{-\top} \widehat{B}_k^\top \mathcal{A} \widehat{A}_k \Big) \\ & \quad + \alpha_3 \widehat{A}_k^\top \mathcal{A} (R + \widehat{B}_k^\top (\widehat{P}_k^i - \mathcal{A}) \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{A} \widehat{A}_k. \end{aligned} \tag{42}$$

The gradient rule (36) improves in each iteration i the matrix $\mathcal{P}^i$ in order to minimize the control gain error $\widetilde{K}^i$. That is, $\lim_{i\to\infty} \nabla_P E^i = 0$ or equivalently $\lim_{i\to\infty} \mathcal{P}^i = \widehat{P}_k^i$. Hence, (42) is reduced to

$$\begin{aligned} & \lim_{i\to\infty} \Big(\widehat{P}_k^{i+1} - \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k \\ & \quad + \widehat{A}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k (R + \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{B}_k)^{-1} \widehat{B}_k^\top \widehat{P}_k^{i+1} \widehat{A}_k \Big) \\ & = \lim_{i\to\infty} \Big(\widehat{P}_k^i - \widehat{A}_k^\top \widehat{P}_k^i \widehat{A}_k \\ & \quad + \widehat{A}_k^\top \widehat{P}_k^i \widehat{B}_k (R + \widehat{B}_k^\top \widehat{P}_k^i \widehat{B}_k)^{-1} \widehat{B}_k^\top \widehat{P}_k^i \widehat{A}_k \Big). \end{aligned} \tag{43}$$

The following can be concluded from the above results

$$\lim_{i\to\infty} S^{i+1} = \lim_{i\to\infty} S^i \Rightarrow S^{i+1} = S^i, \tag{44}$$

and hence $\lim_{i\to\infty} \widehat{P}_k^{i+1} = \widehat{P}_k^i$. This completes the proof. □

In addition, the following theorem establishes the existence of multiple solutions for S^{i+1} and $\widehat{P}^i$ that satisfy the same hippocampus performance $\widehat{K}^i = \widehat{K}_e$.

Theorem 3. *Let the hippocampus gain $\widehat{K}_e$ matches with the expert's gain K_e. Assume that S^i and $\widehat{P}^i$ converge to some matrices S^* and P^* and verify the DARE*

$$S^i - \widehat{P}^i + A^\top \widehat{P}^i A - A^\top \widehat{P}^i B K_e = 0 \tag{45}$$

Then, any $S^* = S_e + S^i$ and $P^* = P_e + \widehat{P}^i$ satisfy the DARE (6) and consequently obtains the same control gain K_e.

Proof. The DARE (6) is written in terms of S^* and P^* as

$$\begin{aligned} &S_e - P_e + A^\top P_e A - A^\top P_e B(R + B^\top P_e B)^{-1} B^\top P_e A \\ &\quad = S^* - P^* + A^\top P^* A - A^\top P^* B K_e \\ &\qquad - S^i + \mathcal{P}^i - A^\top \mathcal{P}^i A + A^\top \mathcal{P}^i B K_e = 0. \end{aligned} \tag{46}$$

Recall that in this paper $R = R_e$. Since S^i and $\mathcal{P}^i$ satisfy the DARE (45) implies that

$$S^* - P^* + A^\top P^* A - A^\top P^* B K_e = 0 \tag{47}$$

The equality (47) is valid when

$$A^\top P^* B = K_e^\top (R + B^\top P^* B)$$

$$B^\top P^* A = (R + B^\top P^* B) K_e$$

which means that K_e can be obtained as

$$K_e = (R + B^\top P^* B)^{-1} B^\top P^* A = K^* \tag{48}$$

where $P^* \neq P_e$ when $\mathcal{P}^i \neq 0$ and $K^* \in \mathbb{R}^{m \times n}$ is the control gain obtained from matrices S^* and P^*. Hence, there exist multiple kernel matrices P^* which yields the same gain matrix K_e and in consequence S^* is not unique.

From the above results, we have that

$$(R + B^\top P^* B) K_e = B^\top P^* A \tag{49}$$

$$(R + B^\top P_e B) K_e = B^\top (P^* - \mathcal{P}^i) A \tag{50}$$

The difference between (49) and (50) gives

$$\begin{aligned} B^\top (P^* - P_e) B K_e &= B^\top \mathcal{P}^i A \\ B^\top \mathcal{P}^i B K_e &= B^\top \mathcal{P}^i A \end{aligned} \tag{51}$$

which means that we are able to compute K_e without R, that is,

$$K_e = (B^\top \mathcal{P}^i B)^{-1} B^\top \mathcal{P}^i A. \tag{52}$$

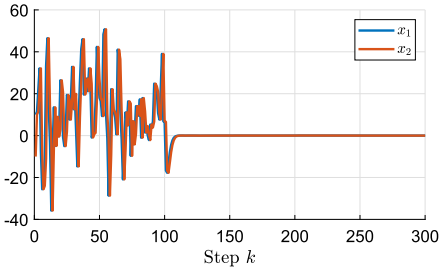
So, the DARE (46) can be written as

$$\begin{aligned} S^i &= S^* - S_e \\ &= (P^* - P_e) - A^\top (P^* - P_e) A + A^\top P^* B K_e \\ &\quad - A^\top P_e B K_e \\ &= \mathcal{P}^i - A^\top \mathcal{P}^i A + A^\top P^* B (R + B^\top P^* B)^{-1} B^\top P^* A \\ &\quad - A^\top P_e B (R + B^\top P_e B)^{-1} B^\top P_e A. \end{aligned} \tag{53}$$

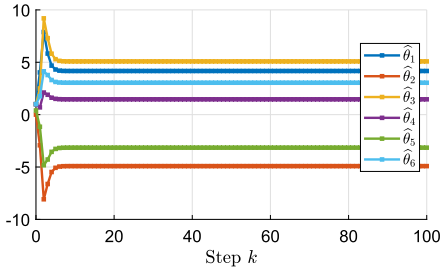
Therefore, from (53) and (51) allows to conclude that S^i has multiple solutions which are not necessarily positive definite due to the cancellation of matrix R. In consequence, the positivity of the kernel matrix $\mathcal{P}^i$ can be violated which can produce unstable closed-loop trajectories. To solve this issue, a constraint that relates the values of the weight matrices S^i and R has to be added to guarantee positive definite and unique solutions. Furthermore, this proof assumes exact knowledge of matrix A and B to compute S^i. However, the estimates $\widehat{A}_k$ and $\widehat{B}_k$ may introduce a small modelling error to matrix S^i if the regressor matrix $\varphi(x,u)$ is not PE. This completes the proof. □

6. Simulation studies

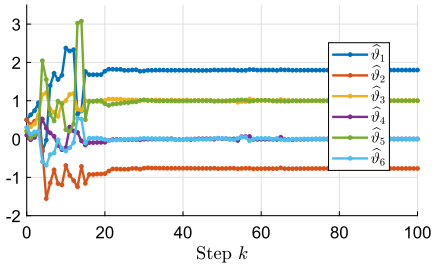
Two systems, an unstable system and a power system, proposed in [17] are used to test the proposed approach.



(a) States evolution x_k



(b) Q-learning algorithm results



(c) Identification algorithm results

Fig. 4. Neocortex results.

6.1. *Unstable linear system*

Consider a linear system (1) with the following matrices

$$A = \begin{bmatrix} 1.8 & -0.77 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The above system is unstable since the eigenvalues of matrix A are $\lambda(A) = \{1.1, 0.7\}$. However, the system is controllable. Consider that an expert user designs an optimal controller using the following weight matrices

$$S_e = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0.25 \end{bmatrix}, \quad R_e = 1.$$

The DARE (6) is used to compute P_e and K_e as

$$P_e = \begin{bmatrix} 2.0467 & -1.1424 \\ -1.1424 & 0.6483 \end{bmatrix},$$

$$K_e = \begin{bmatrix} 0.8342 & -0.5173 \end{bmatrix}.$$

The states and input trajectories under the expert's policy are collected in the vectors $\bar{x}$ and $\bar{u}$. These data are corrupted by a small uniformly distributed random number $\eta \in (0, 0.1)$ to simulate sensor noise. These vectors have 1000 data points in each dimension. The LS rule (9) is used to estimate the expert's control gain from the stored data. The gain estimate $\widehat{K}_e$ is

$$\widehat{K}_e = \begin{bmatrix} 0.8248 & -0.5202 \end{bmatrix}.$$

The estimation error $\widetilde{K}_e = \|K_e - \widehat{K}_e\|$ between the expert's control gain K_e and the hippocampus gain $\widehat{K}_e$ is $\widetilde{K}_e = 0.0098$ which is equivalently to a 0.98% of estimation error. This error is practically negligible but can be reduced by means of a low pass filter or a matrix approximation based on a singular-value decomposition algorithm. Here it is clear that the hippocampus control gain $\widehat{K}_e$ can differ from the real expert's gain K_e in presence of measurement noise which may produce different control performances.

The Q-learning and the identification algorithms of the neocortex learning system are trained with the following learning rates: $\alpha_1 = 0.01$ and $\alpha_2 = 0.003$. These learning rates exhibit the best performance in the training phase. The weights of the reward function are initialized as $S^0 = I_2$ and $R = 1$. The following PE dithering noise [48] is added to the control input u_k in the first 100 steps: $\Delta u_k = 34(\sin^2(100k)\cos(100k) + \sin^2(2k)\cos(0.1k) + \sin^2(-1.2k)\cos(0.5k) + \sin^5(k) + \sin^2(1.12k) + \cos(2.4k)\sin^3(2.4k))$. The amplitude and frequencies of the PE signal were manually tuned until the best estimation performance was achieved.

The obtained results of the neocortex learning system are shown in Fig. 4. The results exhibit the stabilization of all the states of the proposed unstable system. In addition, both the parameters of the Q-learning and identification algorithms converge to their real values in accordance to the kernel matrix $\widehat{P}_k^i$. Here, the estimates of the identification algorithm are

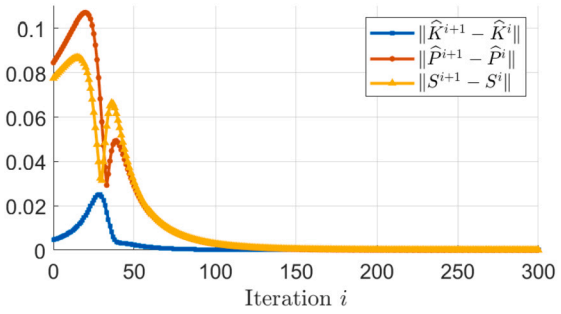


Fig. 5. Striatum learning results of Case 1.

$$\widehat{A}_k = \begin{bmatrix} 1.799 & -0.7696 \\ 1.0002 & -0.0001 \end{bmatrix}, \quad \widehat{B}_k = \begin{bmatrix} 0.999 \\ 0.0002 \end{bmatrix}.$$

The norm of the identification error is used to evaluate the obtained estimates. The results are $\|\widetilde{\vartheta}_k\| = 0.0015$ which is equivalently to the 0.15% of identification error.

The striatum learning system uses a learning rate of $\alpha_3 = 0.9$ to obtain the new kernel matrix $\mathcal{P}^i$. The complementary results are shown in Fig. 5.

The obtained matrices are

$$\widehat{P}^i = \begin{bmatrix} 2.0825 & -1.2060 \\ -1.2060 & 1.8346 \end{bmatrix},$$
$$\widehat{K}^i = \begin{bmatrix} 0.8248 & -0.5202 \end{bmatrix},$$
$$S^i = \begin{bmatrix} -0.0607 & -0.5709 \\ -0.5709 & 1.4341 \end{bmatrix}.$$

Notice that a negative definite matrix S^i is obtained which violates its initial definition (positive semi-definite matrix). The main reason of this weird result is because the initial weight matrix S^0 is greater than the expert's weight matrix S_e, that is, the one-step gradient rule updates $\mathcal{P}^i$ in the direction where the control gain error $\widetilde{K}^i$ decreases; however, there is a point where the gradient update rule cannot decrease the gain error (due to the dynamics of the system and the proposed learning rate α_3) to guarantee a positive definite kernel matrix $\mathcal{P}^i$. Hence, the update rule modifies the direction of the gradient such that the weight matrix S^i (obtained from the IOC algorithm) can take negative values to guarantee convergence of $\widetilde{K}^i$ to zero. This fact can be seen in iteration 40 of Fig. 5.

To notice more this fact, we use another expert's weight matrix denoted by S_e^2 that is bigger than the initial weight matrix $S^{2^0} = I_2$ as

$$S_e^2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

The new expert's kernel matrix and control gain under the new kernel matrix S_e^2 are

$$P_e^2 = \begin{bmatrix} 5.619 & -1.9498 \\ -1.9498 & 2.5033 \end{bmatrix},$$
$$K_e^2 = \begin{bmatrix} 1.2335 & -0.6537 \end{bmatrix}.$$

Assume that the hippocampus gain $\widehat{K}_e^2$ was equal to K_e^2. The same learning rates are used in this example. The complementary results for the control gain $\widehat{K}^{2^i}$, $\widehat{P}^{2^i}$, and S^{2^i} are shown in Fig. 6.

The matrices converge to the following values

$$\widehat{P}^{2^i} = \begin{bmatrix} 5.6208 & -1.9507 \\ -1.9507 & 1.8751 \end{bmatrix},$$
$$\widehat{K}^{2^i} = \begin{bmatrix} 1.2335 & -0.6537 \end{bmatrix},$$
$$S^{2^i} = \begin{bmatrix} 2.6305 & -1.0009 \\ -1.0009 & 1.3718 \end{bmatrix}.$$

The above results show that the weight matrix S^i is positive definite since the initial weight matrix S^0 was less than the expert's weight matrix S_e. Notice that for a small gain error $\widetilde{K}^i$ the updates of the gradient descent algorithm are small and hence it requires more iterations to converge.

It is important to observe that the final weight matrix S^i is completely different to matrix S_e^2. However, both matrices exhibit the same control gain $K_e^2 = \widehat{K}^i$. This is because there are different weight matrices S^i that yields to the same gain $\widehat{K}_e$. Additional

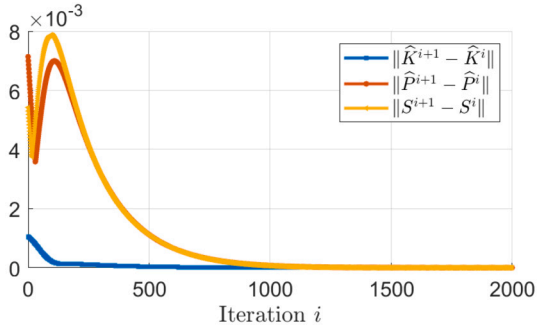


Fig. 6. Striatum learning results of Case 2.

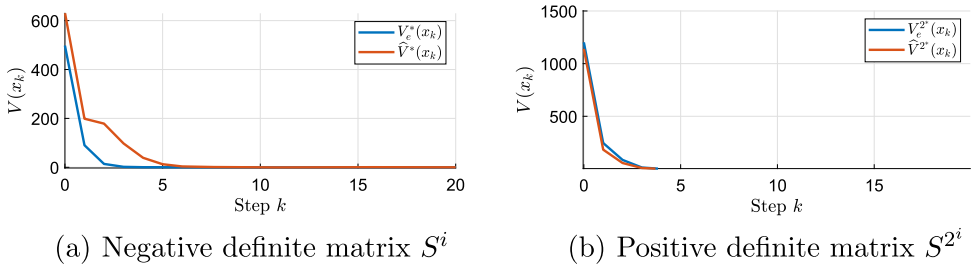


Fig. 7. Optimal value function trajectories.

constraints can be added to the IOC algorithm to reduce the number of possible solutions or an unique solution, that is, the real expert's weight matrix. One simple constraint is to incorporate the values of the reward function as an additional expert's feature.

Two different interpretations can be distinguished as a result of using different matrices S^i. The first one lies in the use of negative definite weight matrix which can achieve the same hippocampus control policy but the performance of optimal value function is degraded. Conversely, the second interpretation lies in an enhancement of the optimal value function. Fig. 7 exhibits these two interpretations.

The performance of the optimal value function is degraded when a negative definite weight matrix is used. Conversely, the second case shows an enhancement in the convergence rate of the optimal value function. Numerically, we have that $\|V_e^*\| = 506.2935$, $\|\widehat{V}^*\| = 695.1129$, $\|V_e^{2^*}\| = 1230.2$, and $\|\widehat{V}^{2^*}\| = 1155.8$.

6.2. Power system

We further test the proposed approach in a high order power system for the load frequency control of an electric system [17]. The discrete-time plant is

$$A = \begin{bmatrix} 0.9616 & 1.0047 & 0.0867 & -0.0450 \\ -0.0739 & 0.7490 & 0.1154 & -0.1038 \\ -0.5354 & -0.3401 & 0.2303 & -0.7378 \\ 0.0593 & 0.0316 & 0.002 & 0.9993 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.0450 & 0.1038 & 0.7378 & 0.0007 \end{bmatrix}^{\top}.$$

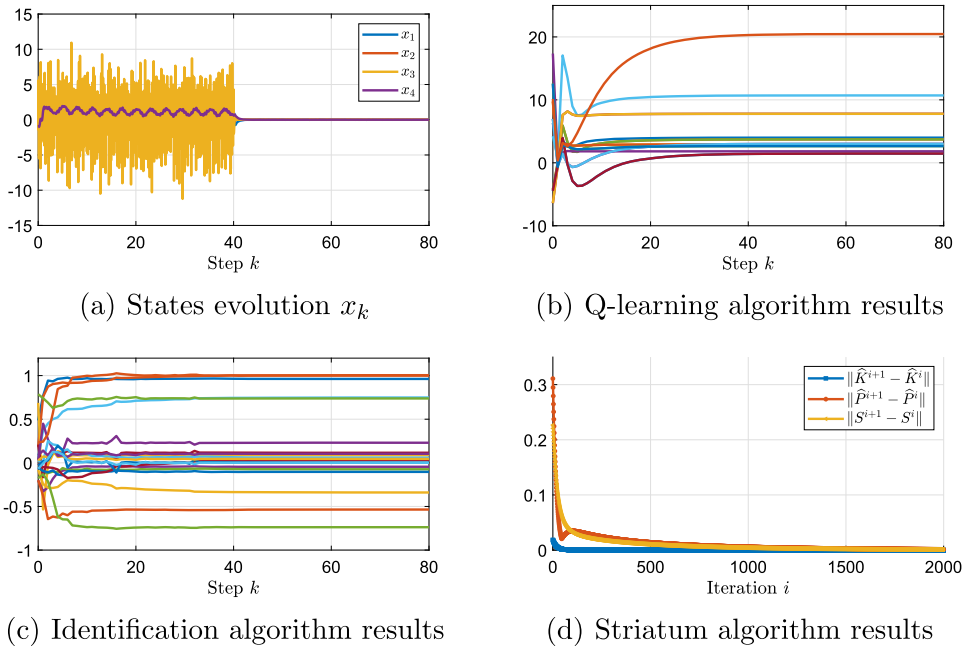
The expert's weight matrices are set to $S_e = 2I_4$ and $R_e = 1$. The expert's kernel matrix and control gain are

$$P_e = \begin{bmatrix} 9.7697 & 13.2744 & 1.2163 & 8.4652 \\ 13.2744 & 35.3318 & 3.7143 & 11.4238 \\ 1.2163 & 3.7143 & 2.4953 & 0.8941 \\ 8.4652 & 11.4238 & 0.8941 & 43.4822 \end{bmatrix},$$

$$K_e = \begin{bmatrix} 0.2857 & 2.0596 & 0.4455 & -0.0789 \end{bmatrix}.$$

Sensor noise in the states measurements is modelled as a small random noise $\eta \in (0, 0.1)$. The estimated hippocampus gain is

$$\widehat{K}_e = \begin{bmatrix} 0.2857 & 2.0596 & 0.4455 & -0.0789 \end{bmatrix}.$$



(a) States evolution x_k (b) Q-learning algorithm results

(c) Identification algorithm results (d) Striatum algorithm results

Fig. 8. Complementary learning results.

The learning rates are manually tuned to ensure stability and fast convergence results. The learning rates are set to: $\alpha_1 = 0.01$, $\alpha_2 = 0.02$, and $\alpha_3 = 0.9$. A duffing system proposed in [49] is used as PE signal to ensure convergence of the parameter estimates. The initial weight matrix is set to $S^0 = I_4$. Fig. 8 exhibits the results of each learning system. The estimation results are

$$\widehat{A}_k = \begin{bmatrix} 0.9619 & 1.0027 & 0.0869 & -0.0448 \\ -0.0737 & 0.7473 & 0.1156 & -0.1036 \\ -0.5355 & -0.3390 & 0.2302 & -0.7379 \\ 0.0594 & 0.0307 & 0.0021 & 0.9994 \end{bmatrix},$$

$$\widehat{B}_k = \begin{bmatrix} 0.0449 & 0.1037 & 0.7379 & 0.0006 \end{bmatrix}^\top,$$

$$\widehat{P}^i = \begin{bmatrix} 3.9844 & 3.6951 & 2.9187 & 3.0055 \\ 3.6951 & 10.6332 & 7.7769 & 1.4440 \\ 2.9187 & 7.7769 & 1.8147 & 2.6496 \\ 3.0055 & 1.4440 & 2.6496 & 20.4464 \end{bmatrix},$$

$$\widehat{K}^i = \begin{bmatrix} 0.2857 & 2.0596 & 0.4455 & -0.0789 \end{bmatrix},$$

$$S^i = \begin{bmatrix} 2.6917 & 4.9724 & 2.9996 & 1.3580 \\ 4.9724 & 15.5964 & 7.5602 & 4.3006 \\ 2.9996 & 7.5602 & 1.6399 & 3.0211 \\ 1.3580 & 4.3006 & 3.0211 & 1.9976 \end{bmatrix}.$$

The obtained matrices show similar results to the previous example. However, both the kernel matrix $\widehat{P}^i$ and S^i are negative definite matrices. Fig. 9 shows the optimal value function trajectories under the expert's and complementary learning kernel matrices.

6.3. Discussion

Recent works in reward inference are mainly applied for continuous-time linear systems with a quadratic reward structure [23]. For discrete-time systems, the approaches that are adopted in the literature consist in model-based IOC or reinforcement learning architectures based on binary reward functions. We do not provide comparisons with those approaches because: i) binary reward functions [22] do not exhibit an optimal performance and, in most cases, the control policy is discontinuous which can destabilize the closed-loop trajectories, and ii) IOC algorithms [21] do not show difference with the results provided in this manuscript since the identification algorithm estimates accurately the parameters of both the unstable and power systems. Here, the proposed approach aims to fill the current gap in inferring the reward function of discrete-time linear systems with unknown dynamics.

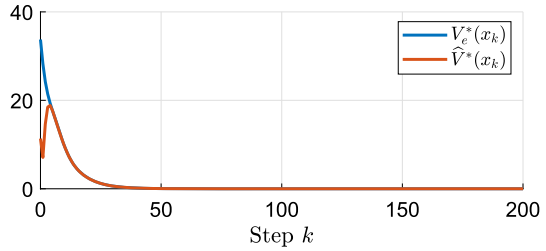


Fig. 9. Optimal value function trajectories.

In view of the results of previous sections we can identify the main advantages and areas of opportunity of the proposed approach. The main advantages of the proposed algorithm are: 1) it can infer a family of reward functions associated to the same hippocampus expert's performance, 2) the persistence of excitation ensures convergence of the Q-learning and identification algorithms simultaneously, and 3) the weight matrices of the reward function can be negative definite and open a gap to investigate the effect of this kind of matrices that can produce stable control policies. The main disadvantages and areas of opportunity lie in: a) three hyperparameters (learning rates) are required to tune for the Q-learning, identification, and gradient algorithms, b) convergence of the neocortex algorithms require the fulfilment of a PE condition, and c) biased estimates of the identification algorithm may lead to solutions that can destabilize the closed-loop trajectories. Further work will work on these disadvantages to increase the impact of the proposed technique.

7. Conclusions

In this paper, the design of a reward function inference algorithm of expert's data is discussed. The proposed approach is modelled by a complementary learning algorithm that relates three main co-dependent learning systems: the striatum, neocortex, and the hippocampus. Whilst the hippocampus is modelled by expert's knowledge and a PE signal for fast learning, the neocortex is designed as a Q-learning and identification algorithms which exhibit good pattern association. The striatum is designed as an IOC algorithm to infer the weight matrix of the expert's reward function. Simulation studies verify the proposed approach with informative results.

Future research includes the incorporation of constraints in the IOC problem and seeks for alternative solutions to relate the neocortex and hippocampus to infer effectively the expert's reward function.

CRediT authorship contribution statement

Adolfo Perrusquía: Conception and design of study, acquisition of data,analysis and/or interpretation of data, drafting the manuscript, revising the manuscript critically for important intellectual content, approval of the version of the manuscript to be published.

Weisi Guo: Conception and design of study, analysis and/or interpretation of data, drafting the manuscript, revising the manuscript critically for important intellectual content, approval of the version of the manuscript to be published.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] F.L. Lewis, Optimal Control, Wiley, New York, NY, USA, 2012.

[2] J.-H. Kim, F. Lewis, Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI, Automatica 46 (2010) 1320–1326, https://doi.org/10.1016/j.automatica.2010.05.002.

[3] A. Perrusquía, W. Yu, Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: an overview, Neurocomputing 438 (2021) 145–154, https://doi.org/10.1016/j.neucom.2021.01.096.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533, https://doi.org/10.1038/nature14236.

[5] J. Mendoza, A. Perrusquía, J.A. Flores-Campos, Mechanical advantage assurance control of quick-return mechanisms in task space, in: 2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), IEEE, 2022, pp. 1–6.

[6] F.L. Lewis, D. Vrabie, K.G. Vamvoudakis, Reinforcement learning and feedback control using natural decision methods to design optimal adaptive controllers, IEEE Conrol Syst. Mag. 32 (6) (2012) 76–105, https://doi.org/10.1109/MCS.2012.2214134.

[7] I. Grondman, L. Buşoniu, G.A. Lopes, R. Babŭska, A survey of actor-critic reinforcement learning: standard and natural policy gradients, IEEE Trans. Syst. Man Cybern., Part C 42 (6) (2012) 1291–1307, https://doi.org/10.1109/TSMCC.2012.2218595.
[8] B. Kiumarsi, K.G. Vamvoudakis, H. Modares, F.L. Lewis, Optimal and autonomous control using reinforcement learning: a survey, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2042–2062, https://doi.org/10.1109/TNNLS.2017.2773458.
[9] Q. Xie, B. Luo, F. Tan, Discrete-time lqr optimal tracking control problems using approximate dynamic programming algorithm with disturbance, in: 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), IEEE, 2013, pp. 716–721.
[10] A. Perrusquía, W. Yu, Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–11, 10.1109/TNNLS.2020.3026010.
[11] R. Kamalapurkar, P. Walters, W. Dixon, Model-based reinforcement learning for approximate optimal regulation, Automatica 64 (2016) 94–104, https://doi.org/10.1016/j.automatica.2015.10.039.
[12] A. Perrusquía, W. Yu, X. Li, Multi-agent reinforcement learning for redundant robot control in task-space, Int. J. Mach. Learn. Cybern. 12 (1) (2021) 231–241, https://doi.org/10.1007/s13042-020-01167-7.
[13] B. Kiumarsi, F.L. Lewis, Actor-critic based optimal tracking for partially unknown nonlinear discrete- time systems, IEEE Trans. Neural Netw. Learn. Syst. 26 (1) (2015) 140–151, https://doi.org/10.1109/TNNLS.2014.2358227.
[14] A. Perrusquía, Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning, Inf. Sci. 595 (2022) 364–377.
[15] B. Kiumarsi, F.L. Lewis, H. Modares, A. Karimpor, M.-B. Naghibi-Sistani, Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics, Automatica 50 (2014) 1167–1175, https://doi.org/10.1016/j.automatica.2014.02.015.
[16] K.G. Vamvoudakis, Q-learning for continuous-time linear systems: a model-free infinite horizon optimal control approach, Syst. Control Lett. 100 (2017) 14–20, https://doi.org/10.1016/j.sysconle.2016.12.003.
[17] S.A.A. Rizvi, Z. Lin, Output feedback Q-learning control for the discrete-time linear quadratic regulator problem, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2018) 1523–1536, https://doi.org/10.1109/TNNLS.2018.2870075.
[18] A. Perrusquía, W. Yu, Human-behavior learning for infinite-horizon optimal tracking problems of robot manipulators, in: 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 57–62.
[19] A. Perrusquía, W. Yu, Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning, Int. J. Robust Nonlinear Control 30 (7) (2020) 2920–2936, https://doi.org/10.1002/rnc.4911.
[20] P. Abbeel, A.Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proceedings of the Twenty-First International Conference on Machine Learning, 2004.
[21] Y. Park, Inverse optimal and robust nonlinear attitude control of rigid spacecraft, Aerosp. Sci. Technol. 28 (1) (2013) 257–265, https://doi.org/10.1016/j.ast.2012.11.006.
[22] A.Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 663–670.
[23] N. Ab Azar, A. Shahmansoorian, M. Davoudi, From inverse optimal control to inverse reinforcement learning: a historical review, Annu. Rev. Control 50 (2020) 119–138, https://doi.org/10.1016/j.arcontrol.2020.06.001.
[24] Y. Pan, Multiple knowledge representation of artificial intelligence, Engineering 6 (3) (2020) 216–217.
[25] Y. Yang, Y. Zhuang, Y. Pan, Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies, Front. Inf. Technol. Electron. Eng. 22 (12) (2021) 1551–1558.
[26] A. Perrusquía, W. Yu, X. Li, Nonlinear control using human behavior learning, Inf. Sci. 569 (2021) 358–375, https://doi.org/10.1016/j.ins.2021.03.043.
[27] R.C. O'Reilly, R. Bhattacharyya, M.D. Howard, N. Ketz, Complementary learning systems, Cogn. Sci. 38 (6) (2014) 1229–1248, https://doi.org/10.1111/j.1551-6709.2011.01214.x.
[28] S. Blakeman, D. Mareschal, A complementary learning systems approach to temporal difference learning, Neural Netw. 122 (2020) 218–230, https://doi.org/10.1016/j.neunet.2019.10.011.
[29] M.G. Mattar, N.D. Daw, Prioritized memory access explains planning and hippocampal replay, Nat. Neurosci. 21 (11) (2018) 1609–1617, https://doi.org/10.1038/s41593-018-0232-z.
[30] A. Perrusquía, W. Guo, Hippocampus experience inference for safety critical control of unknown multi-agent linear systems, ISA Trans. (2022).
[31] K.L. Stachenfeld, M.M. Botvinick, S.J. Gershman, The hippocampus as a predictive map, Nat. Neurosci. 20 (11) (2017) 1643–1653, https://doi.org/10.1038/nn.4650.
[32] H.F. Ólafsdóttir, D. Bush, C. Barry, The role of hippocampal replay in memory and planning, Curr. Biol. 28 (1) (2018) R37–R50, https://doi.org/10.1016/j.cub.2017.10.073.
[33] A. Vilà-Balló, E. Mas-Herrero, P. Ripollés, M. Simó, J. Miró, D. Cucurell, D. López-Barroso, M. Juncadella, J. Marco-Pallarés, M. Falip, et al., Unraveling the role of the hippocampus in reversal learning, J. Neurosci. 37 (28) (2017) 6686–6697, https://doi.org/10.1523/JNEUROSCI.3212-16.2017.
[34] A. Perrusquía, Human-behavior learning: a new complementary learning perspective for optimal decision making controllers, Neurocomputing 489 (7) (2022) 157–166.
[35] L. Buşoniu, R. Babuška, B. De Schutter, D. Ernst, Reinforcement Learning and Dynamic Programming Using Function Approximators, CRC Press, 2010.
[36] A. Perrusquía, W. Guo, Optimal control of nonlinear systems using experience inference human-behavior learning, IEEE/CAA J. Autom. Sin. 10 (1) (2023) 90–102.
[37] D. Kumaran, D. Hassabis, J.L. McClelland, What learning systems do intelligent agents need? Complementary learning systems theory updated, Trends Cogn. Sci. 20 (7) (2016) 512–534, https://doi.org/10.1016/j.tics.2016.05.004.
[38] J.L. McClelland, B.L. McNaughton, R.C. O'Reilly, Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory, Psychol. Rev. 102 (3) (1995) 419, https://doi.org/10.1037/0033-295X.102.3.419.
[39] A. Perrusquía, A complementary learning approach for expertise transference of human-optimized controllers, Neural Netw. 145 (2022) 33–41, https://doi.org/10.1016/j.neunet.2021.10.009.
[40] H. El-Hussieny, J.-H. Ryu, Inverse discounted-based LQR algorithm for learning human movement behaviors, Appl. Intell. 49 (4) (2019) 1489–1501, https://doi.org/10.1007/s10489-018-1331-y.
[41] A. Perrusquía, W. Yu, Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning, IEEE Trans. Cybern. 52 (6) (2022) 4485–4494, https://doi.org/10.1109/TCYB.2020.3028988.
[42] J. Ramírez, W. Yu, A. Perrusquía, Model-free reinforcement learning from expert demonstrations: a survey, Artif. Intell. Rev. 55 (4) (2022) 3213–3241, https://doi.org/10.1007/s10462-021-10085-1.
[43] X. Xie, C. Wei, Z. Gu, K. Shi, Relaxed resilient fuzzy stabilization of discrete-time Takagi-Sugeno systems via a higher order time-variant balanced matrix method, IEEE Trans. Fuzzy Syst. 30 (11) (2022), https://doi.org/10.1109/TFUZZ.2022.3145809.
[44] W. Yu, A. Perrusquía, Simplified stable admittance control using end-effector orientations, Int. J. Soc. Robot. 12 (5) (2020) 1061–1073, https://doi.org/10.1007/s12369-019-00579-y.
[45] A. Al-Tamimi, F. Lewis, M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 38 (4) (2008) 943–949, https://doi.org/10.1109/ADPRL.2007.368167.
[46] F.L. Lewis, S. Jagannathan, A. Yeşildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, Taylor & Francis, 1999.
[47] A. Perrusquía, W. Yu, A. Soria, Position/force control of robot manipulators using reinforcement learning, Ind.l Robot, Iint. J. Rob. Res. Appl. 46 (2) (2019) 267–280, https://doi.org/10.1108/IR-10-2018-0209.

[48] F.L. Lewis, K.G. Vamvoudakis, Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 41 (1) (2010) 14–25.

[49] A. Perrusquía, Robust state/output feedback linearization of direct drive robot manipulators: a controllability and observability analysis, Eur. J. Control 64 (100612) (2022), https://doi.org/10.1016/j.ejcon.2021.12.007.

Reward inference of discrete-time expert's controllers: A complementary learning approach

Perrusquía, Adolfo

2023-03-06

Perrusquia A, Guo W. (2023) Reward inference of discrete-time expert’s controllers: a complementary learning approach, Information Sciences. 631, June 2023, pp. 396-411 https://doi.org/10.1016/j.ins.2023.02.079