CODiT 2022: 8th International Conference on Control, Decision and Information Technologies, 17-20 May 2022, Istanbul, Turkey, pp. 165 - 170
DOI: 10.1109/CoDIT55151.2022.9804118

Cost Inference of Discrete-time Linear Quadratic Control Policies using Human-Behaviour Learning

Adolfo Perrusquía and Weisi Guo

***Abstract*— In this paper, a cost inference algorithm for discrete-time systems using human-behaviour learning is proposed. The approach is inspired in the complementary learning that exhibits the neocortex, hippocampus, and striatum learning systems to achieve complex decision making. The main objective is to infer the hidden cost function from expert's data associated to the hippocampus (off-policy data) and transfer it to the neocortex for policy generalization (on-policy data) in different systems and environments. The neocortex is modelled by a Q-learning and a least-squares identification algorithms for on-policy learning and system identification. The cost inference is obtained using a one-step gradient descent rule and an inverse optimal control algorithm. Convergence of the cost inference algorithm is discussed using Lyapunov recursions. Simulations verify the effectiveness of the approach.**

I. Introduction

The increasing field in machine learning and data science has provided a wide amount of expert's data that defines the desired performance of a given system under hidden constraints [1]. In particular, expert's control policies obtained from adaptive dynamic programming (ADP) [2] or reinforcement learning (RL) [3]–[5] algorithms give a solution of a hidden optimization problem [3], that is, these policies minimize a hidden cost or reward function. In this paper, discrete-time linear quadratic control [6] policies [7], [8] are considered for sake of simplicity.

The success of any ADP/RL algorithm [9]–[12] is given by an adequate design of the cost or reward function that is in charge of defining the control task [13]. In most cases, the cost is designed heuristically in accordance to expert's knowledge and it is hidden within the final control policy. However, the control policy generalization is poor for different systems and environments. Therefore, it is necessary to extract the hidden cost of expert's policies to infer the definition of the task to new systems and environments maintaining the expert's desired performance. However, the cost inference becomes a hard problem since the aim of ADP/RL is not to imitate performance, but use experience to facilitate learning and improve the performance of the controller [14], [15].

One of the main challenges for cost inference in discrete-time systems lies in the high nonlinearity of the control policy respect to the system dynamics and the kernel matrix associated to an algebraic Riccati equation [16]. One way to simplify this problem is by assuming prior knowledge of the system dynamics, otherwise identification techniques are required to estimate the parameters of the system [16]. In this paper, the cost inference is elegantly achieved using a complementary learning [17] based on a human-behavior learning approach.

Human-Behaviour Learning [18] is relatively a new learning paradigm inspired by the humans' brain activity for decision making [19], [20]. Three main learning systems can be distinguished at the brain cortex [21]: the hippocampus, the neocortex, and the striatum. These systems merge different sources of knowledge, which are not necessarily independent to each other, coming from experience and system-environment interaction [16] to achieve complex reward-driven behaviour. In this sense, human-behavior learning provides an interesting way to relate experience and current knowledge for decision making.

Whilst the hippocampus relies in fast learning structures associated to memory and experience [22]–[25] such as replay [5], [18], model-based algorithms [7], exploration-exploitation techniques [26], [27], the neocortex contains pattern dependent structures [28] related to online learning models, e.g., ADP/RL algorithms [10], [29]–[31], neural networks [32], [33], function approximators [34], [35]. In fact, the hippocampus gives an adequate direction for the neocortex updating.

On the other hand, the striatum is a brain's structure that evaluates different sources of information (provided by the neocortex and the hippocampus) for decision making [36]. In this context, the striatum has complementary properties [37], [38] to connect the other learning systems to achieve complex behaviour. Furthermore, the striatum can be modelled as a communication channel between different sources of knowledge which takes their advantages to enhance learning and the improvement of the final control policy.

In view of the above, this paper proposes a human-behaviour learning algorithm for cost inference of discrete-time linear quadratic control policies. The proposed approach extracts the cost from two different sources of knowledge (hippocampus and neocortex) given by expert's and online data of the system dynamics and a least-squares Q-learning algorithm [7]. These sources of knowledge are combined together via the striatum learning system which is modelled by an inverse optimal control (IOC) problem. Simulations verify the proposed approach under a high-order power system.

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 programme.

Adolfo Perrusquía and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL Bedford, UK. Adolfo.Perrusquia-Guzman@cranfield.ac.uk; weisi.guo@cranfield.ac.uk.

II. Hippocampus Expert Control Policy

First we need to define how a linear quadratic control policy is modelled. The next discrete-time linear system structure [39] is considered

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

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times sm}$ define the unknown coupling matrices, $k \in \mathbb{N}$ denotes the time step index.

The control policy of the linear quadratic control problem is expressed by $u_k^* = -Kx_k$, for some stabilizing gain $K \in \mathbb{R}^{m\times n}$, that minimizes the next value function in an infinite horizon

$$V(x_k) = \sum_{i=k}^{\infty} r(x_i, u_i), \tag{2}$$

where $r(\cdot)$ is the cost or reward function. Typically the cost function for a LQR controller has a quadratic structure [40] in the control input (the quadratic term in the state is optional), i.e., $r(x_k, u_k) = x_k^\top S x_k + u_k^\top R u_k$, where $S = S^\top \geq 0 \in \mathbb{R}^{n\times n}$ and $R = R^\top > 0 \in \mathbb{R}^{m\times m}$ are positive semidefinite and definite weight matrices of the cost. The optimal control policy of the linear quadratic problem is

$$u_k^* = -Kx_k = -(R + B^\top P B)^{-1} B^\top P A x_k, \tag{3}$$

where $P = P^\top > 0 \in \mathbb{R}^{n\times n}$ is the kernel matrix solution of the following discrete algebraic Riccati equation (DARE) [32]

$$A^\top P A + S + A^\top P B (R + B^\top P B)^{-1} B^\top P A + S - P = 0. \tag{4}$$

In the sequel of the paper the control policies satisfy (3). Assume that there exists expert's data stored in the vectors $\bar{x} = [x_0^e, \cdots, x_{l-1}^e] \in \mathbb{R}^{n\times l}$ and $\bar{u} = [u_0^e, \cdots, u_{l-1}^e] \in \mathbb{R}^{m\times l}$, where $x_i^e \in \mathbb{R}^n$ and $u_i^e \in \mathbb{R}^m$ are the states and control input of expert's trajectories, with $i = 0, ..., l$ and l is the number of data points. The data satisfy the dynamic equation

$$x_{k+1}^e = Ax_k^e + Bu_k^e. \tag{5}$$

These data are collected from an expert control policy $u_k^{e^*} = -K_e x_k^e$ for some stabilizing gain $K_e \in \mathbb{R}^{m\times n}$ that satisfies (3) under the dynamics (5). It is possible to rewrite u_k^e in terms of $\bar{x}$ and $\bar{u}$ and compute an estimate of K_e denoted by $\widehat{K}_e \in \mathbb{R}^{m\times n}$ using a least-squares update rule [13] as

$$\begin{aligned} \bar{u} &= -\widehat{K}_e \bar{x} \\ \widehat{K}_e &= -\bar{u}\bar{x}^\top (\bar{x}\bar{x}^\top)^{-1}. \end{aligned} \tag{6}$$

The LS rule (6) provides an approximate value of the control gain K_e due to noise at the state or control measurements and the number of data points stored in the vectors $\bar{x}$ and $\bar{u}$. This gain is used to uncover the hidden cost of the expert's data. Since we are dealing with an inverse [41] optimal control problem there exists multiple weight matrices S and R that achieve the same hippocampus' performance [42]. For sake of simplicity, the weight function R is assumed to be known in advance.

III. Neocortex Online Control Policy

The on-line control policy of the neocortex is computed by a Q-learning algorithm [11] using the cost inferred by the striatum. Moreover, we are interested in computing new control gains $\widehat{K}_k \in \mathbb{R}^{m\times n}$ and kernel matrices $\widehat{P}_k \in \mathbb{R}^{n\times n}$ that at each iteration k approximate to the experts gain $\widehat{K}_e$ and kernel matrix P, respectively. In parallel, a least squares identification algorithm is used to estimate the parameters of the system A and B which will be used by the striatum in the inference step.

A. Q-learning

The optimal value function of (2) is defined as

$$V^*(x_k) = x_k^\top P x_k. \tag{7}$$

The Hamiltonian associated to (1) and (2) is

$$\begin{aligned} H(x_k, u_k) = \; & x_k^\top S x_k + u_k^\top R u_k - x_k^\top P x_k \\ & + (Ax_k + Bu_k)^\top P (Ax_k + Bu_k). \end{aligned} \tag{8}$$

Since the matrices A and B are unknown, then an action-value function $Q(x_k, u_k) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is used to derive the model-free Q-learning algorithm. The action value function verifies the following equality [7]

$$Q(x_k, u_k) = V^*(x_k) + H(x_k, u_k). \tag{9}$$

The optimal Q-function satisfies the equality $Q^*(x_k, u_k^*) = V^*(x_k)$. Substituting (7) and (8) in (9) gives

$$\begin{aligned} Q(x_k, u_k) &= \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} A^\top P A + S & B^\top P A \\ A^\top P B & B^\top P B + R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &= z_k^\top \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} z_k = z_k^\top M z_k. \end{aligned} \tag{10}$$

where $z_k = [x_k^\top, u_k^\top]^\top \in \mathbb{R}^{n+m}$ and $M \in \mathbb{R}^{(n+m)\times(n+m)}$. The Q-function is linearly parametrized as

$$Q(x_k, u_k) = \theta^\top (z_k \otimes z_k), \tag{11}$$

where $\theta = \text{vech}(M) \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$, $\text{vech}(M)$ stands to the half vectorization of matrix M and $\otimes$ denotes the symmetric Kronecker product. The optimal control policy $u_k^* = -Kx_k$ is derived from the stationary condition $\frac{\partial Q(x_k, u_k)}{\partial u_k} = 0$ as

$$u_k^* = -Kx_k = -Q_{uu}^{-1} Q_{xu}^\top x_k. \tag{12}$$

The optimal Bellman equation [34] for the Q-function (9) is

$$Q(x_k, u_k^*) = x_k^\top S x_k + (u_k^*)^\top R u_k^* + Q(x_{k+1}, u_{k+1}^*) \tag{13}$$

which can be equivalently written as

$$0 \quad = x_k^\top S x_k + u_k^\top R u_k - \theta^\top \Phi_k \tag{14}$$

where $\Phi_k = (z_k \otimes z_k - z_{k+1} \otimes z_{k+1})$. Sine θ is unknown, then the following approximation is used,

$$\widehat{Q}(x_k, u_k) = \widehat{\theta}_k^\top (z_k \otimes z_k), \tag{15}$$

where $\widehat{\theta}_k \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ is an estimate of θ. Then $\widehat{\theta}_k$ can be obtained through a LS rule by collecting at least $\frac{1}{2}(n+m)(n+m+1)$ points as

$$\widehat{\theta}_k = (\Phi_k \Phi_k^\top)^{-1} \Phi_k (x_k^\top S x_k + u_k^\top R u_k) \tag{16}$$

The estimate $\widehat{M}_k$ is obtained in each iteration k and the optimal control policy is computed by

$$\widehat{u}_k^* = -\widehat{K}_k x_k = -\widehat{Q}_{uu}^{-1} \widehat{Q}_{xu}^\top x_k, \tag{17}$$

where $\widehat{K}_k \in \mathbb{R}^{m \times n}$ is an estimate of the control gain K. The matrix $\widehat{M}_k$ and gain matrix $\widehat{K}_k$ satisfy the following relation [8]

$$\widehat{P}_k = \begin{bmatrix} I \\ -\widehat{K}_k \end{bmatrix}^\top \widehat{M}_k \begin{bmatrix} I \\ -\widehat{K}_k \end{bmatrix}, \tag{18}$$

where $\widehat{P}_k \in \mathbb{R}^{n \times n}$ is an estimate of the kernel matrix P which verifies $\widehat{V}^*(x_k) = x_k^\top \widehat{P}_k x_k$.

B. Least-Squares Identification rule

The control gain of the optimal control policy (3) or (12) exhibits a nonlinear relation between the coupling matrices A and B and the kernel matrix P such that it is not possible to express the policy as a linear parameterization in terms of those matrices. This issue can be solved by using a least-squares rule [41], [43] to identify the matrices A and B. Define an estimated model of (1) as

$$\widehat{x}_{k+1} = \widehat{A}_k x_k + \widehat{B}_k u_k = \varphi_k^\top \widehat{\vartheta}_k, \tag{19}$$

where $\widehat{A}_k \in \mathbb{R}^{n \times n}$ and $\widehat{B}_k$ are estimates of matrices A and B, $\varphi_k = \varphi(x_k, u_k) \in \mathbb{R}^p$ is a regressor matrix that depends on x_k and u_k, and $\widehat{\vartheta}_k \in \mathbb{R}^p$ is an estimate vector composed by the estimates $\widehat{A}_k$ and $\widehat{B}_k$. Define the dynamic error as

$$\widetilde{x}_k = \widehat{x}_{k+1} - x_{k+1}. \tag{20}$$

The main goal is to minimize the following cost index

$$J_1 = \sum_{k=1}^{n} \widetilde{x}_k^\top \widetilde{x}_k. \tag{21}$$

The minimum of the cost index is a zero of the gradient, i.e., $\frac{\partial J_1}{\partial \widehat{\vartheta}_k} = 0$,

$$\begin{aligned} \frac{\partial J_1}{\partial \widehat{\theta}_k} &= \frac{\partial J_1}{\partial \widetilde{x}_{k+1}} \frac{\partial \widetilde{x}_{k+1}}{\partial \widehat{\vartheta}_k} = 2 \sum_{k=1}^{n} \widetilde{x}_{k+1}^\top \frac{\partial}{\partial \widehat{\vartheta}_k} \left(\varphi_k^\top \widehat{\vartheta}_k - x_{k+1} \right) \\ &= 2 \sum_{k=1}^{n} (\varphi_k^\top \widehat{\vartheta}_k - x_{k+1})^\top \varphi_k^\top = 0. \end{aligned}$$

If the inverse of $\sum_{k=1}^{n} \varphi_k \varphi_k^\top$ exists, then the LS solution identification rule is

$$\widehat{\vartheta}_k = \left(\sum_{k=1}^{n} \varphi_k \varphi_k^\top \right)^{-1} \sum_{k=1}^{n} \varphi_k x_{k+1}. \tag{22}$$

Convergence of both the Q-learning and LS-identification rules are achieved under the fulfilment of a persistent of excitation (PE) condition [44].

IV. Striatum Cost Inference

The striatum relates the hippocampus and the neocortex control policies to infer the cost function $r(\cdot)$ used to obtain the expert's data.

A. Kernel Matrix Estimation

The first step is to relate the hippocampus and neocortex control gains to generate a new kernel matrix $\mathcal{P}_k \in \mathbb{R}^{n \times n}$ that is closer to the expert's kernel matrix P [45]. Define the control gain error $\widetilde{K}_k \in \mathbb{R}^{m \times n}$ as

$$\begin{aligned} \widetilde{K}_k &= \widehat{K}_e - \widehat{K}_k \\ &= -\left(\bar{u}\bar{x}^\top (\bar{x}\bar{x}^\top)^{-1} + L_k^{-1} \widehat{B}_k^\top \widehat{P}_k \widehat{A}_k \right), \end{aligned} \tag{23}$$

where $L_k = R + \widehat{B}_k^\top \widehat{P}_k \widehat{B}_k$. The first goal of the striatum is to minimize the following cost index

$$E = \text{tr}\{\widetilde{K}_k^\top \widetilde{K}_k\}. \tag{24}$$

A modified gradient descent rule of the form

$$\mathcal{P}_k = \widehat{P}_k - \alpha \nabla_P E \tag{25}$$

is used to obtain $\mathcal{P}_k$, where $\alpha \in \mathbb{R}_+$ is a learning rate and $\nabla_P = \frac{\partial}{\partial \widehat{P}_k}$ is the gradient respect to the kernel matrix $\widehat{P}_k$. First, notice that

$$\begin{aligned} \nabla_P \Big\{ L_k L_k^{-1} = I \Big\} \\ \widehat{B}_k^\top \widehat{B}_k L_k^{-1} + L_k \nabla_P L_k^{-1} = 0 \\ \nabla_P L_k^{-1} = -L_k^{-1} \widehat{B}_k^\top \widehat{B}_k L_k^{-1}. \end{aligned}$$

The final update rule is

$$\begin{aligned} \mathcal{P}_k = \quad & \widehat{P}_k + \alpha \Big(\widetilde{K}_k^\top L_k^{-1} \widehat{B}_k^\top (\widehat{A}_k - \widehat{B}_k \widehat{K}_k) \\ & + (\widehat{A}_k - \widehat{B}_k \widehat{K}_k)^\top \widehat{B}_k L_k^{-1} \widetilde{K}_k \Big). \end{aligned} \tag{26}$$

Notice that the update rule (26) uses as initial value the kernel matrix $\widehat{P}_k$ and the estimates $\widehat{A}_k$ and $\widehat{B}_k$ obtained from the neocortex learning systems.

B. Cost Inference

The cost inference algorithm computes the weight matrix S^i of iteration i using an inverse optimal control (IOC) algorithm based on the DARE (4), the estimates $\widehat{A}_k$ and $\widehat{B}_k$, and the new kernel matrix $\mathcal{P}_k$ as

$$S^i = \mathcal{P}_k - \widehat{A}_k^\top \mathcal{P}_k \widehat{A}_k + \widehat{A}_k^\top \mathcal{P}_k \widehat{B}_k (R + \widehat{B}_k^\top \mathcal{P}_k \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{P}_k \widehat{A}_k \tag{27}$$

Theorem 1 discusses the convergence of the weight matrix S^i using the IOC (27) and the one-step gradient rule (26).

Theorem 1: The weight matrix S^i obtained in (27) converges in the sense that $\|S^{i+1} - S^i\| \leq \varepsilon_S$, for some small constant $\varepsilon_S \in \mathbb{R}_+$, as the number of iterations i increases. In consequence, E converges to zero and hence, the control gain error $\widetilde{K}_k$ converges to zero which implies that $\widehat{K}_e$ converges to K_e.

Proof: Consider the DARE (27) of the IOC problem

$$\begin{aligned} S^{i+1} = & \mathcal{P}_k^i + \widehat{A}_k^\top \mathcal{P}_k^i \widehat{B}_k (R + \widehat{B}_k^\top \mathcal{P}_k^i \widehat{B}_k)^{-1} \widehat{B}_k^\top \mathcal{P}_k^i \widehat{A}_k \\ & - \widehat{A}_k^\top \mathcal{P}_k^i \widehat{A}_k \end{aligned} \tag{28}$$

Substituting (25) in (28) gives

$$\begin{aligned} S^{i+1} =&(\widehat{P}_k^i - \alpha\nabla_P E^i) - \widehat{A}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{A}_k \\ &+ \widehat{A}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{B}_k(R + \widehat{B}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{B}_k)^{-1} \\ &\times \widehat{B}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{A}_k. \end{aligned} \tag{29}$$

where $G = \nabla_P E^i$. The Q-learning algorithm satisfies the following DARE

$$\begin{aligned} S^{i+1} =&\widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k(R + \widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k)^{-1}\widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k \\ &+ \widehat{P}_k^{i+1} - \widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k. \end{aligned} \tag{30}$$

Matching (29) and (30) gives

$$\begin{aligned} &\widehat{P}_k^{i+1} + \widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k(R + \widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k)^{-1}\widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k \\ &- \widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k = \widehat{P}_k^i - \widehat{A}_k^\top \widehat{P}_k^i \widehat{A}_k \\ &+ \widehat{A}_k^\top \widehat{P}_k^i \widehat{B}_k(R + \widehat{B}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{B}_k)^{-1}\widehat{B}_k^\top \widehat{P}_k^i \widehat{A}_k \\ &- \alpha\Big(\widehat{A}_k^\top G\widehat{B}_k(R + \widehat{B}_k^\top(\widehat{P}_k^i - \alpha\nabla_P E^i)\widehat{B}_k)^{-1}\widehat{B}_k^\top \nabla_P E^i \widehat{A}_k \\ &+ \widehat{A}_k^\top G\widehat{B}_k(R + \widehat{B}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{B}_k)^{-\top}\widehat{B}_k^\top G\widehat{A}_k\Big) \\ &+ \alpha^2 \widehat{A}_k^\top \nabla_P E^i(R + \widehat{B}_k^\top(\widehat{P}_k^i - \alpha G)\widehat{B}_k)^{-1}\widehat{B}_k^\top G\widehat{A}_k. \end{aligned} \tag{31}$$

The rule (25) updates in each iteration i the kernel matrix $\mathcal{P}_k^i$ such that the control gain error $\widetilde{K}_k^i$ converges towards to zero. That is, $\lim_{i\to\infty}\nabla_P E^i = 0$ implies that $\lim_{i\to\infty}\mathcal{P}_k^i = \widehat{P}_k^i$. Therefore, (31) is simplified to

$$\begin{aligned} &\lim_{i\to\infty}\Big(\widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k(R + \widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{B}_k)^{-1}\widehat{B}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k \\ &\qquad + \widehat{P}_k^{i+1} - \widehat{A}_k^\top \widehat{P}_k^{i+1}\widehat{A}_k + \Big) = \\ &\lim_{i\to\infty}\Big(\widehat{A}_k^\top \widehat{P}_k^i \widehat{B}_k(R + \widehat{B}_k^\top \widehat{P}_k^i \widehat{B}_k)^{-1}\widehat{B}_k^\top \widehat{P}_k^i \widehat{A}_k \\ &\qquad + \widehat{P}_k^i - \widehat{A}_k^\top \widehat{P}_k^i \widehat{A}_k\Big). \end{aligned} \tag{32}$$

From the above equality we can conclude that

$$\lim_{i\to\infty} S^{i+1} = \lim_{i\to\infty} S^i \Rightarrow S^{i+1} = S^i. \tag{33}$$

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

Fig. 1 depicts the block diagram of the proposed inference algorithm. The scheme is summarized as follows: two control policies coming from expert's data (hippocampus fast learning) and from online data (neocortex pattern association learning) are connected through the striatum (modelled as an IOC) to infer the hippocampus cost function to the neocortex to enable policy generalization for different systems and environments.

V. Simulation studies

The proposed approach was tested in a high order power system [11]. The discrete-time plant is

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

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

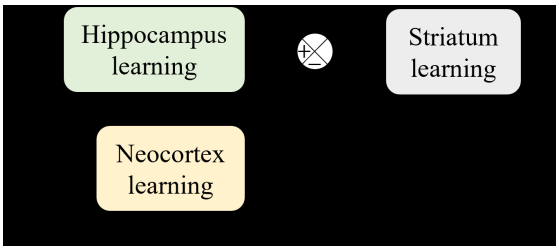


Fig. 1. Cost Inference block diagram based on the proposed Human-Behaviour Learning approach.

The weight matrices of the expert's cost are chosen as $S = 2I_4$ and $R = 1$. The kernel matrix and control gain under the expert's cost are

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

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

The data of the power system trajectories under the expert's control policy are stored in the vector $\bar{x}$ and $\bar{u}$. A small uniformly random noise is added at the state and control measurements to model sensor noise. The hippocampus policy gain is computed using (6) whose value is

$$\widehat{K}_e = \begin{bmatrix} 0.2845 & 2.0589 & 0.4452 & -0.0791 \end{bmatrix}.$$

The learning rate of the striatum learning system is $\alpha = 0.9$. The sinusoidal PE signal is added to the control input to ensure parameter convergence. The initial weight matrix of the neocortex cost is setted to $S^0 = I_4$. Fig. 2 shows the results of the proposed inference algorithm. The learned matrices of the identification algorithm are

$$\widehat{A}_k = \begin{bmatrix} 0.9616 & 1.0046 & 0.0867 & -0.0450 \\ -0.0739 & 0.7489 & 0.1154 & -0.1038 \\ -0.5354 & -0.3399 & 0.2303 & -0.7378 \\ 0.0593 & 0.0313 & 0.0020 & 0.9993 \end{bmatrix},$$

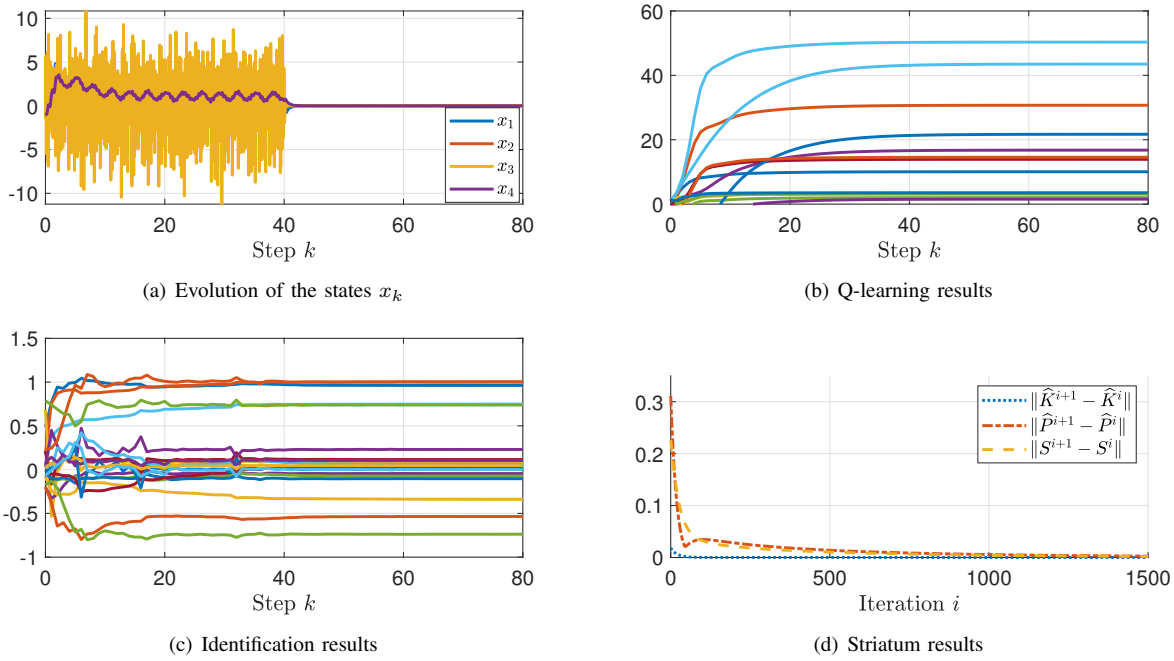
$$\widehat{B}_k = \begin{bmatrix} 0.0450 & 0.1038 & 0.7378 & 0.0007 \end{bmatrix}^\top,$$

$$\widehat{P}^i = \begin{bmatrix} 3.9820 & 3.7353 & 2.8939 & 3.0092 \\ 3.7353 & 10.8356 & 7.7231 & 1.4985 \\ 2.8939 & 7.7231 & 1.8089 & 2.6277 \\ 3.0092 & 1.4985 & 2.6277 & 20.4560 \end{bmatrix},$$

$$\widehat{K}^i = \begin{bmatrix} 0.2845 & 2.0589 & 0.4452 & -0.0791 \end{bmatrix},$$

$$S^i = \begin{bmatrix} 2.6718 & 4.9403 & 2.9683 & 1.3431 \\ 4.9403 & 15.5254 & 7.4807 & 4.2791 \\ 2.9683 & 7.4807 & 1.6318 & 2.9920 \\ 1.3431 & 4.2791 & 2.9920 & 1.9874 \end{bmatrix}.$$

The LS-identification algorithm of the neocortex shows high accurate results for the coupling matrices $\widehat{A}_k$ and $\widehat{B}_k$. On the other hand, the striatum learning algorithm converges to the same gain of the hippocampus control policy. However, notice that both the kernel matrix $\widehat{P}^i$ and the weight matrix S^i were completely different to the expert's matrices. Moreover, both matrices are negative definite instead of positive



(a) Evolution of the states x_k

(b) Q-learning results

(c) Identification results

(d) Striatum results

Fig. 2. Complementary learning results

definite due to the lack of constraints in the IOC rule. This causes that the inference algorithm to seek at any direction where the gradient is minimized and hence negative definite matrices can be found.

To further exhibit the benefits of the proposed approach, we compare the performance of the optimal value function under the kernel matrices P and $\widehat{P}_k$. Fig. 3 shows the comparison results.

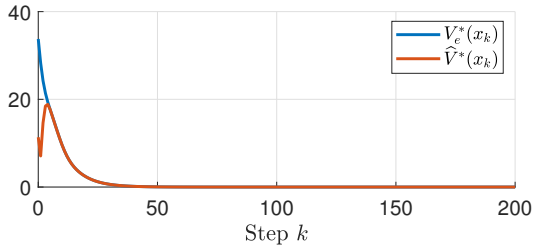


Fig. 3. Optimal value function trajectories

The results show an improvement of the optimal value function trajectories despite the kernel matrix $\widehat{P}^i$ is negative definite. Different initial weight matrices S^0 are tested to verify the approach. However, it is observed that for small weight matrices $\|S^0\| \leq 1$, the striatum learning system gives negative definite kernel matrices $\mathcal{P}^i$ and S^i that destabilizes the neocortex learning system. This problem is informative because it shows that the striatum gives one of the multiple solutions of the IOC problem which are not necessarily a stable solution of the optimal control problem. Furthermore, for high dimensional systems we need to calculate at least $\frac{n(n+1)}{2}$ terms of the reward's weight matrix and hence the number of combinations that give the same expert's control gain increases infinitely. To solve this issue, constraints have to be added to reduce the number of possible solutions and to obtain real and representative weights matrices of the expert's reward function. This is topic for further work.

VI. Conclusions

This paper reports a cost inference algorithm of discrete-time linear quadratic policies. The approach is based on a human-behavior learning algorithm that exploits on-line and off-line policies data obtained from the neocortex and the hippocampus learning systems. Whilst the hippocampus offers fast learning models, the neocortex learns well defined pattern structures for decision making. These learning systems are related by the striatum to exploit the advantages of each learning system to achieve generalization in the final control policy. The key idea is to infer the hippocampus cost function to the neocortex in order to generalize policies to different systems and environments. A Q-learning and LS-identification algorithms were used to model the neocortex and an IOC algorithm to model the striatum. Simulations studies show that it is possible to infer the desired performance through the cost function, however constraints have to been added in the inference algorithm to achieve more accurate results.

Future research covers the extension of the proposed approach for nonlinear systems and for for non-quadratic reward functions is the next challenge of the work.

References

[1] A. Perrusquía and W. Yu, "Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2920–2936, 2020.

[2] J.-H. Kim and F. Lewis, "Model-free H_∞ control design for unknown linear discrete-time systems via Q-learining with LMI," *Automatica*, vol. 46, pp. 1320–1326, 2010.

[3] F. L. Lewis, *Optimal Control*. New York, NY, USA: Wiley, 2012.

[4] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpor, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, pp. 1167–1175, 2014.

[5] 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*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] A. Perrusquía, "Solution of the linear quadratic regulator problem of black box linear systems using reinforcement learning," *Information Sciences*, vol. 595, pp. 364–377, 2022.

[7] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, pp. 14–20, 2017.

[8] A. Perrusquía, W. Yu, and A. Soria, "Position/force control of robot manipulators using reinforcement learning," *Industrial Robot: the international jounral of robotics research and application*, vol. 46, no. 2, pp. 267–280, 2019.

[9] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control using natural decision methods to design optimal adaptive controllers," *IEEE Conrol Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[10] I. Grondman, L. Buşoniu, G. A. Lopes, and R. Babŭska, "A survey of actor-critic reinforcement learning: standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, PART C*, vol. 42, no. 6, pp. 1291–1307, 2012.

[11] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 5, pp. 1523–1536, 2018.

[12] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

[13] M. Wiering and M. van Otterlo, *Reinforcement Learning: State-of-art*. Springer, 2012.

[14] N. Ab Azar, A. Shahmansoorian, and M. Davoudi, "From inverse optimal control to inverse reinforcement learning: A historical review," *Annual Reviews in Control*, vol. 50, pp. 119–138, 2020.

[15] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," *Artificial Intelligence Review*, pp. 1–29, 2021.

[16] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.

[17] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in cognitive sciences*, vol. 20, no. 7, pp. 512–534, 2016.

[18] A. Perrusquía, W. Yu, and X. Li, "Nonlinear control using human behavior learning," *Information Sciences*, vol. 569, pp. 358–375, 2021.

[19] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.

[20] A. Perrusquía and 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.

[21] R. C. O'Reilly, R. Bhattacharyya, M. D. Howard, and N. Ketz, "Complementary learning systems," *Cognitive science*, vol. 38, no. 6, pp. 1229–1248, 2014.

[22] M. G. Mattar and N. D. Daw, "Prioritized memory access explains planning and hippocampal replay," *Nature neuroscience*, vol. 21, no. 11, pp. 1609–1617, 2018.

[23] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman, "The hippocampus as a predictive map," *Nature neuroscience*, vol. 20, no. 11, pp. 1643–1653, 2017.

[24] H. F. Ólafsdóttir, D. Bush, and C. Barry, "The role of hippocampal replay in memory and planning," *Current Biology*, vol. 28, no. 1, pp. R37–R50, 2018.

[25] 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," *Journal of Neuroscience*, vol. 37, no. 28, pp. 6686–6697, 2017.

[26] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[27] A. Perrusquía, W. Yu, and A. Soria, "Large space dimension reinforcement learning for robot position/force discrete control," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2019, pp. 91–96.

[28] S. Blakeman and D. Mareschal, "A complementary learning systems approach to temporal difference learning," *Neural Networks*, vol. 122, pp. 218–230, 2020.

[29] A. Perrusquía, W. Yu, and X. Li, "Multi-agent reinforcement learning for redundant robot control in task-space," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 1, pp. 231–241, 2021.

[30] B. Kiumarsi and F. L. Lewis, "Actor-critic based optimal tracking for partially unknown nonlinear discrete- time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 140–151, 2015.

[31] A. Perrusquía, W. Yu, and X. Li, "Redundant robot control using multi agent reinforcement learning," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 1650–1655.

[32] A. Perrusquía and W. Yu, "Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2020.

[33] R. Kamalapurkar, P. Walters, and W. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[34] L. Buşoniu, R. Babuška, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming using Function Approximators*. CRC Press, 2010.

[35] A. Perrusquía and W. Yu, "Continuous-time reinforcement learning for robust control under worst-case uncertainty," *International Journal of Systems Science*, vol. 52, no. 4, pp. 770–784, 2021.

[36] W. Schultz, P. Apicella, E. Scarnati, and T. Ljungberg, "Neuronal activity in monkey ventral striatum related to the expectation of reward," *Journal of neuroscience*, vol. 12, no. 12, pp. 4595–4610, 1992.

[37] A. Perrusquía, "A complementary learning approach for expertise transference of human-optimized controllers," *Neural Networks*, vol. 145, pp. 33–41, 2022.

[38] J. L. McClelland, B. L. McNaughton, and 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." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[39] C.-T. Chen, *Linear System Theory and Design*. Oxford University Press, 1999.

[40] A. Perrusquía and W. Yu, "Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning," *IEEE Transactions on Cybernetics*, pp. 1–10, 2020.

[41] W. Yu and A. Perrusquía, "Simplified stable admittance control using end-effector orientations," *International Journal of Social Robotics*, vol. 12, no. 5, pp. 1061–1073, 2020.

[42] H. El-Hussieny and J.-H. Ryu, "Inverse discounted-based LQR algorithm for learning human movement behaviors," *Applied Intelligence*, vol. 49, no. 4, pp. 1489–1501, 2019.

[43] A. Perrusquía, R. Garrido, and W. Yu, "An input error method for parameter identification of a class of euler-lagrange systems," in *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2021, pp. 1–6.

[44] F. L. Lewis, S. Jagannathan, and A. Yeşildirek, *Neural network control of robot manipulators and nonlinear systems*. Taylor & Francis, 1999.

[45] A. Perrusquía, "Human-behavior learning: A new complementary learning perspective for optimal decision making controllers," *Neurocomputing*, 2022.

2022-06-30

Cost inference of discrete-time linear quadratic control policies using human-behaviour learning

Perrusquía, Adolfo

IEEE

Perrusquia A, Guo W. (2022) Cost inference of discrete-time linear quadratic control policies using human-behaviour learning. In: CODiT 2022: 8th International Conference on Control, Decision and Information Technologies, 17-20 May 2022, Istanbul, Turkey, pp. 165-170
https://doi.org/10.1109/CoDIT55151.2022.9804118