Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), 20-24 August 2022, Mexico City, Mexico
DOI:10.1109/CASE49997.2022.9926671

Performance Objective Extraction of Optimal Controllers: A Hippocampal Learning Approach

Adolfo Perrusquía Weisi Guo

***Abstract*— Intention inference of autonomous vehicles is crucial to guarantee safety and to mitigate risk. This paper reports a performance objective extraction from expert's data trajectories for experience transference and to uncover the hidden cost associated to the intent. The algorithm is inspired in the hippocampus learning system for experience exploitation that exhibits the human brain. The hippocampus is responsible of memory and to store past experiences to enable transfer learning and fast convergence.**

The proposed algorithm extracts, from expert's data, the performance matrices associated to a hidden utility function using a complementary approach based on an off-policy policy iteration and a matrix extraction inverse reinforcement learning algorithms. Exact performance extraction is obtained by adding a constraint in terms of the measurements of the utility function in a batch-least squares algorithm. Convergence of the proposed approach is verified using Lyapunov recursions. Simulation studies are carried out to demonstrate the effectiveness of the proposed approach.

I. Introduction

In the last years the number of datasets [1], [2] regarding to regression, classification, and control problems has been increasing due to the high capabilities that exhibit artificial intelligence (AI) and machine learning (ML) algorithms [3], [4] for decision making in a human-behavior perspective [5], that is, they have generalization and inference properties.

In a control perspective, these data belong to states, inputs, or any signal of interest that depicts a desired performance or an expert behavior [6], [7]. This performance/behavior or intention objective is in most cases hidden and requires knowledge of the physics of the system to extract it; this is known as physics-informed intention inference or model-based inference. Furthermore, model-free approaches [8] cannot infer adequately the performance objective due to the lack of constraints [9]. The main problem regards in extracting the performance from expert's data to enable transfer learning [10] and intention inference.

The performance objective is directly related to a cost, utility function, or reward to be optimized in an infinite or discounted horizon. This function serves as a stimuli [11] that receives the system to adjust the control actions similarly as humans do [12], [13]. Reinforcement Learning (RL) [14], [15] is one of the main machine learning algorithms that seeks to optimize a reward function using either model-based, e.g., Linear Quadratic Regulator (LQR) and Adaptive Dynamic Programming (ADP) algorithms [16]–[18], critic [19] and actor-critic algorithms [20]–[22]; or model-free algorithms, e.g., Q-learning [23]–[25] and policy iterations [26] algorithms. These algorithms obtain the optimal control policy by using a pre-defined performance objective which differs from the expert's performance objective/reward. In addition, the reward function is considered as the most succinct, robust [27], and transferable definition of the task. This is why it is of high importance to extract the hidden reward function from the expert's data. There exists several model-based approaches to infer the performance matrices associated to a quadratic performance objective. In general, these approaches solve an inverse optimal control (IOC) problem [28]–[30] and its model-free version is known as inverse reinforcement learning (IRL) [31] which are generally solved by linear (LP) or quadratic programming (QP) algorithms under a binary reward function which is a very restrictive approach [32].

In the last decade, a novel perspective known as human-behavior learning [33] has been used as a general approach that combines different sources of knowledge to enhance decision making [34]. This approach models the three main learning systems of the brain cortex: the hippocampus, the neocortex, and the striatum. The hippocampus is related to memory and experiences [35]–[37] and enables fast learning and experience transference. The neocortex provides of well-distributed structures [38], [39] for pattern dependent learning which is slow in comparison to the hippocampus. The striatum [40] is mainly a communication channel that relates the hippocampus and neocortex to enable complementary learning that enhances decision making [41]–[43].

In this context, expert's data is directly associated to the hippocampus learning system. The hippocampus is responsible to teach the neocortex the best way to execute a task [44], [45]. Analogously, the expert's data are used as a demonstration of how the system has to behave. Furthermore, the extraction of the performance objective enables: i) experience transference and ii) intention inference. In this paper, a performance objective extraction based on a hippocampal learning approach is proposed. The algorithm is able to extract the hidden performance objective using only expert's data that models an optimal desired behavior. The approach is divided in two main parts: 1) an off-policy learning algorithm that computes an optimal control policy in terms of an initial performance objective function and 2) an objective extraction algorithm that updates the objective

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

function in each episode. Convergence to the real expert's performance is achieved by adding a constraint in terms of the measurements of the reward/utility function. Simulations studies are carried out to verify the proposed approach.

The main contributions of this paper are: i) a model-free performance objective extraction for linear systems, ii) the estimates of the performance matrices converge to the expert's matrices by incorporating constraints in the learning law, iii) convergence of the proposed approach is verified using Lyapunov recursions.

Throughout this paper, $\mathbb{N}$, $\mathbb{Z}^+$, $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the spaces of natural numbers, positive integers, real numbers, real n-vectors, and real $n \times m$-matrices, respectively; $I_n \in \mathbb{R}^{n\times n}$ denotes an identity matrix; $\otimes$, $\bar{\otimes}$, vec(A), and vech(A) defines the Kronecker product, the symmetric Kronecker product, the matrix vectorization, and the half-vectorization; the norms $\|A\| = \sqrt{\lambda_{\max}(A^\top A)}$ and $\|x\|$ stand for the induced matrix and vector Euclidean norms, respectively; where $x \in \mathbb{R}^n$, $A, B \in \mathbb{R}^{n\times n}$ and $n, m \in \mathbb{N}$.

II. Hippocampus learning

The hippocampus is directly related with experiences and memory. These experiences are generally stored in datasets that provide an effective way to exhibit a desired performance under a hidden objective function or reward function. Whilst many reinforcement learning architectures use online data measured from system trajectories to derived the optimal control policy, the hippocampus learning uses stored data to derived new control policies under iterative objective functions until the same expert policy is achieved.

Assume we collect expert's data [41] from the measurements of the states $x_e \in \mathbb{R}^n$, the inputs $u_e \in \mathbb{R}^m$, and the values of the utility function $\xi(x_e, u_e) \in \mathbb{R}$ of an expert trajectory in a time $t = kT$ with sampling period $T > 0$ and $k \in \mathbb{Z}^+$. These data are stored in the following matrices $X_e = [x_e(0), \cdots, x_e((k-1)T)] \in \mathbb{R}^{n\times k}$, $U_e = [u_e(0), \cdots, u_e((k-1)T)] \in \mathbb{R}^{m\times k}$, and $\Xi = [\xi(0), \cdots, \xi((k-1)T)] \in \mathbb{R}^{1\times k}$. The states x_e and control input u_e verify the following dynamic equation

$$\dot{x}_e = Ax_e + Bu_e. \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the matrices that define the dynamics of the unknown linear system. In addition, the expert's input u_e is the control input that minimizes the utility function $\xi(x_e, u_e)$ in an infinite horizon [46] and satisfies the following value function

$$\begin{aligned} V(x_e) &= \int_t^\infty \xi(x_e, u_e) d\tau \\ &= \int_t^\infty (x_e^\top S_e x_e + u_e^\top R_e u_e) d\tau \end{aligned} \tag{2}$$

where $S_e = S_e^\top \geq 0 \in \mathbb{R}^{n\times n}$ and $R_e = R_e^\top > 0 \in \mathbb{R}^{m\times m}$ define the unknown performance objective matrices of the utility function. In terms of the classic ADP/RL formulation, the optimal value function $V^*(x_e)$ is quadratic in the state [47], i.e.,

$$V^*(x_e) = x_e^\top P_e x_e, \tag{3}$$

for some positive definite kernel matrix $P_e = P_e^\top > 0 \in \mathbb{R}^{n\times n}$ which is the solution of the following algebraic Riccati equation [48]

$$A^\top P_e + P_e A - PBR_e^{-1}B^\top P_e + S_e = 0. \tag{4}$$

The Hamiltonian associated to (2) with respect to (1) and (3) is

$$\begin{aligned} H(x_e, u_e) = \;& x_e^\top P_e(Ax_e + Bu_e) + (Ax_e + Bu_e)^\top P_e x_e \\ & + x_e^\top S_e x_e + u_e^\top R_e u_e = 0. \end{aligned} \tag{5}$$

Applying the stationary condition $\frac{\partial H(x_e, u_e)}{\partial u_e} = 0$ [49] and solving for u_e yields the optimal control policy

$$u_e^* = -K_e x_e = -R_e^{-1}B^\top P_e x_e, \tag{6}$$

where $K_e = R_e^{-1}B^\top P_e \in \mathbb{R}^{m\times n}$ is the optimal stabilizing gain. Notice that K_e cannot be computed since R_e, B, P_e are unknown. However, we can compute an estimate of K_e denoted by $\widehat{K}_e \in \mathbb{R}^{m\times n}$ using only the control input and states measurements from the expert trajectory as

$$\begin{aligned} U_e &= -\widehat{K}_e X_e \\ \widehat{K}_e &= -U_e X_e^\top (X_e X_e^\top)^{-1}. \end{aligned} \tag{7}$$

The equation (7) is a least-squares (LS) solution for $\widehat{K}_e$ which is affected directly by the measurement noise. To overcome this issue, let construct the following matrix

$$\mathcal{A} = \begin{bmatrix} X_e \\ U_e \end{bmatrix} \in \mathbb{R}^{(n+m)\times k}. \tag{8}$$

Then, we can use a matrix approximation using singular-value-decomposition (SVD) or principal component analysis (PCA) [1] to maintain only the relevant dimensions associated to the largest singular values and delete the dimensions associated to the measurement noise.

At this point, we cannot apply any iterative algorithm since the expert's data is fixed. To fix this issue, an additional control input can be added to (1) as

$$\begin{aligned} \dot{x}_e &= Ax_e + B(u_e + v_i^j - v_i^j), \quad v_i^j = -K_i^j x_e, \\ &= (A - BK_i^j)x_e + B(u_e + K_i^j x_e) \\ &= A_k x_e + B(u_e + K_i^j x_e) \end{aligned} \tag{9}$$

where $K_i^j \in \mathbb{R}^{m\times n}$, $P_i^j \in \mathbb{R}^{n\times n}$, $S_i \in \mathbb{R}^{n\times n}$, and $R_i \in \mathbb{R}^{m\times m}$ denote the control gain, kernel matrix, and performance matrices which will be iteratively updated in each step j of episode i until they converge to the optimal values, that is, K_e, P_e, S_e, and R_e. The Hamiltonian (5) in terms of (9) under the new control policy $v^{j+1} = -K^{j+1}x_e$ where $K_i^{j+1} = R_i^{-1}B^\top P_i^j$ yields the following Bellman equation [50]

$$\begin{aligned} H(x_e, u_e) =\;& x_e^\top S_i x_e + x_e^\top P_i^j(A_k x_e + B(u_e + K_i^j x_e)) \\ & + (A_k x_e + B(u_e + K_i^j x_e))^\top P_i^j x_e \\ & - 2(u_e + K_i^j x_e)^\top R_i K_i^{j+1} x_e \\ & + x_e^\top (K_i^j)^\top R_i K_i^j x_e = 0. \end{aligned} \tag{10}$$

The performance objective is directly related to the unknown matrices S and R, that is, they determine the importance of each state and boundedness in the optimal control design. The performance matrices S and R will be extracted iteratively in each episode from the expert's data. Integrating (10) in a time window of length $[t : t+\mathcal{T}]$ for some small $\mathcal{T} > 0$ gives

$$\begin{aligned} x_e^\top(t+\mathcal{T})P_i^j x_e(t+\mathcal{T}) - x_e^\top(t)P_i^j x_e(t) \\ -2\int_t^{t+\mathcal{T}} (u_e + K_i^j x_e)^\top R_i K_i^{j+1} x_e d\tau \\ = -\int_t^{t+\mathcal{T}} x_e^\top (S_i + (K_i^j)^\top R_i K_i^j) x_e d\tau \end{aligned} \quad (11)$$

A least-squares (LS) algorithm is used to find the optimal kernel matrix P_i^j and the optimal control gain K_i^j associated to the initial performance matrix S_i. Then, a system of equations composed of κ equations are constructed from the collection of measurements of the extended trajectories (9). The following matrices are constructed

$$z = \left[x_e(\tau)\bar{\otimes}x_e(\tau)\Big|_t^{t+\mathcal{T}}, \cdots, x_e(\tau)\bar{\otimes}x_e(\tau)\Big|_{t+(\kappa-1)\mathcal{T}}^{t+\kappa\mathcal{T}} \right]^\top,$$

$$I_{xx} = \left[\int_t^{t+\mathcal{T}} x_e \otimes x_e d\tau, \cdots, \int_{t+(\kappa-1)\mathcal{T}}^{t+\kappa\mathcal{T}} x_e \otimes x_e d\tau \right]^\top,$$

$$I_{xu} = \left[\int_t^{t+\mathcal{T}} x_e \otimes u_e d\tau, \cdots, \int_{t+(\kappa-1)\mathcal{T}}^{t+\kappa\mathcal{T}} x_e \otimes u_e d\tau \right]^\top$$

So, the system of equations written in matrix form can be solved as

$$\begin{aligned} \Phi\Theta &= \Omega \\ \Theta &= (\Phi^\top\Phi)^{-1}\Phi^\top\Omega, \end{aligned} \quad (12)$$

where

$$\Theta = \begin{bmatrix} \text{vech}(P_i^j) \\ \text{vec}(K_i^{j+1}) \end{bmatrix} \in \mathbb{R}^p, \quad p = \frac{1}{2}n(n+1) + nm$$

$$\Phi = \left[z, -2[I_{xx}(I_n \otimes (K_i^j)^\top R_i) + I_{xu}(I_n \otimes R_i)] \right] \in \mathbb{R}^{\kappa \times p}$$

$$\Omega = -I_{xx}\text{vec}(S_i + (K_i^j)^\top R_i K_i^j) \in \mathbb{R}^\kappa$$

If the regressor Φ fulfils a persistent excitation condition [51], then both the kernel matrix P_i^j and the control gain K_i^j converge to their optimal value respect to the initial performance matrices S_i and R_i. In the next section the performance matrices are updated and extracted from the expert's gain $\widehat{K}_e$. Convergence of a similar approximation of the batch-least squares algorithm (12) is discussed in [26].

III. PERFORMANCE MATRIX EXTRACTION

The hippocampus learning finds an optimal/near optimal control gain K_i^j in terms of the initial performance matrices S_i and R_i. For instance, let write K_i^j as K_i and P_i^j as P_i since we will work at the episode level. Define the gain error between the approximate expert gain $\widehat{K}_e$ and the hippocampus gain K_i as

$$\begin{aligned} e_k &= K_i - \widehat{K}_e \\ &= R_i^{-1}B^\top P_i + U_e X_e^\top (X_e X_e^\top)^{-1}. \end{aligned} \quad (13)$$

The kernel matrix P_i is the only free parameter than can be adjusted to reduce the gain error e_k. Therefore, the first main goal is to find the kernel matrix that minimizes the following cost index

$$E = \text{tr}\{e_k^\top e_k\} \quad (14)$$

Taking the partial derivative of E respect to the kernel matrix P_i and equalling to zero gives

$$\frac{\partial E}{\partial P_i} = \text{tr}\{BR_i^{-1}e_k + e_k^\top R_i^{-1}B^\top\} = 0.$$

Two considerations are needed: i) the solution of the optimization problem is a new kernel matrix $\mathcal{P}_i = \mathcal{P}_i^\top > 0 \in \mathbb{R}^{n\times n}$ which is only used to extract the performance matrices S and R, and ii) the term $R_i^{-1}B^\top = K_i P_i^{-1}$ which holds due to the invertibility of the kernel matrix P_i. Then the kernel matrix can be computed using the following one-step gradient rule

$$\mathcal{P}_i = P_i - \alpha[P_i^{-1}K_i^\top e_k + e_k^\top K_i P_i^{-1}] \quad (15)$$

where $\alpha > 0$ is the learning rate of the gradient rule. At instance, a LS rule cannot be used to compute the kernel matrix $\mathcal{P}$ since it requires to solve a linear Lyapunov equation [19] of the form $M^\top\mathcal{P} + \mathcal{P}M + Q = 0$, for some matrix $M \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{n\times n}$. However, matrix Q is not positive definite and hence multiple solutions for $\mathcal{P}_i$ can be obtained which are not necessarily positive definite.

Notice that if $\widehat{P}_i = \mathcal{P}_i$ implies that the gain error $e_k = 0_{m\times n}$ which means that K_i is equivalent to $\widehat{K}_e$. From this fact, is easy follow that

$$\begin{aligned} R_i^{-1}B^\top &= R_i^{-1}B^\top \\ \mathcal{K}_i\mathcal{P}_i^{-1} &= K_i P_i^{-1} \\ \mathcal{K}_i &= K_i P_i^{-1}\mathcal{P}_i \end{aligned} \quad (16)$$

for some stabilizing gain $\mathcal{K}_i \in \mathbb{R}^{m\times n}$ which is associated to the updated kernel matrix $\mathcal{P}_i$. After the updated kernel matrix $\mathcal{P}_i$ and gain $\mathcal{K}_i$ are found, then we can follow a similar approach to the hippocampus learning algorithm to estimate the performance matrices S_i and R_i associated to the expert's data using an inverse reinforcement learning algorithm (IRL). We can build a new extended dynamics from the expert's trajectories as

$$\begin{aligned} \dot{x}_e &= Ax_e + B(u_e + w_i - w_i), \quad w_i = -\mathcal{K}_i x_e, \\ \dot{x}_e &= A_w x_e + B(u_e + \mathcal{K}_i x_e), \end{aligned} \quad (17)$$

where $A_w = A - B\mathcal{K}_i$. Then the Hamiltonian associated to the new extended dynamics (17) is

$$\begin{aligned} H(x_e, u_e) =& x_e^\top S_{i+1}x_e + x_e^\top\mathcal{P}_i(A_w x_e + B(u_e + \mathcal{K}_i x_e)) \\ &+ (A_w x_e + B(u_e + \mathcal{K}_i x_e))^\top\mathcal{P}_i x_e \\ &+ 2u_e^\top R_{i+1}w_i - w_i^\top R_{i+1}w_i = 0. \end{aligned} \quad (18)$$

The gain $\mathcal{K}_i$ and the kernel matrix $\mathcal{P}_i$ in (18) are fixed. Integrating (18) in a time window of length $[t : t+\mathcal{T}]$ gives

$$\begin{aligned} \int_t^{t+\mathcal{T}} x_e^\top S_{i+1}x_e d\tau + 2\int_t^{t+\mathcal{T}} \eta_i^\top R_{i+1}w_i d\tau \\ = x_e^\top(t)\mathcal{P}_i x_e(t) - x_e^\top(t+\mathcal{T})\mathcal{P}_i x_e(t+\mathcal{T}) \end{aligned} \quad (19)$$

where $\eta_i = u_e - \frac{1}{2} w_i$. To find the next performance matrices S_{i+1} and R_{i+1}, a set of ι linear equations are constructed and subsequently a batch-least squares algorithm is applied. Define the following matrices

$$I_{xs} = \left[\int_t^{t+\mathcal{T}} x_e \bar{\otimes} x_e d\tau, \cdots, \int_{t+(\iota-1)\mathcal{T}}^{t+\iota\mathcal{T}} x_e \bar{\otimes} x_e d\tau \right]^\top$$
$$I_{uw} = \left[\int_t^{t+\mathcal{T}} \eta_i \otimes w_i d\tau, \cdots, \int_{t+(\iota-1)\mathcal{T}}^{t+\iota\mathcal{T}} \eta_i^\top \otimes w_i d\tau \right]^\top$$

In contrast to the hippocampus learning, we cannot compute the performance matrices S_{i+1} and R_{i+1} simultaneously because multiple solutions can be obtained, furthermore some solutions cause divergence in the hippocampus learning. To solve this issue we need to add constraints in the performance matrices so, the easiest constraint is to take into account the value of the cost $\xi(x_e, u_e)$ using the expert's trajectories. Therefore, we can collect ι samples of the expert's utility function $\Xi(x_e, u_e)$ and define the following matrices

$$I_{x\xi} = \left[x_e(t) \bar{\otimes} x_e(t), \cdots, x_e(t+\iota\mathcal{T}) \bar{\otimes} x_e(t+\iota\mathcal{T}) \right]^\top,$$
$$I_{u\xi} = \left[u_e(t) \otimes u_e(t), \cdots, u_e(t+\iota\mathcal{T}) \otimes u_e(t+\iota\mathcal{T}) \right]^\top.$$

Then, the ι samples of $\Xi(x_e, u_e)$ denoted as $\bar{\Xi}(x_e, u_e) \in \mathbb{R}^{1\times\iota} \subseteq \Xi(x_e, u_e)$ are written as

$$\text{vec}(\bar{\Xi}(x_e, u_e)) = I_{x\xi} \text{vech}(S_{i+1}) + I_{u\xi} \text{vec}(R_{i+1}) \tag{20}$$

The system of equations written in matrix form is solved as

$$\begin{aligned} \Sigma\Psi &= \Pi \\ \Psi &= (\Sigma^\top \Sigma)^{-1} \Sigma^\top \Pi, \end{aligned} \tag{21}$$

where

$$\Psi = \begin{bmatrix} \text{vech}(S_i) \\ \text{vec}(R_i) \end{bmatrix} \in \mathbb{R}^s, \quad s = \frac{1}{2}[n(n+1) + 2m^2]$$
$$\Sigma = \begin{bmatrix} I_{xs} & 2I_{uw} \\ I_{x\xi} & I_{u\xi} \end{bmatrix} \in \mathbb{R}^{2\iota\times s}$$
$$\Pi = \begin{bmatrix} -I_{xx}\text{vech}(\mathcal{P}) \\ \text{vec}(\Xi(x_e, u_e)) \end{bmatrix} \in \mathbb{R}^{2\iota}$$

By adding the constraint we are able to extract both performance matrices and guarantee convergence to their real values under the fulfilment of a PE condition. If the restriction is not added we can only estimate one performance matrix but convergence to their real values cannot be guaranteed.

The following theorem establishes the convergence of the proposed performance extraction algorithm as the number of episodes increases infinitely.

Theorem 1: The matrices S_{i+1} and R_{i+1} of the performance objective converge if the LS rule (21) restricts the possible solutions of the performance matrices as the number of episodes i increases. Here convergence mean that

$$\lim_{i\to\infty} S_i = \lim_{i\to\infty} S_{i+1} \quad \text{and} \quad \lim_{i\to\infty} R_i = \lim_{i\to\infty} R_{i+1}.$$

Furthermore, the constraint in (21) implies that the matrices S_{i+1} and R_{i+1} converge to the expert's performance matrices.

Proof: A Lyapunov recursions approach will be used to prove Theorem 1. The kernel matrix $\mathcal{P}_i$ of the inverse reinforcement learning part (18) satisfies the following Riccati equation

$$-S_{i+1} = A^\top \mathcal{P}_i + \mathcal{P}_i A - \mathcal{P}_i B R_{i+1}^{-1} B^\top \mathcal{P}_i. \tag{22}$$

Equivalently, the hippocampus learning model verifies the following Riccati equation in the episode $i+1$

$$-S_{i+1} = A^\top P_{i+1} + P_{i+1} A - P_{i+1} B R_{i+1}^{-1} B^\top P_{i+1}. \tag{23}$$

Substituting (23) in (22) gives

$$\begin{aligned} &A^\top P_{i+1} + P_{i+1} A - P_{i+1} B R_{i+1}^{-1} B^\top P_{i+1} \\ &= A^\top \mathcal{P}_i + \mathcal{P}_i A - \mathcal{P}_i B R_{i+1}^{-1} B^\top \mathcal{P}_i. \end{aligned} \tag{24}$$

The rule (15) updates the kernel matrix $\mathcal{P}$ in each episode i such that the error e_k is minimized, that is, $K_i \to \widehat{K}_e$, then $\lim_{i\to\infty} \frac{\partial E_i}{\partial P_i} = 0$. From the above result it follows that $\lim_{i\to\infty} \mathcal{P}_i = P_i$. Then

$$\begin{aligned} &A^\top P_{i+1} + P_{i+1} A - P_{i+1} B R_{i+1}^{-1} B^\top P_{i+1} \\ &= A^\top P_i + P_i A - P_i B R_{i+1}^{-1} B^\top P_i \pm P_i B R_i^{-1} B^\top P_i. \end{aligned} \tag{25}$$

Hence, for an infinite number of episode i the Riccati equation (25) is simplified to

$$\lim_{i\to\infty} (S_{i+1} - P_i B R_{i+1}^{-1} B^\top P_i) = \lim_{i\to\infty} (S_i - P_i B R_i^{-1} B^\top P_i), \tag{26}$$

Notice that the above equality has multiple solutions for the performance matrices S_i and R_i. However, the constraint (20) asserts that S_i and R_i have unique values in the limit such that the only way that (26) is fulfilled is when

$$\lim_{i\to\infty} S_{i+1} = \lim_{i\to\infty} S_i, \quad \lim_{i\to\infty} R_{i+1} = \lim_{i\to\infty} R_i.$$

This implies that the kernel matrix also converges, that is, $\lim_{i\to\infty} P_{i+1} = \lim_{i\to\infty} P_i$ and consequently the control gain converges, $\lim_{i\to} K_{i+1} = \lim_{i\to\infty} K_i$. This completes the proof. ■

IV. Simulation studies

The F-16 aircraft dynamics used in [21] was considered as case of study. The following matrices are used

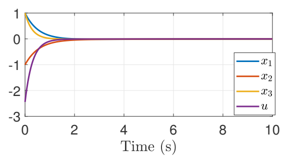
$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Assume we have a collection of data measurements of the states, control input, and utility function of an expert trajectory. These data are collected under the following performance matrices $S_e = 10I_3$ and $R_e = 1$. Fig. 1(a) and

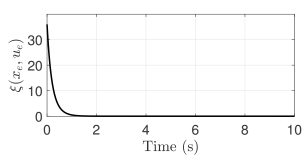
Fig.1(b) exhibit the expert's trajectories. The optimal kernel matrix and gain are

$$P_e = \begin{bmatrix} 13.7583 & 11.1733 & -0.5819 \\ 11.1733 & 13.8172 & -0.6719 \\ -0.5819 & -0.6719 & 2.3524 \end{bmatrix}$$
$$K_e = \begin{bmatrix} -0.5919 & -0.6719 & 2.3524 \end{bmatrix}.$$

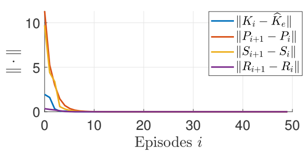
Assume measurements without noise. So, $\widehat{K}_e$ is equivalent to the expert's gain K_e. The initial performance matrices are set to $S_0 = I_3$ and $R_0 = 0.8$. The learning rate is manually tuned until the best convergence results are achieved. The final learning rate is $\alpha = 0.5$. Fig. 1 shows the results of the performance extraction algorithm.



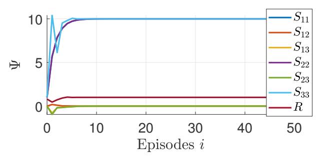
(a) States and input trajectories



(b) Utility function values



(c) Matrices convergence



(d) Performance matrices estimates

Fig. 1. Results for diagonal performance matrices

Fig. 1(c) shows the convergence results of the the gain matrix K_i, the kernel matrix P_i, the weight matrix S_i, and the weight matrix R_i. Convergence of the matrices means the any of the above matrices in episode $i+1$ is equal to its previous value in episode i. Convergence to the real expert's values can only be guaranteed by adding constraints. Fig. 1(d) shows the estimates of the each element of the performance matrices where we can observe the convergence of the estimates to the real expert's weight matrices, that is, $\lim_{i\to\infty} S_i = S_e$ and $\lim_{i\to\infty} R_i = R_e$.

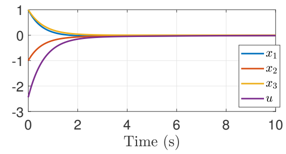
The approach is further verified by considering non-diagonal expert's performance matrices. Consider that the expert's data are obtained from an optimal control law using the next performance matrices

$$S_e = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 4 \end{bmatrix}, \quad R_e = 2.$$

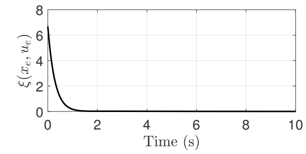
Fig. 2(a) and Fig. 2(b) exhibit the expert's trajectories under the new performance matrices. The optimal kernel matrix and control gain for the above matrices are

$$P = \begin{bmatrix} 8.8656 & 7.9481 & -0.1 \\ 7.9481 & 8.0622 & 0.1703 \\ -0.1 & 0.1703 & 1.4469 \end{bmatrix},$$
$$K_e = \begin{bmatrix} -0.05 & 0.0852 & 0.7235 \end{bmatrix}.$$

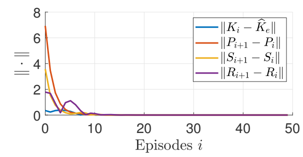
The same initial performance matrices are considered and also the same learning rate. Fig. 2 shows the results of the proposed performance extraction for non-diagonal performance matrices. Notice that we are able to extract the same performance matrices by using only the expert's data. Furthermore, the addition of the constraint associated to the values of the utility function avoids the multiple solutions problem.



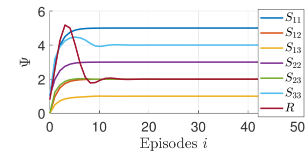
(a) States and input trajectories



(b) Utility function values



(c) Matrices convergence



(d) Performance matrices estimates

Fig. 2. Results for non-diagonal performance matrices

V. Conclusions

This paper reports a performance objective extraction using data from expert's trajectories. The algorithm is inspired in the hippocampus functionality to store past memories and experiences for experience transference to facilitate decision making. Two steps are considered: an hippocampus learning algorithm that estimates an optimal control policy from initial performance matrices and an extraction algorithm that obtain the improved performance matrices iteratively until they converge to the real expert's performance matrices. Unique solutions are guaranteed by adding constraints to the performance matrices. This is achieved by using the measurements of the expert's utility function. Simulation studies are carried out to verify the proposed algorithm under diagonal and non-diagonal performance matrices. Further work will investigate which other constraints can be used when the measurement of the utility function is not available. Furthermore, non-quadratic utility functions will be investigated to enhance the scope of the approach.

References

[1] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.

[2] E. de la Rosa and W. Yu, "Data-driven fuzzy modeling using restricted boltzmann machines and probability theory," *IEEE Transactions on System, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2316–2326, 2020.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2020.

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

[6] J. A. Flores-Campos, A. Perrusquía, L. H. Hernández-Gómez, N. González, and A. Armenta-Molina, "Constant speed control of slider-crank mechanisms: A joint-task space hybrid control approach," *IEEE Access*, vol. 9, pp. 65 676–65 687, 2021.

[7] D. Luviano and W. Yu, "Continuous-time path planning for multi-agents with fuzzy reinforcement learning," *Journal of Intelligent & Fuzzy Systems*, vol. 33, pp. 491–501, 2017.

[8] 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.

[9] 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.

[10] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of al and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

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

[12] B. Kiumarsi, G. V. Kyriakos, 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] 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.

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

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

[16] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[17] A. Perrusquía and W. Yu, "Continuous-time reinforcement learning for robust control under worst-case uncertainty," *International Journal of Systems Science*, pp. 1–15, 2020.

[18] Q. Xie, B. Luo, and F. Tan, "Discrete-time LQR optimal tracking control problems using approximate dynamic programming algorithm with disturbance," *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 716–721, 2013.

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

[20] K. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online policy iteration based algorithms to solve the continuous-time infinite horizon optimal control problem," *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2009.

[21] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, pp. 878–888, 2010.

[22] C. Wang, Y. Li, S. S. Ge, and T. H. Lee, "Optimal critic learning for robot control in time-varying environments," *IEEE Transactions on Neural Netowrks and Learning Systems*, vol. 26, no. 10, pp. 2301–2310, 2015.

[23] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 165–176, 2015.

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

[25] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.

[26] S. A. A. Rizvi and Z. Lin, "Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback," *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4670–4679, 2019.

[27] A. Perrusquía, "Robust state/output feedback linearization of direct drive robot manipulators: A controllability and observability analysis," *European Journal of Control*, 2022.

[28] Y. Park, "Inverse optimal and robust nonlinear attitude control of rigid spacecraft," *Aerospace Science and Technology*, vol. 28, no. 1, pp. 257–265, 2013.

[29] M. Johnson, N. Aghasadeghi, and T. Bretl, "Inverse optimal control for deterministic continuous-time nonlinear systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 2906–2913.

[30] E. N. Sanchez and F. Ornelas-Tellez, *Discrete-time inverse optimal control for nonlinear systems*. CRC Press, 2017.

[31] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," *Proceedings of the twenty-first International Conference on Machine Learning*, 2004.

[32] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad, "A framework for intelligence and cortical function based on grid cells in the neocortex," *Frontiers in neural circuits*, vol. 12, p. 121, 2019.

[40] 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.

[41] A. Perrusquía, "A complementary learning approach for expertise transference of human-optimized controllers," *Neural Networks*, vol. 145, pp. 33–41, 2021.

[42] 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.

[43] 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.

[44] 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.

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

[46] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, pp. 237–246, 2009.

[47] B. Luo, H.-N. Wu, Huand-Tingwen, and D. Liu, "Reinforcement learning solution for HJB equation arising in constrained optimal control problem," *Neural Networks*, vol. 71, pp. 150–158, 2015.

[48] A. Perrusquía, W. Yu, and A. Soria, "Position force/control of robot manipulators using reinforcement learning," *Industrial Robot*, vol. 46, no. 2, pp. 267–280, 2019.

[49] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuostime linear systems," *Automatica*, vol. 48, pp. 2850–2859, 2012.

[50] A. Perrusquía, W. Yu, and X. Li, "Multi-agent reinforcement learning for redundant robot control in task space," *International Journal of Machine Learning & Cybernetics*, vol. 12, pp. 231–241, 2021.

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

2022-10-28

Performance objective extraction of optimal controllers: a hippocampal learning approach

Perrusquía, Adolfo

IEEE

Perrusquia A, Guo W. (2022) Performance objective extraction of optimal controllers: a hippocampal learning approach. In: 2022 IEEE 18th International Conference on Automation Science and Engineering, 20-24 August 2022, Mexico City, Mexico

https://doi.org/10.1109/CASE49997.2022.9926671