IEEE CAA Journal of Automatica Sinica, Volume 10, Issue 1, January 2023, pp. 1-13
DOI:10.1109/JAS.2022.000000

Optimal Control of Nonlinear Systems using Experience Inference Human-Behavior Learning

Adolfo Perrusquía, *Member, IEEE*, Weisi Guo, *Senior Member, IEEE*

***Abstract*—Safety critical control is often trained in a simulated environment to mitigate risk. Subsequent migration of the biased controller requires further adjustments. In this paper, an experience inference human-behavior learning is proposed to solve the migration problem of optimal controllers applied to real-world nonlinear systems. The approach is inspired in the complementary properties that exhibits the hippocampus, the neocortex, and the striatum learning systems located in the brain. The hippocampus defines a physics informed reference model of the real-world nonlinear system for experience inference and the neocortex is the adaptive dynamic programming (ADP) or reinforcement learning (RL) algorithm that ensures optimal performance of the reference model. This optimal performance is inferred to the real-world nonlinear system by means of an adaptive neocortex/striatum control policy that forces the nonlinear system to behave as the reference model. Stability and convergence of the proposed approach is analyzed using Lyapunov stability theory. Simulation studies are carried out to verify the approach.**

***Index Terms*—Experience Inference, Nonlinear Systems, Linear Time-Variant (LTV) systems, Optimal control, Hippocampus learning system, Neocortex/Striatum Learning systems**

I. Introduction

OPTIMAL control [1], [2] is a well-known control philosophy that seeks the control law that minimizes a predefined cost function that defines a desired performance [3], [4]. There exist an extensive number of applications of optimal control applied to aerospace design problems, mathematical biology modelling, computer science, economics, social sciences, autonomous driving, robotics [5].

Optimal control has most of its advancements in linear systems [5], [6] by proposing different ways to solve an Algebraic Riccati equation (ARE) [7] either offline [8] or online [9]–[11]. Whilst the off-line solution is obtained from the well known linear quadratic regulator (LQR) [12]–[14], the on-line solution is obtained either by using adaptive dynamic programming (ADP) [5], [15], [16] or reinforcement learning (RL) algorithms [17]–[20].

For nonlinear systems, the solution of the optimal control problem is based on the solution of the underlying Hamilton-Jacobi-Bellman (HJB) equation [21], [22]. However, only local solutions can be obtained due to the intractability of the HJB equation [23]. There are some approaches that use a successive approximation approach (SAA) [24] to obtain a global solution of the optimal control problem. However, SAA methodology requires to compute in parallel i interconnected liner-time-variant (LTV) systems [25] such that system i has an equivalent performance to the nonlinear system. In a mathematical point of view, the SAA gives impressive results, however knowledge of the parameters and the nonlinear dynamic structure are required.

ADP and RL algorithms have been also used for nonlinear systems as an alternative to avoid knowledge of the nonlinear dynamics [26]–[28]. They key idea is to incorporate function approximators based on neural networks, radial basis functions, fuzzy systems, etc., to approximate the value function or control policy. Synchronous policy iteration and value iteration algorithms [9] and actor-critic structures [29] require partial knowledge of the system dynamics to compute optimal control policies for both linear and nonlinear systems. Q-learning also known as Action Dependent Heuristic Dynamic Programming (ADHDP) [30]–[32] is a data-driven method that does not require dynamic knowledge to obtain the optimal control policy. However, in a control perspective is only formulated for linear systems [33].

For physical systems such as mechanical, electrical or power systems, the ADP and RL methods are not trained in real-time because they require to fulfil a persistent of excitation (PE) condition [34], [35] to guarantee parameter convergence [36]. This excitation signal can cause severe damages to the environment or the system itself; instead, simulations are used to avoid this issue and guarantee safety in the training algorithm. In the simulation, the nonlinear system is modelled by either a mathematical model or a simulator under certain constraints and parameters which differ from the real nonlinear system [37], [38]. The optimal controller obtained from the simulation is then tested in the real non-linear system which, in most cases, will not behave as the expected performance and hence, the controller requires to be adjusted by an expert [39]. So, a biased control policy is obtained in terms of the simulations constraints. In this sense, the term biased control policy refers to the policies learned by the RL/ADP algorithms which guarantee an optimal/near optimal performance of the system trajectories under the simulations constraints but lacks of robustness and generalization in the real system. Throughout the paper, the term biased control policies refers to the previous definition.

The above issue is getting relevance in recent RL applications, where high accurate simulators are designed to obtain realistic control policies that can be applied to the real system and get almost the same simulator's performance. However, some of these simulators require high computational cost and

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.

A. Perrusquía and Weisi Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford, UK (e-mail: {Adolfo.Perrusquia-Guzman,Weisi.Guo}@cranfield.ac.uk).

they are difficult to modify. In view of the above, one of the main challenges is to design an algorithm that gives solution to the optimal control problem of nonlinear systems without knowledge of the system's parameters and that guarantees unbiased control policies for robustness and generalization.

In this context, the model that the RL/ADP algorithm uses for training can be regarded as a prior knowledge of the real system that can be used by an inference algorithm for experience transference. This process is quite similar in how humans infers their prior knowledge to develop a task to new similar tasks by adapting their actions. This kind of process inspires the novel perspective known as human-behavior learning.

Human-behavior learning (HBL) [40], [41] is a recent technique that its main aim is to model how human learns through the effective combinations of different sources of knowledge and experiences [42]–[44]. This can be achieved by exploiting the complementary properties [45] of the main learning systems in the brain: the hippocampus, the neocortex, and the striatum. Despite each learning system executes different learning procedures, they are strongly correlated and co-dependent. The hippocampus is responsible of fast learning architectures related to memory [46] and experience [47], [48]. This system offers an explainable knowledge of how a task should be performed [49], [50]. On the other hand, the neocortex is characterized by slow-learning architectures that serve for the acquisition of new information that is subsequently organized and distributed in different structures [51] to realize a certain task. The striatum relates the hippocampus and the neocortex information for decision making [52]. In most cases, it is difficult to separate each learning system since they are strongly correlated but it is important to distinguish their main functionalities.

In a control sense, the hippocampus serves as a reference model that will teach the neocortex how the nonlinear system should behave. In other words, the HBL infers the experience (desired performance) of a completely independent system and forces the real nonlinear system to exhibit the same performance. In contrast to previous optimal control methodologies, this approach finds the optimal control policy for the reference model using any ADP/RL algorithm such that the PE condition can be easily fulfilled. On the other hand, the real-world nonlinear system is controlled by a neocortex/striatum control policy to give state feedback and to establish, in an adaptive way, a relationship with the previous experience for the decision making of the final control policy [53].

Based on the above facts, this paper proposes an experience inference HBL algorithm that solves the optimal control problem of nonlinear systems and guarantee robustness and generalization for model uncertainty. The main contributions of this paper are: i) a novel optimal control solution of nonlinear systems based on a human-behavior learning approach, ii) the algorithm does not require to solve a HJB equation and does not require knowledge of the parameters of the real system, iii) the final control policy applied to the real system is unbiased to the simulation constraints.

This paper is organized as follows. Section II introduces the optimal control problem of nonlinear systems, followed by an alternative notation of nonlinear systems based on state-dependent coefficient matrices. Section III discusses the proposed HBL for experience inference and defines the main elements of each learning system. Section IV exhibits the simulation studies using a 2-degree of freedom robot with/without gravitational torques vector to verify the effectiveness of the approach. Section V concludes the paper.

Throughout this paper, $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}^+$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the spaces of natural numbers, real numbers, positive real numbers, positive integers, real n-vectors, and real $n \times m$-matrices, respectively; $\|\cdot\|$ denotes the Euclidean norm, where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$ and $n, m \in \mathbb{N}$.

II. Optimal Control of Nonlinear Systems

The following nonlinear system is considered [21]

$$\dot{x} = f(x, v), \;\; x(0) = x_0, \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the state vector, $v \in \mathbb{R}^m$ is the control input, and the mapping $f(x, v) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the unknown nonlinear dynamics. Suppose that f is differentiable at $x = 0$ and $f(0, 0) = 0$ for all t. Furthermore, without loss of generality it is assumed that f is Lipschitz.

It is always possible to make (1) linear in the control by increasing the dimensionality of the state-space by the dimension of the control v [24]. Thus, if we set $\dot{v} = u$ where $u \in \mathbb{R}^n$, then the dynamics (1) can be written as

$$\dot{X} = f_1(X) + g_1(X)u, \;\; X(0) = X_0, \tag{2}$$

where $X = [x^\top, v^\top]^\top \in \mathbb{R}^{n+m}$ is the new coordinates, $f_1(X) : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$, and $g_1(X) : \mathbb{R}^{n+m} \to \mathbb{R}^{(n+m)\times m}$ are the nonlinear dynamics associated to the new coordinates X. In the sequel of the paper it is assumed nonlinear systems of the form (2) which satisfies the following nonlinear model and dimensions [54]

$$\dot{x} = f(x) + g(x)u, \;\; x(0) = x_0, \tag{3}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f(x) : \mathbb{R}^n \to \mathbb{R}^n$, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$.

A. Optimal Control Design

The following value function/cost index is used for the design of the optimal control of the nonlinear system [33]

$$V(x) = \int_t^\infty (x^\top S_1 x + u^\top R_1 u) d\tau, \tag{4}$$

where $S_1 \in \mathbb{R}^{n\times n}$ and $R_1 \in \mathbb{R}^{m\times m}$ are positive semi-definite and positive definite weight matrices, respectively. The infinitesimal version of (4) is the so-called nonlinear Lyapunov equation [54] which can be also defined by the Hamiltonian of the system

$$\begin{aligned} H(x, u, \nabla V) = \;& \nabla^\top V(f(x) + g(x)u) \\ & + x^\top S_1 x + u^\top R_1 u = 0. \end{aligned} \tag{5}$$

where $\nabla V = \frac{\partial V}{\partial x}$. The optimal value function $V^*(x)$ defined by

$$V^*(x) = \min_u \left(\int_t^\infty (x^\top S_1 x + u^\top R u) d\tau \right),$$

satisfies the HJB equation

$$0 = \min_u [H(x, u, \nabla V^*].$$

Assuming that the optimal control exists, then the optimal control policy is computed by taking the stationary condition $\frac{\partial H}{\partial u} = 0$ which yields

$$u^* = -\frac{1}{2} R_1^{-1} g^\top(x) \nabla V^*. \tag{6}$$

Substituting the control policy (6) in the Hamiltonian (5) gives the following Hamilton-Jacobi-Bellman (HJB) equation [54]

$$\nabla^\top V f(x) \quad + x^\top S_1 x - \frac{1}{4} \nabla^\top V g(x) R_1^{-1} g^\top(x) \nabla V = 0. \tag{7}$$

The HJB equation (7) is hard to solve even intractable despite the nonlinear dynamics $f(x)$ and $g(x)$ are known in advance [54]. To overcome the above issue, this paper proposes a HBL algorithm for experience inference [41] as a local solution of the optimal control problem of nonlinear systems. Before developing the proposed HBL approach we need to introduce an alternative notation for the nonlinear dynamics (3).

B. From Nonlinear to Linear time-Variant Systems

The nonlinear dynamics (3) can be written as a linear time-variant (LTV) system [25] with state disturbances of the form

$$\dot{x} = A_1(x)x + B_1(x)u + A_2(x), \;\; x(0) = x_0, \tag{8}$$

for non-unique state dependent coefficient (SDC) matrices $A_1(x) : \mathbb{R}^n \to \mathbb{R}^{n \times n}$, $A_2(x) : \mathbb{R}^n \to \mathbb{R}^n$, and $B_1(x) : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ [55]. Assume that the pair $(A_1(x), B_1(x))$ is controllable for any state vector x, that is, $\mathcal{C}[A_1(x), B_1(x)] = n \; \forall x$, where $\mathcal{C}[\cdot]$ denotes the controllability matrix.

The LTV system with state disturbance (8) can be written as a standard LTV system by incorporating a new stable dynamics of the form

$$\dot{y} = -\alpha y, \;\; y(0) = y_0 \neq 0, \tag{9}$$

for some $0 < \alpha \ll 1$ and $y \in \mathbb{R}$. Define the new state vector as $z = [x^\top \; y]^\top \in \mathbb{R}^{n+1}$, then the new LTV dynamics is

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= \begin{bmatrix} A_1(x) & \frac{1}{y} A_2(x) \\ 0_{1\times n} & -\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} B_1(x) \\ 0_{1\times m} \end{bmatrix} u \\ \dot{z} &= A(z)z + B(z)u, z(0) = z_0 = [x_0^\top, y_0]^\top, \end{aligned} \tag{10}$$

where $A(z) : \mathbb{R}^{n+1} \to \mathbb{R}^{(n+1)\times(n+1)}$ and $B(z) : \mathbb{R}^{n+1} \to \mathbb{R}^{(n+1)\times m}$.

Remark 1: The term $\frac{1}{y}$ serves to linearly parameterize $A_1(x)$ and $A_2(x)$ in terms of the new coordinates z. The new state y converges exponentially to zero with decay factor α, that is, $y(t) = e^{-\alpha t} y_0$. Hence, a small decay factor α must be used to avoid an unbounded matrix $A(z)$.

Remark 2: The pair $(A(z), B(z))$ is not controllable, that is, $\mathcal{C}[A(z), B(z)] = n$ because the new dynamics (9) is not controllable. However, it is still possible to stabilize the states of (8) using any feedback control policy based on the new state vector z.

Remark 3: A small final value $y_f \neq 0$ can be added to (9) such that the trajectories of y never converges to zero, that is, the following constraint can be added $y(t) = \max(e^{-\alpha t} y_0, y_f)$ and hence, the matrix $A(z)$ never diverges. Optionally, the state y can be fixed to a small value such that the effect of matrix $A_2(x)$ is not attenuated. In a worst-case uncertainty scenario [39], a small value of y allows the control algorithm to stabilize a set of nonlinear systems with disturbances of radius $\mu = \|A_2(x)/y\|$.

The cost index (4) for the optimal control is rewritten for the LTV system (10) as

$$V(z) = \int_t^\infty (z^\top S z + u^\top R u) d\tau, \tag{11}$$

where $S \in \mathbb{R}^{(n+1)\times(n+1)}$ and $R \in \mathbb{R}^{m \times m}$ are positive semi-definite and positive definite matrices. Notice that

$$S = \begin{bmatrix} S_1 & 0_{n\times 1} \\ 0_{1\times n} & S_y \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

where $S_y \geq 0 \in \mathbb{R}$. The value functions (11) and (4) are equivalent when $S_y = 0$. If $S_y \neq 0$, is easy to verify that

$$\lim_{t\to\infty} V(z_r) = \lim_{t\to\infty} V(x_r)$$

since $\lim_{t\to\infty} y(t) = 0$. If the final condition y_f is used then

$$\lim_{t\to\infty} V(z_r) = \lim_{t\to\infty} V(x_r) + \varepsilon$$

where $\lim_{t\to\infty} y^\top S_y y = y_f^\top S_y y_f = \varepsilon > 0$ is a small term added by the new dynamics which is practically neglected for small y_f and S_y. It is shown in [8] that the optimal value function is quadratic in terms of the state vector [56], [57] so that

$$V(z) = z^\top P(z) z,$$

for some positive definite kernel matrix $P(z) = P^\top(z) > 0 \in \mathbb{R}^{(n+1)\times(n+1)}$ which its solution of the following matrix differential Ricatti equation (MDRE) [57],

$$\begin{aligned} -\dot{P}(z) = \;& A^\top(z)P(z) + P(z)A(z) + S \\ & -P(z)B(z)R^{-1}B^\top(z)P(z). \end{aligned} \tag{12}$$

Remark 4: Previous studies [58], [59] use the extended linearization control technique [60] such that matrices $A(z)$ and $B(z)$ are treated as constant matrices in any time instance t and hence, $P(z)$ will be also constant and can be computed using the ARE. In this work, we will use the MDRE instead of the ARE and the extended linearization technique.

Notice that the MDRE (12) is numerically easier than the HJB equation (6) and hence, it is possible to find a local solution of the optimal control problem. The optimal control policy that minimizes (11) is given by

$$u^* = -R^{-1}B^\top(z)P(z)z. \tag{13}$$

The main drawback of this approach is that it requires knowledge of the parameters and a dynamic model of the nonlinear system [61]. This strong assumption in most real cases is not satisfied because we only have access to parameter estimates and an approximate model structure of the real-world nonlinear system dynamics [62]. Therefore, the optimal control policy of each sequence is biased and cannot guarantee an optimal performance of the closed-loop trajectories of the real-world nonlinear system.

In the next section, the proposed HBL is developed to guarantee unbiased control policies and optimal performance of the trajectories of the real-world nonlinear system.

III. HBL for Experience Inference

In contrast to classical optimal control architectures, this approach is inspired in the inference property that exhibits humans to develop similar tasks [42]. That is, humans are capable to infer previous knowledge to new activities and adapt its knowledge to achieve a similar or equivalent performance in the new task. For example, if a human knows how to drive a car then the human is also capable to drive a truck. Hence, the main contribution of this paper is to develop an experience inference HBL algorithm that is able to achieve online unbiased control policies and guarantee optimal performances of the closed-loop trajectories of the unknown real-world nonlinear system.

Fig. 1 shows the proposed HBL approach for experience inference. The scheme is based on two main learning systems: the hippocampus and the neocortex/striatum learning systems. The hippocampus is modelled as a desired reference model that represents previous experience in how the nonlinear system should behave. The neocortex/striatum learning systems compute a new control policy based on the states of the real nonlinear system and the hippocampus policy, such that the closed-loop system trajectories behave as the hippocampus reference model.

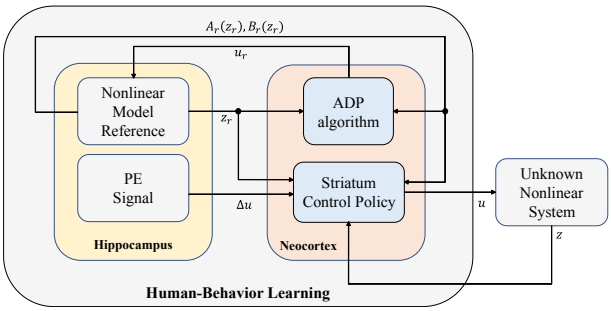


Fig. 1. HBL Experience Inference Algorithm. The hippocampus reference model can be trained either offline or online and defines a desired performance that we want to infer to the real-world nonlinear system. The control policy of the hippocampus is inferred by means of a striatum/neocortex algorithm which relates online data and experience in an adaptive fashion such that the closed-loop trajectories of the real-world non linear system behaves as the hippocampus reference model.

A. Hippocampus learning System

The hippocampus models experience, memory, and previous knowledge which permits to infer a desired performance [63]. This desired performance is given by a known reference model constructed from the estimates and the model structure of the unknown nonlinear system dynamics. The reference model is given by the next LTV system with state disturbance

$$\dot{x}_r = A_r^1(x_r)x_r + B_r^1(x_r)u_r + A_r^2(x_r), \tag{14}$$

where $A_r^1(x_r) \in \mathbb{R}^{n\times n}$, $A_r^2(x_r) \in \mathbb{R}^n$, and $B_r^1(x_r) \in \mathbb{R}^{n\times m}$ are known SDC matrices, $x_r \in \mathbb{R}^n$ and $u_r \in \mathbb{R}^m$ denote the state and control input of the reference model. System (14) can be written as in (10) as

$$\begin{aligned}\begin{bmatrix}\dot{x}_r\\ \dot{y}\end{bmatrix} &= \begin{bmatrix}A_r^1(x_r) & \frac{1}{y}A_r^2(x_r)\\ 0_{1\times n} & -\alpha\end{bmatrix}\begin{bmatrix}x_r\\ y\end{bmatrix} + \begin{bmatrix}B_r^1(x_r)\\ 0_{1\times m}\end{bmatrix}u_r,\\ \dot{z}_r &= A_r(z_r)z_r + B_r(z_r)u_r,\ \ z_r(0) = z_0,\end{aligned} \tag{15}$$

where $z_r = [x_r^\top, y]^\top \in \mathbb{R}^{n+1}$, $A_r(z_r) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{(n+1)\times(n+1)}$ and $B_r(z_r) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{(n+1)\times m}$. The key idea of the approach is to obtain an unbiased and independent optimal control policy for the hippocampus reference model (14) using any ADP or RL technique such that it gives a notion in how the real system must behave. The neocortex/striatum learning systems are responsible to infer the hippocampus experience and adapt the final control policy.

In addition, the hippocampus acquires information by exploring all the possible combinations between states and actions. This can be modeled as the fulfillment of a persistency of excitation (PE) condition (refer to [35]).

In this paper, the MDRE (12) is used to compute the kernel matrix $P(z_r)$ associated to the hippocampus reference model (15) and hence, the hippocampus control policy is computed by (13).

B. Neocortex/Striatum Learning System

Recall that the neocortex and the striatum learning systems are co-dependent systems (also the hippocampus) which are responsible for learning and decision making. In this approach, the neocortex and the striatum are executed simultaneously to achieve the same hippocampus performance in an adaptive fashion.

1) Hippocampus-Neocortex Co-dependence: First, we need to establish a co-dependence of the hippocampus and the neocortex. Whilst the hippocampus gives a model structure and parameter estimates as previous knowledge, the neocortex facilitates the tools to learn and compute the hippocampus control policy. In other words, the neocortex solves the nonlinear optimal control problem using the hippocampus reference model. Similarly to (11), the value function written in terms of the reference model is

$$V(z_r) = \int_t^\infty (z_r^\top S z_r + u_r^\top R u_r)d\tau, \tag{16}$$

for some positive semi-definite and definite matrices S and R of appropriate dimension. The hippocampus optimal control policy that minimizes (16) is

$$\begin{aligned}u_r^* &= -K(z_r)z_r\\ &= -R^{-1}(t)B^\top(z_r)P(z_r)z_r,\end{aligned} \tag{17}$$

for some stabilizing gain $K(z_r) \in \mathbb{R}^{m\times n}$ and $P(z_r) \in \mathbb{R}^{(n+1)\times(n+1)}$ is a positive definite kernel matrix which its solution of the next MDRE

$$\begin{aligned}-\dot{P}(z_r) = \ & A_r^\top(z_r)P(z_r) + P(z_r)A_r(z_r) + S\\ & -P(z_r)B_r^\top(z_r)R^{-1}B_r^\top(z_r)P(z_r).\end{aligned} \tag{18}$$

In contrast to linear time-invariant (LTI) systems [5], the control gain $K(z_r)$ is adaptive due to the nature of the MDRE (18) and the non-linearity of the reference model.

The following theorem establishes the exponential convergence of the hippocampus reference model trajectories under the extended coordinates z_r and optimal control policy (17).

Theorem 1: Consider the closed-loop system between the hippocampus reference model (15) and the optimal control (17). If the reference model (15) is Lipschitz then, the trajectories of reference model converges exponentially to zero as $t \to \infty$.

Proof: Consider the following Lyapunov function

$$V_1 = z_r^\top P(z_r) z_r, \tag{19}$$

The time-derivative of (19) along the trajectories of (15) and the optimal control policy (17) is

$$\begin{aligned}\dot{V}_1 =& 2z_r^\top P(z_r)(A_r(z_r) - B_r(z_r)R^{-1}B_r^\top(z_r)P(z_r))z_r \\ &+ z_r^\top \dot{P}(z_r) z_r \\ =& - z_r^\top (S + P(z_r)B_r(z_r)R^{-1}B_r^\top(z_r)P(z_r))z_r, \\ =& - z_r^\top \Omega(z_r) z_r \leq -\lambda_{\min}(\Omega(z_r)P^{-1}(z_r))V_1 \\ =& - \lambda_{\min}(\Psi(z_r))V_1, \end{aligned} \tag{20}$$

where $\Omega(z_r) = S + P(z_r)B_r(z_r)R^{-1}B_r^\top(z_r)P(z_r)$ and $\Psi(z_r) = \Omega(z_r)P^{-1}(z_r)$. The solution of (20) is

$$V_1(t) = e^{-\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau} V_1(0).$$

So,

$$\begin{aligned}\lambda_{\min}(P(z_r))\|z_r\|^2 \leq z_r^\top P(z_r) z_r = V(t) \\ \leq e^{-\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau} V_1(0) \\ = e^{-\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau} z_r^\top P(z_r) z_r \\ \leq \lambda_{\max}(P(z_r)) e^{-\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau} \|z_r(0)\|^2\end{aligned}$$

Hence, the states of (15) converges exponentially to zero by a rate of $\frac{1}{2}\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau$ and satisfies

$$\|z_r(t)\| \leq \sqrt{\frac{\lambda_{\max}(P(z_r))}{\lambda_{\min}(P(z_r))}} e^{-\frac{1}{2}\int_0^t \lambda_{\min}(\Psi(z_r(\tau)))d\tau} \|z_r(0)\|. \tag{21}$$

It is important to mention that the new dynamics y converges exponentially to zero but is not controllable. This causes that matrix $A(z_r)$ to not meet the Lipschitz condition. To see this fact more clearly let write the kernel matrix $P(z_r)$ as

$$P(z_r) = \begin{bmatrix} P_{xx} & P_{xy} \\ P_{xy}^\top & P_{yy} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)},$$

where $P_{xx} \in \mathbb{R}^{n\times n}$ is the kernel matrix associated to (4), $P_{xy} \in \mathbb{R}^n$ are the cross-terms between the states x_r and y, $P_{yy} \in \mathbb{R}$ is the kernel element associated to the stable state y. If $A_r(z_r)$ is not Lipschitz then the term P_{yy} will diverge because $\lim_{y\to 0} 1/y = \infty$. On the other hand, if $A_r(z_r)$ is Lipschitz then P_{xy} remain bounded. Theoretically, the term P_{yy} is not used in the design of the optimal control policy such that (15) can be stabilized. However, it is important to ensure boundedness of $A_r(z_r)$ to avoid any numerical issue. **Remark 3** is used to maintain boundedness of both matrix $A(z_r)$ and thus $P(z_r)$. This completes the proof. ■

In view of the above, the closed-loop system between the reference model (15) and the optimal control policy (17) can be written as

$$\dot{z}_r = (A_r(z_r) - B_r(z_r)K(z_r))z_r = A_c(z_r)z_r. \tag{22}$$

where $A_c(z_r) = A_r(z_r) - B_r(z_r)K(z_r)$. The above closed-loop system is stable and exhibits an adaptive optimal performance.

2) Experience Inference: First of all, we need to consider the extended dynamics (10) of the real-world nonlinear system to have equivalent dimensions. Assume full dynamic knowledge, then it is possible to find the striatum control policy $u^* = -Lz$ for some stabilizing gain $L \in \mathbb{R}^{m\times(n+1)}$ such that the nonlinear system behaves as the hippocampus reference model. For this, define the estimation error $e \in \mathbb{R}^{n+1}$ as

$$e = z - z_r. \tag{23}$$

The closed-loop error dynamics between (10) and (22) gives

$$\begin{aligned}\dot{e} &= A(z)z + B(z)u - A_c(z_r)z_r \\ &= (A(z) - B(z)L)z - A_c(z_r)z_r \\ &= A_s(z)z - A_c(z_r)z_r, \end{aligned} \tag{24}$$

where $A_s(z) = A(z) - B(z)L$. Since (10) satisfies the Lipschitz condition, then the following inequalities are satisfied

$$\begin{aligned}\|A_s(z) - A_s(z_r)\| \leq k_1\|z - z_r\|, \\ \|A_c(z) - A_c(z_r)\| \leq k_2\|z - z_r\|,\end{aligned}$$

for some $k_1, k_2, \in [0, 1)$. Then, by defining $k_3 = \max\{k_1, k_2\}$ it follows that

$$\begin{aligned}&\|(A_s + A_c)(z) - (A_s + A_c)(z_r)\| \\ &= \|A_s(z) - A_c(z) + A_s(z_r) - A_c(z_r)\| \\ &\leq \|A_s(z) - A_s(z_r)\| + \|A_c(z) - A_c(z_r)\| \\ &\leq k_1\|z - z_r\| + k_2\|z - z_r\| \leq 2k_3\|e\|\end{aligned}$$

Therefore, is easy to verify that

$$\|A_s(z) - A_c(z_r)\| \leq k_3\|e\|.$$

However, the nonlinear dynamics is unknown. Therefore, the striatum policy is modified by an estimate control policy of the form

$$u = -\widehat{L}z, \tag{25}$$

where $\widehat{L} \in \mathbb{R}^{m\times(n+1)}$ is an estimate of the optimal control gain L. Then, the closed-loop error dynamics between (10) and (22) under the control policy (25) is

$$\begin{aligned}\dot{e} &= A(z)z - B(z)\widehat{L}z - A_c(z_r)z_r \\ &= A_s(z)z - A_c(z_r)z_r - B(z)\widetilde{L}z, \end{aligned} \tag{26}$$

where $\widetilde{L} = \widehat{L} - L \in \mathbb{R}^{m\times(n+1)}$ is the gain error.

Remark 5: The SDC matrix $B(z)$ can be written in terms of the known SDC matrix $B_r(z)$ as $B(z) = B_r(z)\Lambda$ for some unknown positive definite matrix $\Lambda \in \mathbb{R}^{m\times m}$. Notice that matrix B_r is valued in the nonlinear states z instead of the reference states z_r.

Then, the closed-loop error dynamics (26) is rewritten as

$$\dot{e} = A_c(z_r)e + (A_s(z) - A_c(z_r))z - B_r(z)\Lambda\widetilde{L}z. \quad (27)$$

The following theorem establishes the stability and convergence of the experience inference algorithm.

Theorem 2: Consider the closed-loop error dynamics (27). If the control gain $\widehat{L}$ is updated as

$$\dot{\widetilde{L}} = \dot{\widehat{L}} = \Gamma B_r^\top(z)\mathcal{P}(z_r)ez^\top, \quad (28)$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is a positive definite matrix gain and $\mathcal{P}(z_r) \in \mathbb{R}^{(n+1)\times(n+1)}$ is the solution of the Lyapunov differential equation

$$-\dot{\mathcal{P}}(z_r) = A_c^\top(z_r)\mathcal{P}(t) + \mathcal{P}(z_r)A_c(z_r) + Q, \quad (29)$$

for some positive definite matrix $Q \in \mathbb{R}^{(n+1)\times(n+1)}$ that satisfies

$$\lambda_{\min}(Q) \geq 2k_3\lambda_{\max}(\mathcal{P}(z_r))\|z\| + \rho, \quad (30)$$

where $\rho > 0$. Then $\widetilde{L}$ remains bounded and e converges to zero which implies that $z \to z_r$.

Proof: Consider the following Lyapunov function candidate

$$V_2 = e^\top\mathcal{P}(z_r)e + \text{tr}\{\widetilde{L}^\top\Lambda\Gamma^{-1}\widetilde{L}\}. \quad (31)$$

The time-derivative of (31) along the trajectories of (27) under the update rule (28) and condition (30) gives

$$\begin{aligned}
\dot{V}_2 =& 2e^\top\mathcal{P}(z_r)(A_c(z_r)e + (A_s(z) - A_c(z_r))z \\
&- B_r(z)\Lambda\widetilde{L}z) + e^\top\dot{\mathcal{P}}(z_r)e + 2\text{tr}\{\widetilde{L}^\top\Lambda\Gamma^{-1}\dot{\widetilde{L}}\} \\
=& -e^\top Qe + 2e^\top\mathcal{P}(z_r)(A_s(z) - A_c(z_r))z \\
\leq& -\lambda_{\min}(Q)\|e\|^2 + 2k_3\lambda_{\max}(\mathcal{P}(z_r))\|z\|\|e\|^2 \\
=& -(\lambda_{\min}(Q) - 2k_3\lambda_{\max}(\mathcal{P}(z_r))\|z\|)\|e\|^2 \\
\leq& -\rho\|e\|^2. \quad (32)
\end{aligned}$$

From (32), it is clear that e is an $\mathcal{L}_\infty$ function and $V_2(0) \geq V_2$. On the other hand, boundedness of e implies boundedness of z and z_r. Batbalat's lemma [64] is applied to prove convergence of e to zero. Integrating (32) gives

$$V_2(t) - V_2(0) \leq -\int_0^t \rho\|e\|^2 d\tau.$$

The next inequality follows from the last result

$$\int_0^t \|e\|^2 d\tau \leq \frac{V_2(0)}{\rho} < \infty. \quad (33)$$

From (33), it follows that e is an $\mathcal{L}_2$ function. Boundedness of the error gain $\widetilde{L}$, z, and z_r in (32), and the Lipschitz condition in $A_c(z_r)$ and $A_s(z)$, allow concluding that $\dot{e}$ is an $\mathcal{L}_\infty$ function. Applying Barbalat's lemma permits concluding that e converges to zero and hence $z \to z_r$. This completes the proof. ∎

Remark 6: Notice that the experience inference algorithm has a model reference adaptive control (MRAC) structure [58]. However, there are some key points that differentiate them. For instance,

- MRAC uses a stable reference model to guarantee the desired reference tracking [59].
- In most cases, the reference model is not controlled, that is, is an open-loop system with stable dynamics and bounded desired reference trajectory.
- Typically the reference model does not provide information of the physics of the real system.
- Indirect MRAC requires to compute estimates of the dynamics of the system.
- Direct MRAC usually computes two different gain matrices to guarantee that the real system behaves as the reference model.
- Nonlinear MRAC assumes knowledge of the nonlinear functions of the real system to perform a feedback linearization controller.

On the other hand,

- The reference model of the HBL is stabilizable by the control input u_r and hence, the open-loop model is not necessarily stable.
- The reference model is controlled by a RL/ADP algorithm to obtain an optimal performance which will be inferred to the real system.
- The reference model is constructed by a model of the real system and hence, informs the physics of the real system.
- Does not require knowledge of the real system dynamics and parameters. Instead, it uses the reference model dynamics.
- The HBL computes only one control gain to force the states of the real system to behave as the states of the reference model.
- Feedback linearization cannot be applied since the dynamics of the real system and the reference model are different.

IV. Simulation Studies

To verify the proposed approach, a 2-DOF planar robot [65] was considered with and without gravity torques vector. Here I_2 denote a 2×2 identity matrix and 0_2 denote a 2×2 matrix with zeros. The simulations were made in Matlab/Simulink 2021a with a sampling time of 1ms.

A. Without Gravity torques vector

The dynamic model of a horizontal 2-DOF planar robot without gravitational torques vector is

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau,$$

where $M(q) \in \mathbb{R}^{2\times 2}$ denotes the symmetric and positive definite inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{2\times 2}$ denotes the Coriolis and centripetal forces matrix, $\tau = [\tau_1, \tau_2]^\top \in \mathbb{R}^2$ is the driven torque vector, and $q, \dot{q}, \ddot{q} \in \mathbb{R}^2$ are the joint position, velocity, and acceleration vectors where $q = [q_1, q_2]^\top$. Define the state vector $x = [q^\top, \dot{q}^\top]^\top \in \mathbb{R}^4$, where $x_1 = q_1$, $x_2 = q_2$, $x_3 = \dot{q}_1$, and $x_4 = \dot{q}_2$. Then, the robot dynamics can be written as in (3) as

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ -M^{-1}(q)C(q,\dot{q})\dot{q} \end{bmatrix} + \begin{bmatrix} 0_2 \\ M^{-1}(q) \end{bmatrix} u,$$

where $u = \tau$. The above nonlinear system can be written in its LTV form (8) as follows

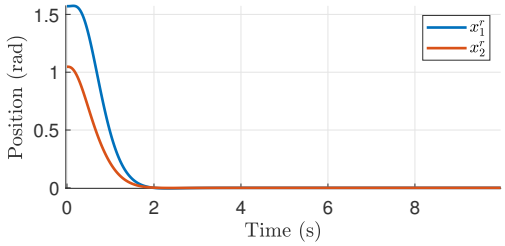
$$\dot{x} = \left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0_2 & 0_2 \\ 0_2 & -M^{-1}(q)C(q,\dot{q}) \end{bmatrix} \right\} x + \begin{bmatrix} 0_2 \\ M^{-1}(q) \end{bmatrix} u = A_1(x)x + B_1(x)u.$$

The hippocampus reference model is constructed from the Euler-Lagrange formulation, that is, the reference model is given by

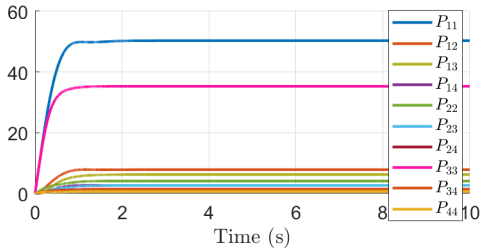
$$M_r(q_r)\ddot{q}_r + C_r(q_r,\dot{q}_r)\dot{q}_r = u_r,$$

where $M_r(q_r), C_r(q_r,\dot{q}_r) \in \mathbb{R}^{2\times 2}$ denote the inertia and Coriolis matrix, respectively; $q_r, \dot{q}_r, \ddot{q}_r$ denote the joint position, velocity, and acceleration vectors of the reference model, and $u_r \in \mathbb{R}^2$ is the control input. Here $q_r = [q_1^r, q_2^r]^\top$ and the inertia and Coriolis matrices satisfy

$$M_r(q_r) = \begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix},$$
$$C_r(q_r,\dot{q}_r) = \begin{bmatrix} -C_1\dot{q}_2^r & -C_1(\dot{q}_1^r + \dot{q}_2^r) \\ -C_1\dot{q}_1^r & 0 \end{bmatrix},$$
$$M_{11} = m_1 l_{c_1}^2 + m_2(l_1^2 + l_{c_2}^2 + 2l_1 l_{c_2}\cos(q_2^r)) + I_1 + I_2,$$
$$M_{12} = m_2 l_1 l_{c_2}\cos(q_2^r) + m_2 l_{c_2}^2 + I_2,$$
$$M_{22} = m_2 l_{c_2}^2 + I_2$$
$$C_1 = m_2 l_1 l_{c_2}\sin(q_2^r).$$



(a) States tracking x_r



(b) Kernel matrix $P(x_r)$ elements

Fig. 2. Hippocampus-Neocortex results

Notice that the matrices M and C of the real robot are different to the structure of M_r and C_r of the reference model in terms of both the nonlinear terms and their parameters.

Then, the hippocampus reference model written as a LTV system is

$$\dot{x}_r = \left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0_2 & 0_2 \\ 0_2 & -M_r^{-1}(q_r)C_r(q_r,\dot{q}_r) \end{bmatrix} \right\} x_r + \begin{bmatrix} 0_2 \\ M_r^{-1}(q_r) \end{bmatrix} u_r = A_1^r(x_r)x_r + B_1^r(x_r)u_r,$$

where $x_r = [q_r^\top, \dot{q}_r^\top]^\top \in \mathbb{R}^4$. Assume that the real-world nonlinear system has the same dynamic structure, however the parameters of the hippocampus reference model and the real parameters of the robot were completely different. The parameters of the reference model and the robot are given in Table I. Both the hippocampus reference model are initialized in the

TABLE I
PARAMETERS OF THE 2-DOF ROBOT

Parameter	Reference model	Real system
m_1 (kg)	1	2
m_2 (kg)	1	1.4
l_1 (m)	0.5	0.8
l_2 (m)	0.5	0.8
l_{c_1} (m)	0.25	0.4
l_{c_2} (m)	0.25	0.4
I_1 (kgm^2)	0.08	0.5
I_2 (kgm^2)	0.08	0.1

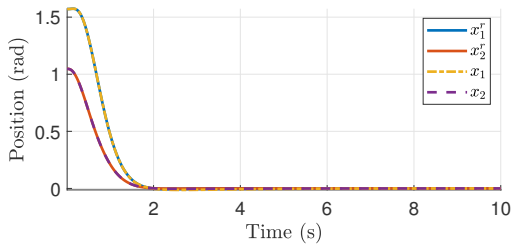
same initial conditions, that is, $q(0) = q_r(0) = [\frac{\pi}{2}, \frac{\pi}{3}, 0, 0]^\top$. Several weight matrices are tested for the design of the optimal control policy (13). The best weight matrices are set to $S = \text{diag}\{100, 100, 10, 10\}$ and $R = I_2$. The hippocampus results are exhibited in Fig. 2.

The results show a relative fast convergence to zero of the states x_r (see Fig. 2(a)) and convergence to the optimal kernel matrix $P(x_r)$ (see Fig. 2(b)). This desired performance is inferred to the nonlinear system via the necortex/striatum learning systems.

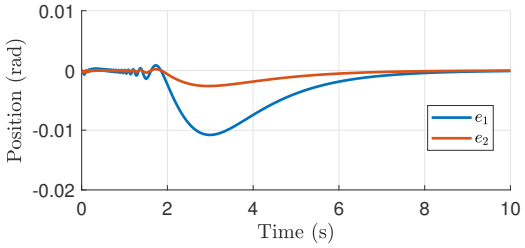
For the neocortex/striatum learning systems the weight matrix of the Lyapunov differential equation (32) is set to $Q = \text{diag}\{100, 100, 10, 10\}$ and the best gain of the update rule (28) is set to $\Gamma = 4000 I_2$. Fig. 3 shows the experience inference results.

The above results show that the real-world nonlinear system exhibit a similar performance to that shown by the reference model, that is, the hippocampus teaches the neocortex how to behave and achieve the control objective (see Fig. 3(a) and Fig. 3(b)). The matrices Γ, Q, and $\mathcal{P}(t)$ have an important role to infer the hippocampus experience because they give an adequate direction to update the gradient of (28). Fig. 3(c) and Fig. 3(d) show the convergence of the kernel matrix $\mathcal{P}(t)$ and the estimates of the control gain $\widehat{L}$ under the proposed Q matrix.

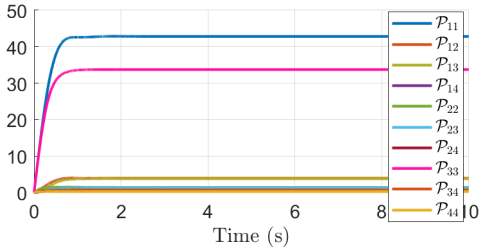
Notice that the real-world nonlinear system exhibits a near optimal performance similarly to the hippocampus reference model. However, the closed-loop system matrices $A_c^r(x_r) = A_1^r(x_r) - B_1^r(x_r)R^{-1}B_1^{r\top}(x_r)P(x_r)$ and $A_c^s(x) = A_1(x) - B_1(x)\widehat{L}$ are different. The numerical results for matrices



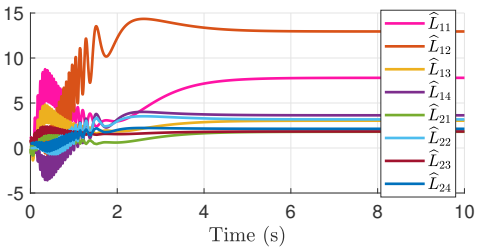
(a) States comparison x_r vs x



(b) Estimation error e

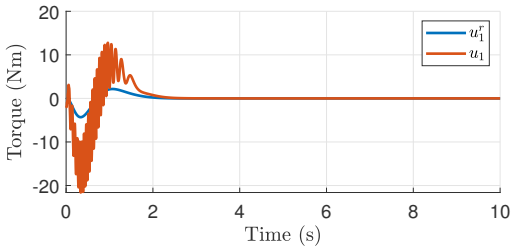


(c) Kernel matrix $\mathcal{P}(t)$ elements

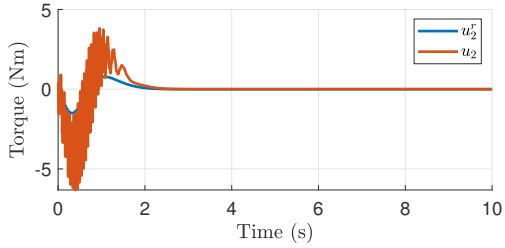


(d) Control gain $\widehat{L}$ estimates

Fig. 3. Experience inference results



(a) Control torque u_1^r vs. u_1



(b) Control torque u_2^r vs. u_2

Fig. 4. Control input policies comparison

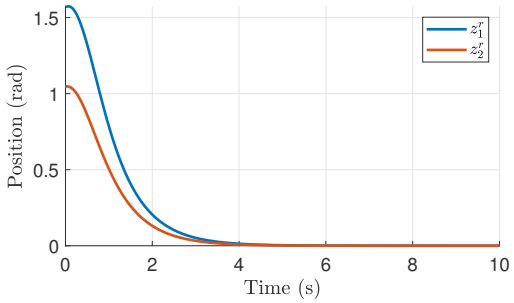
$A_c^r(x_r^d)$ and $A_c^s(x^d)$ in the desired stabilization point $x_r^d = x^d = [0,0,0,0]^\top$ are

$$A_c^r(x_r^d) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -35.35 & 66.37 & -13.64 & 21.22 \\ 66.37 & -194.8 & 21.22 & -64.6 \end{bmatrix},$$

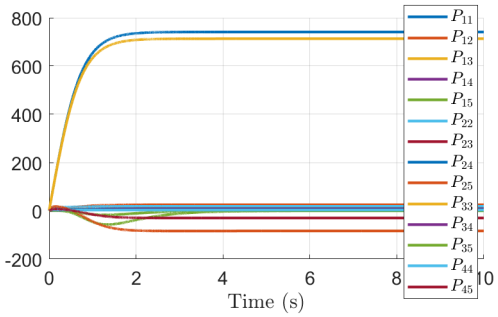
$$A_c^s(x^d) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -3.182 & 1.223 & -4.904 & 1.322 \\ 2.001 & -8.587 & 1.877 & -9.734 \end{bmatrix}.$$

Their respective eigenvalues are $\lambda(A_c^r)$={17.055, -2.9595,-6.163±3.343j} and $\lambda(A_c^s)$={2.3179,-0.681,-1.9141,-8.2339}.

Here we have an unstable eigenvalue because the robot is at a singularity point. Furthermore, notice that the second term in A_r^r is quadratic in the input dynamics $B_r(x_r)$. On the other hand, the second term in A_c^s is linear in the input dynamics $B(x)$. Matrices A_r^r and A_c^s are different because the neocortex/striatum control policy is not constrained as in the hippocampus case, then the control gain increases in a near optimal way such that it is obtained an almost equivalent hippocampus performance. This fact can be viewed in Fig. 4.

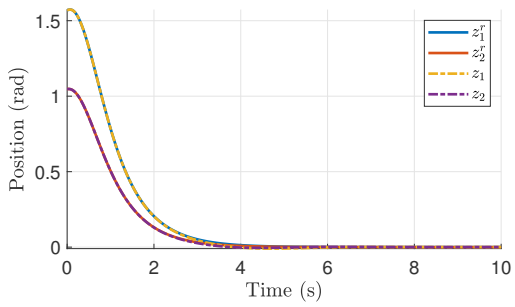


(a) States tracking z_r

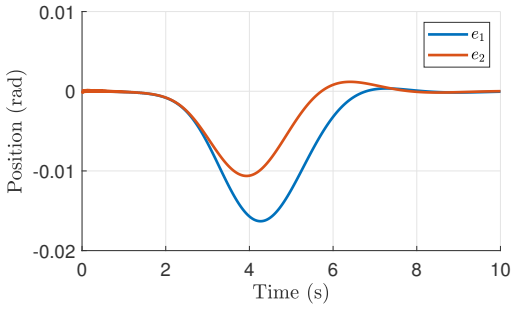


(b) Kernel matrix $P(z_r)$ elements

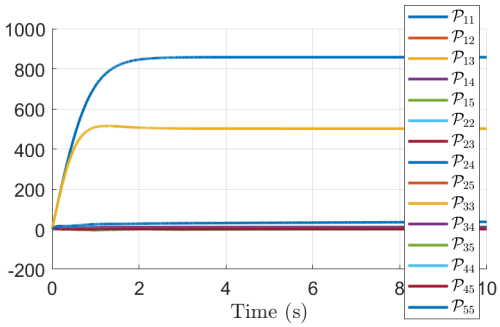
Fig. 5. Hippocampus-Neocortex results



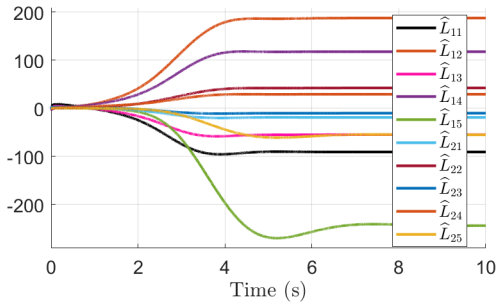
(a) States tracking z



(b) Estimation error e



(c) Kernel matrix $\mathcal{P}(z_r)$ elements



(d) Control gain $\widehat{L}$ estimates

Fig. 6. Experience inference results

B. With Gravity torques vector

To further illustrate the approach in a more general way consider the following vertical 2-DOF planar robot dynamics

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau,$$

where $G(q) \in \mathbb{R}^2$ stands to the gravitational torques vector. This robot dynamics can be rewritten as in (3) as

$$\dot{x} = \begin{bmatrix} x_3 \\ x_4 \\ -M^{-1}(q)\left[C(q,\dot{q})\dot{q} + G(q)\right] \end{bmatrix} + \begin{bmatrix} 0_2 \\ M^{-1}(q) \end{bmatrix} u.$$

The above non-linear dynamics written in its LTV with state disturbance (8) as

$$\begin{aligned}\dot{x} = &\left\{ \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0_2 & 0_2 \\ 0_2 & M^{-1}(q)C(q,\dot{q}) \end{bmatrix} \right\} x \\ &+ \begin{bmatrix} 0_{2\times 1} \\ -M^{-1}(q)G(q) \end{bmatrix} + \begin{bmatrix} 0_2 \\ M^{-1}(q) \end{bmatrix} u.\end{aligned}$$

By incorporating the new stable dynamics $\dot{y} = -\alpha y$ and defining the extended coordinates $z = [x^\top y]^\top \in \mathbb{R}^5$, then it is possible to write the LTV system with state disturbance as the following system

$$\begin{aligned}\dot{z} = &\left\{ - \begin{bmatrix} 0_2 & 0_2 & 0 \\ 0_2 & M^{-1}(q)C(q,\dot{q}) & \frac{M^{-1}(q)G(q)}{y} \\ 0_{1\times 2} & 0_{1\times 2} & 0 \end{bmatrix} \right. \\ &\left. + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha \end{bmatrix} \right\} z + \begin{bmatrix} 0_2 \\ M^{-1}(q) \\ 0_{1\times 2} \end{bmatrix} u \\ = &A(z)z + B(z)u.\end{aligned}$$

The hippocampus reference model is given by

$$M_r(q_r)\ddot{q}_r + C_r(q_r,\dot{q}_r)\dot{q}_r + G_r(q_r) = u_r,$$

where $G_r(q_r) \in \mathbb{R}^2$ denotes the gravitational torques vector of the reference model and is written as

$$\begin{aligned} G_r(q_r) = &\begin{bmatrix} G_1^r & G_2^r \end{bmatrix}^\top, \\ G_1^r = &m_1 g l_{c_1} \cos(q_1^r) \\ &+ m_2 g \left(l_1 \cos(q_1^r) + l_{c_2} \cos(q_1^r + q_2^r)\right), \\ G_2^r = &m_2 g l_{c_2} \cos(q_1^r + q_2^r). \end{aligned}$$

The hippocampus reference model written as a LTV system has the following structure

$$\dot{z}_r = \left\{ - \begin{bmatrix} 0_2 & 0_2 & 0 \\ 0_2 & M_r^{-1}(q_r)C_r(q_r,\dot{q}_r) & \frac{M_r^{-1}(q_r)G_r(q_r)}{y} \\ 0_{1\times 2} & 0_{1\times 2} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha \end{bmatrix} \right\} z_r + \begin{bmatrix} 0_2 \\ M_r^{-1}(q_r) \\ 0_{1\times 2} \end{bmatrix} u_r$$
$$= A_r(z_r)z_r + B_r(z_r)u_r,$$

where $z_r = [x_r^\top, y]^\top \in \mathbb{R}^5$. The same parameters of Table I are used for this simulation case. The decay rate of the new stable dynamics is set to $\alpha = 1 \times 10^{-4}$ such that it decays slowly to a fixed value $y_f = 0.05$ as it is stated in Remark 3. The same initial conditions are used with an additional value of $y(0) = 0.1$ for the new coordinate. The weights for the ADP algorithm are set to $S = \text{diag}\{1000, 1000, 500, 500, 1\}$ and $R = I_2$. The hippocampus results are shown in Fig. 5.

Notice that the last term of the kernel matrix $P(z_r)$, that is, P_{55} is not shown (see Fig. 5(b)) since the extended coordinates y is not controllable and hence P_{55} tends to increase infinitely. The term y_f is used to avoid this issue, however large learning time is required due to the small decay rate α. With the proposed weight matrices it is achieved a smooth and optimal performance of the hippocampus closed-loop trajectories.

For the experience inference algorithm, the best matrix for the Lyapunov differential equation is $Q = \text{diag}\{1000, 1000, 700, 200, 1\}$ and the gain matrix for the update rule (28) is set to $\Gamma = 3500 I_2$. Fig. 6 shows the results of the proposed experience inference algorithm.

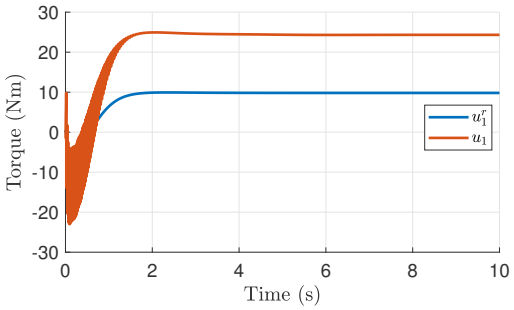
For this study, the gravitational torques of the real system are bigger than the reference model. This implies that the neocortex/striatum control policy to be bigger than the hippocampus control policy in order to compensate the gravitational terms. Furthermore, since the neocortex/striatum control policy is unconstrained then the control gain estimate $\widehat{L}$ can be as large as possible (is not an optimal gain) such that the estimation error decreases and converges to zero. Fig. 7 shows the comparison of the control policies.

C. Comparisons

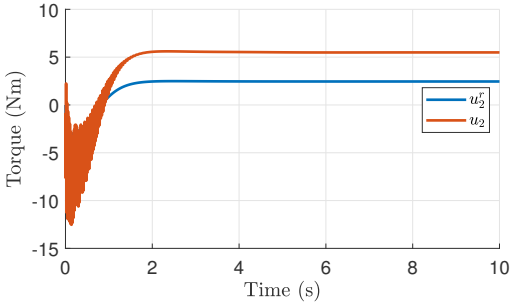
A synchronous reinforcement learning algorithm proposed in [54] is used for comparison purposes to show the biased control policies issue. Any RL/ADP algorithm can be used for this study which may exhibit similar results. The robot model without/with gravitational torques are considered in this section.

Whilst the proposed inference algorithm relates the hippocampus with the neocortex/striatum learning systems, this comparison can be regarded as independent learning systems where the hippocampus is the synchronous RL algorithm and the neocortex/striatum is given by the hippocampus control policy evaluated in the real nonlinear system's states.

The RL algorithm is trained with the reference model parameters to obtain the control policy that stabilizes the system. Subsequently, the control policy is tested in the real system using the real parameters of Table I. The results are given in Fig. 8.



(a) u_1^r vs. u_1



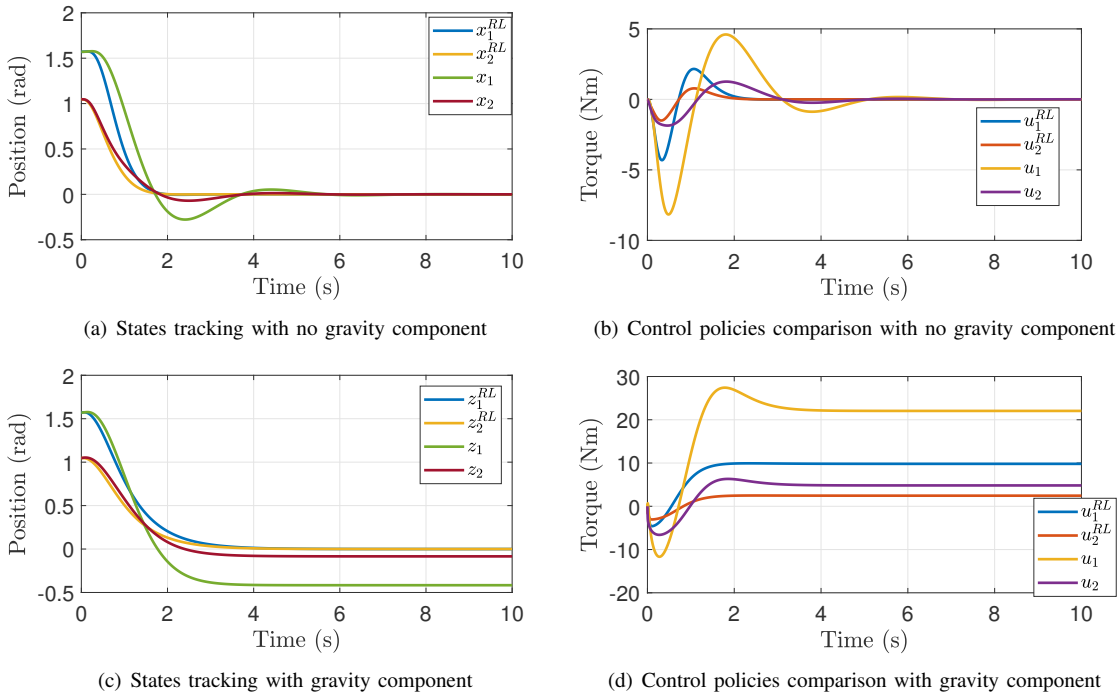
(b) u_2^r vs. u_2

Fig. 7. Control policies comparisons

The synchronous RL algorithm is able to stabilize the robot states trajectories without overshoots and smooth responses. However, when the final control policy is applied to the real system, the states trajectories do not behave as the trained model. Moreover, Fig. 8(c) exhibits steady state errors which means that the control policy obtained from the RL algorithm is not good enough to achieve the stabilization task effectively. The term of biased control policies takes importance in this scenario since the RL algorithm is trained under a certain model of the real system, then the output control policy is biased to that model and not necessarily exhibits a good performance in the real system.

In contrast, the proposed inference algorithm is able to infer the desired performance to the real system by adjusting the learned control policy via the striatum learning algorithm (see Fig. 3 and Fig. 6).

To further exhibit the migration problem, we change the parameters of the reference model twice the original real value of Table I, i.e., $m_1 = 4$, $m_2 = 2.8$, $l_1 = l_2 = 1.6$, $I_1 = 1$ and $I_2 = 0.2$ as it is proposed in [39]. Here the authors argue that if the RL is trained under a standard $\mathcal{H}_2$ controller, then the control policy will not be robust against disturbances or its optimal performance will be affected. This fact effectively demonstrates the migration problem under hidden disturbances and modelling error. To deal with the migration problem, the unrestricted optimization problem of the RL algorithm is modified to an optimization problem with constraints, where the constraints are given by a known upper bound of the



(a) States tracking with no gravity component

(b) Control policies comparison with no gravity component

(c) States tracking with gravity component

(d) Control policies comparison with gravity component

Fig. 8. Reinforcement Learning Inference Results

modelling error which is unknown in real applications.

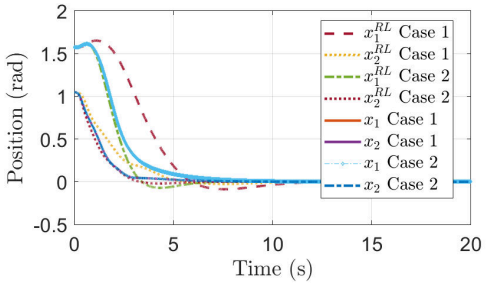
Two cases are considered in this study: Case 1 the standard synchronous RL algorithm, and Case 2 a worst-case synchronous RL algorithm with known minimal upper-bound $\bar{\omega}$, i.e., $0 < \omega \leq \bar{\omega} < \infty$. The trained control policy of the RL algorithms are migrated to the real system to show its robustness and safety control under different scenarios. In addition, we build a new reference model using the new parameters to test the proposed HBL algorithm. For the RL methods we use the standard notation of $f(x)$ and $g(x)$ instead of the SDC matrices of the LTV formulation.

The results are shown in Fig. 9. We can observe that the states trajectories of the robot (with/without gravity term) have smooth and different optimal performances (see Fig. 9(a) and Fig. 9(c)) due to the incorporation of the worst-case uncertainty constraint. Without gravity component, the migration problem is quite accurate for both cases with better transient performances because the real system's parameters are less than the parameters of the trained model. However, the migration problem appears in Case 1 in presence of the gravity term. In this scenario, the trained policy adds a small bias term due to a temporal difference error caused by the modelling error (see Fig. 9(c)). Case 2 overcomes this bias term by changing the unrestricted optimization problem to and optimization problem with constraints. Nevertheless, this approach is sensitive to the upper bound $\bar{\omega}$, that is, whilst large $\bar{\omega}$ gives large control policies which can destabilize the closed-loop trajectories, small $\bar{\omega}$ causes large bias terms and poor robustness properties. Furthermore, the upper bound $\bar{\omega}$ is unknown in a real migration problem. On the other hand, the proposed HBL overcomes this issue elegantly by adapting the RL control policy to the real-world system by means of the proposed experience inference algorithm (see Fig. 9(e) and Fig. 9(g)). The results show that the real-world system behaves as the reference model with high accuracy without any prior knowledge of the disturbance's upper bound.

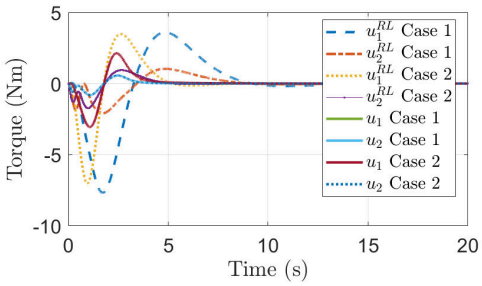
V. Conclusions

In this paper an experience inference human-behavior learning algorithm is proposed to solve the migration problem of optimal controllers applied to real-world nonlinear systems. The algorithm is inspired in how humans infers previous experiences to perform new similar tasks using the main complementary learning systems of the brain: the hippocampus, the neocortex, and the striatum. A reference model is used to emulate the hippocampus previous knowledge constructed by parameter estimates and a dynamic model structure. This model is written as a LTV system that can be handled by any ADP or RL algorithm. On the other hand, the experience inference algorithm is used in the real-world nonlinear system to infer the hippocampus desired performance by means of a neocortex/striatum control policy and a gradient update law. Simulations studies show that the proposed algorithm is capable to infer a desired performance to a real-world nonlinear system with near optimal results and fast learning phase.

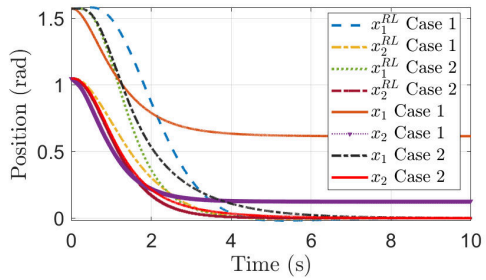
Further work considers to add constraints in the inference part to avoid large control gains. Furthermore, experience inference to completely different systems, in terms of structure and nonlinear dynamics, is the main concern for our future work. In addition, data-driven hippocampus model is focus of future research when a model structure is not available.



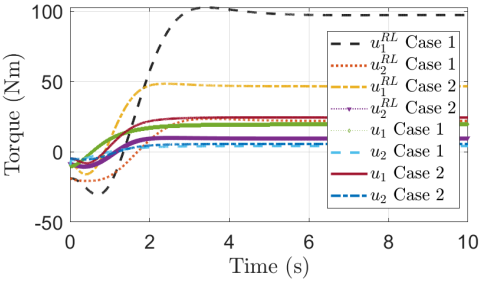
(a) States tracking of Worst-case RL with no gravity component



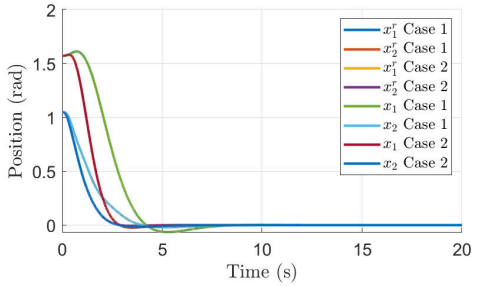
(b) Control policies of Worst-case RL with no gravity component



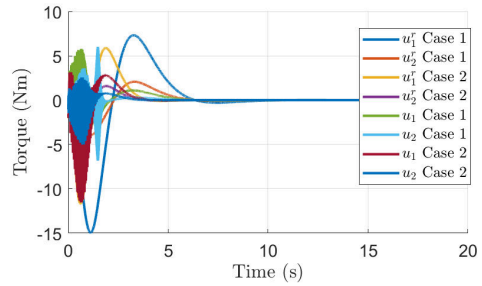
(c) States tracking of Worst-case RL with gravity component



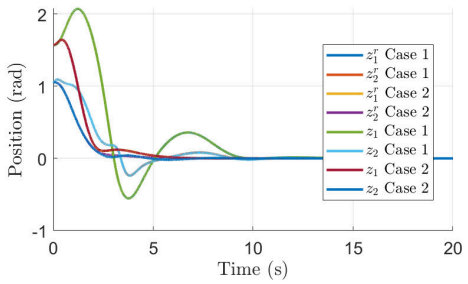
(d) Control policies of Worst-case RL with no gravity component



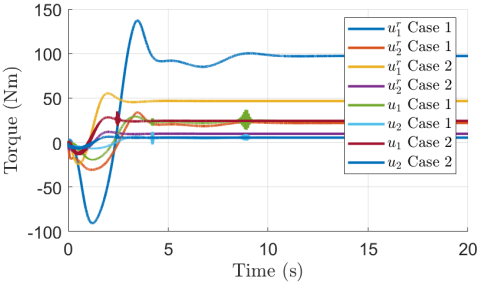
(e) States tracking of the HBL with no gravity component



(f) Control policies of the HBL with no gravity component



(g) States tracking of the HBL with gravity component



(h) Control policies of the HBL with gravity component

Fig. 9. Comparisons with different reference model under the worst-case RL philosophy

REFERENCES

[1] D. Liberzon, *Calculus of variations and optimal control theory*. Princeton university press, 2011.

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

[3] 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.

[4] 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.

[5] 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.

[6] Z.-P. Jiang, T. Bian, and W. Gao, "Learning-based control: A tutorial and some recent results," *Foundations and Trends® in Systems and Control*, vol. 8, no. 3, 2020.

[7] S. Tu and B. Recht, "The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint," in *Conference on Learning Theory*. PMLR, 2019, pp. 3036–3083.

[8] 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.

[9] 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.

[10] 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.

[11] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[12] 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.

[13] A. Perrusquía and W. Yu, "Robot position/force control in unknown environment using hybrid reinforcement learning," *Cybernetics and Systems*, vol. 51, no. 4, pp. 542–560, 2020.

[14] Q. Xie, B. Luo, and 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.

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

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

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

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

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

[20] 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.

[21] 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.

[22] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Transactions on System, Man, and Cybernetics Part B, Cybernetics*, vol. 38, no. 4, pp. 943–949, 2008.

[23] A. Gheibi, A. Ghiasi, S. Ghaemi, and M. Badamchizadeh, "Designing of robust adaptive passivity-based controller based on reinforcement learning for nonlinear port-Hamiltonian model with disturbance," *International Journal of Control*, vol. 93, no. 8, pp. 1754–1764, 2020.

[24] M. Tomás-Rodríguez and S. P. Banks, *Linear, time-varying approximations to nonlinear dynamical systems: with applications in control and optimization*. Springer Science & Business Media, 2010, vol. 400.

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

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

[27] 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.

[28] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive dynamic programming for control*. London, U.K.: Springer-Verlag, 2013.

[29] 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.

[30] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 584–598, 2016.

[31] ——, "Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1460–1470, 2016.

[32] B. Pang and Z.-P. Jiang, "Robust reinforcement learning: A case study in linear quadratic regulation," *arXiv preprint arXiv:2008.11592*, 2020.

[33] 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.

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

[35] 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.

[36] J. Young Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, 2015.

[37] 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.

[38] 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.

[39] 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.

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

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

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

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

[52] 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.

[53] 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.

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

[55] T. Cimen, "Survey of state-dependent riccati equation in nonlinear optimal feedback control synthesis," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1025–1047, 2012.
[56] N. Babaei and M. U. Salamci, "State dependent riccati equation based model reference adaptive control design for nonlinear systems," in *2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE, 2013, pp. 1–8.
[57] ——, "State dependent riccati equation based model reference adaptive stabilization of nonlinear systems with application to cancer treatment," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1296–1301, 2014.
[58] S. R. Nekoo, "Model reference adaptive state-dependent riccati equation control of nonlinear uncertain systems: Regulation and tracking of free-floating space manipulators," *Aerospace Science and Technology*, vol. 84, pp. 348–360, 2019.
[59] N. T. Nguyen, "Model-reference adaptive control," in *Model-Reference Adaptive Control*. Springer, 2018, pp. 83–123.
[60] J. R. Cloutier, D. T. Stansbery, and M. Sznaier, "On the recoverability of nonlinear state feedback laws by extended linearization control techniques," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, vol. 3. IEEE, 1999, pp. 1515–1519.
[61] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2550–2562, 2015.
[62] C.-Y. Lee and J.-J. Lee, "Adaptive control for uncertain nonlinear systems based on multiple neural networks," *IEEE Transactions on Systems Man and Cybernetics Part B*, vol. 34, no. 1, pp. 325–333, 2004.
[63] 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.
[64] 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.
[65] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.

Adolfo Perrusquía received the B.Eng. degree in Mechatronic Engineering from the National Polytechnic Institute (UPIITA-IPN) in 2014, and the M.S. and Ph.D. degrees, both in Automatic Control from the Automatic Control Department at the CINVESTAV-IPN in 2016 and 2020, respectively. He is currently a research fellow at the School of Aerospace, Transport and Manufacturing, Cranfield University, and a UK IC Postdoctoral Research Fellow. He is a member of the IEEE Computational Intelligence Society. His main research of interest focuses on robotics, mechanisms, machine learning, reinforcement learning, nonlinear control, system modeling and system identification.

Weisi Guo (S07, M11, SM17) received his MEng, MA, and Ph.D. degrees from the University of Cambridge, UK. He is Chair Professor of Human Machine Intelligence at Cranfield University. He has published over 180 papers and is PI on a number of molecular communication research grants. His research has won him several international awards. He was a Turing Fellow at the Alan Turing Institute.

2022-09-23

Optimal control of nonlinear systems using experience inference human-behavior learning

Perrusquía, Adolfo

IEEE

Perrusquia A, Guo W. (2023) Optimal control of nonlinear systems using experience inference human-behavior learning. IEEE CAA Journal of Automatica Sinica, Volume 10, Issue 1, January 2023, pp. 1-13

https://doi.org/10.1109/JAS.2022.000000