ISA Transactions 137 (2023) 646–655

Contents lists available at ScienceDirect

ISA Transactions

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

Practice article

Hippocampus experience inference for safety critical control of unknown multi-agent linear systems

Adolfo Perrusquía *, Weisi Guo

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

ARTICLE INFO

Article history:
Received 8 April 2022
Received in revised form 20 November 2022
Accepted 10 December 2022
Available online 16 December 2022

Keywords:
Hippocampus
Neocortex/striatum
Experience inference
Model reference
ADP
Actor algorithm
Unknown linear system
Multi-agent systems
Unbiased control policies

ABSTRACT

Risk mitigation is usually addressed in simulated environments for safety critical control. The migration of the final controller requires further adjustments due to the simulation assumptions and constraints. This paper presents the design of an experience inference algorithm for safety critical control of unknown multi-agent linear systems. The approach is inspired in the close relationship between three main areas of the brain cortex that enables transfer learning and decision making: the hippocampus, the neocortex, and the striatum. The hippocampus is modelled as a stable linear model that communicates to the striatum how the real-world system is expected to behave. The hippocampus model is controlled by an adaptive dynamic programming (ADP) algorithm to achieve an optimal desired performance. The neocortex and the striatum are designed simultaneously by an actor control policy algorithm that ensures experience inference to the real-world system. Experimental and simulations studies are carried out to verify the proposed approach.
© 2022 The Author(s). Published by Elsevier Ltd on behalf of ISA. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Control of multiple-input multiple output (MIMO) linear dynamical systems is a well-established area of research that designs control policies that force a real-world system to behave in accordance to a desired performance. These control policies are designed in terms of the physics model of the real-world system [1] that can be computed either offline (e.g., pole placement [2,3], linear quadratic regulator (LQR) [4,5], sliding mode control (SMC) [6]) or online (e.g., adaptive controllers [7]). All the aforementioned controllers are able to guarantee a desired-closed loop performance by adjusting their respective control gains.

Recent approaches use adaptive dynamic programming (ADP) [8,9] and reinforcement learning (RL) algorithms [10–12] to achieve an optimal closed-loop performance either offline or online. In this context, the MIMO linear system is known as a multi-agent system [13] where each agent corresponds to each degree-of-freedom (DOF) of the linear system. Moreover, in [14] it was stated that the main disadvantage of classical controllers (e.g., LQR, PID, SMC) is their adaptability and robustness against endogenous and exogenous disturbances. On the other hand, ADP/RL algorithms exhibit adaptability [15,16] and optimal/robust performances [17,18] under the fulfilment of a persistence of excitation (PE) condition [19–21], that is, a bad design of the PE signal can cause convergence to a local minima or divergence of the algorithm.

However, classical controllers are designed under certain assumptions, e.g., parameters, dynamic equations, and constraints that differ from the real-world system [22]. Analogously, ADP/RL algorithms need to fulfil a PE condition to guarantee parameters convergence. Hence, they are trained in simulation in order to prevent damage in the surrounding environment and to the system itself. The simulations are designed with several assumptions of the unknown linear dynamics in order to train the algorithm and to obtain the optimal control policy. This control policy will not behave as in the simulation case when it is migrated to the real-world system because it is biased to the simulation assumptions. This issue can be critical in terms of safety because, in the worst-case scenario, the trained control policy can destabilize the real-world system and cause damage to the surrounding infrastructure. We define a biased control policy as a control policy obtained from classical controllers or ADP/RL algorithms that guarantees a desired or optimal performance in the simulated system, but it cannot ensure the same performance in the real-world system due to robustness problems.

This issue is getting relevance in recent control and artificial intelligence (AI) applications, where realistic simulators have been developed to model the real-world interaction in a safe way. Nevertheless, some of these simulators require specific hardware and software requirements which cannot be easily modified by a

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

https://doi.org/10.1016/j.isatra.2022.12.011
0019-0578/© 2022 The Author(s). Published by Elsevier Ltd on behalf of ISA. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

non-expert user. Therefore, this paper deals with the safety critical control problem[1] that arises from the policy transference from simulation environment to the real-world multi-agent system.

The model used to obtain the control policy defines prior information about the real-world multi-agent system that can be used by an experience transference algorithm for policy adaptation and improvement. This procedure is similar to the humans' inference capabilities that allow them to perform different tasks that shared some similarities to their respective prior knowledge. Human-behaviour learning [23,24] is becoming a big challenge and trending topic in the research community [25]. Basically, it is inspired in how human learns and make decisions. Furthermore, it tries to understand the brain activity and which areas of the brain cortex are responsible for learning [26,27]. The brain cortex [28] is composed by: the neocortex, the striatum, and the neocortex. Each learning system is strongly interconnected such that is difficult to distinguish which system is responsible to develop a specific learning activity. In addition, each learning system exhibits complementary properties that derive complex learning behaviours [29].

The neocortex is a central part for reward based learning. This learning system is slow and associates pattern-dependent data to obtain well distributed, interconnected, and sometimes bidirectional representations for complex reward-driven behaviours. This is achieved by the encoding of temporal difference (TD) error by phasic midbrain dopaminergic neurons [29]. These neurons are updated in small rates such that they not generate good pattern association in the short term. However, in the long term the associations get stronger and gives a feasible way for decision making [30].

The hippocampus [31,32], which is the main subject of this paper, provides of fast learning architectures related to experience [33] and memory [34]. Humans execute a wide variety of activities based on their experience, that is, humans are capable to relate their previous experiences and transfer them to new activities in an elegant fashion. Some authors recognize the hippocampus as the teacher of the neocortex [28]. However, this system lacks of adaptability and generality since it is a completely pattern-independent system.

The striatum [35,36] connects the hippocampus and the neocortex in a complementary mechanism for better leaning generalization. Here two sources of knowledge of different nature (fast and slow learning) are combined to take their advantages and guarantee good generalization of learning. Notice that neither the neocortex or the hippocampus can make decisions without the striatum. Most of times, the striatum is modelled as a subsystem of both the hippocampus and the neocortex to avoid functionality confusion. However, it is important to distinguish the striatum as a complementary learning [28,30] system that learns how to relate the experiences and memories of the hippocampus with the distributed representations of the neocortex.

In this paper, a comprehensive human-behaviour learning application that guarantees safety critical control of multi-agent linear systems is provided. The approach exploits the complementary learning of the hippocampus, the neocortex, and the striatum for experience inference. The main contributions of the work are: (i) an actor control policy that guarantees safety and optimal performances in real-world multi-agent linear systems, (ii) the final control policy is unbiased to the simulations constraints, and (iii) a rigorous Lyapunov stability analysis is provided to support the proposed inference technique.

[1] Safety critical control refers to the control actions that a system must take to prevent risk and guarantee safety. In this paper, safety critical control is assessed in a migration problem of trained control policies in simulation into real-world environments to guarantee optimal and safe performances.

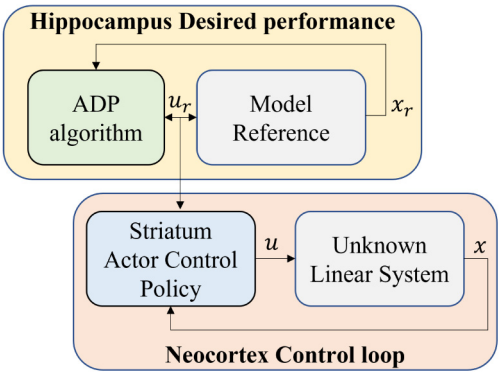


Fig. 1. Experience Inference Algorithm of Agent i.

2. Preliminaries

It is considered a multi-agent system [37,38] which can be decoupled into N linear systems of the form

$$\dot{x}^i = A^i x^i + B^i u^i + D^i(x^1, \ldots, x^N), \quad x^i(t_0) = x_0^i, \quad i = 1, \ldots, N \tag{1}$$

where $x^i \in \mathbb{R}^2$ is the state vector, $u^i \in \mathbb{R}$ is the control input, $A^i \in \mathbb{R}^{2\times 2}$, $B^i \in \mathbb{R}^2$ denote the dynamics of the linear system i and $D^i = D^i(x^1, \ldots, x^N) \in \mathbb{R}^2$ defines the interaction term between each agent which can be regarded as a bounded state dependent disturbance vector. It is assumed that matrices A^i, B^i and D^i have the form

$$A^i = \begin{bmatrix} 0 & 1 \\ a_1^i & a_2^i \end{bmatrix}, \quad B^i = \begin{bmatrix} 0 \\ b^i \end{bmatrix}, \quad D^i = \begin{bmatrix} 0 \\ d^i(x^1, \ldots, x^N) \end{bmatrix}, \tag{2}$$

where a_j^i, b^i, and d^i are unknown coefficients of the agent i. Assume that the sign of b^i is known in advance. The above formulation corresponds to many industrial robots, mechanical, electrical, and hydraulic systems which are modelled as decoupled linear systems with a disturbance to facilitate the control design. Noteworthy that in this study the interaction term D^i plays an important role for the control design in contrast to classical multi-agent systems with independent agents where the interaction term D^i is neglected.

In the sequel of the paper, the superscript i is omitted for sake of simplicity. The main control goal is to design a human-behaviour learning algorithm that is able to track a desired constant reference $x^d \in \mathbb{R}^2$ associated to each agent i without compromising the stability and safety of both the environment and the multi-agent linear system in real applications.

Fig. 1 shows a simplified block scheme of the proposed human-behaviour learning algorithm for each agent i of the multi-agent linear system. The hippocampus is modelled as a stable reference model that permits to infer useful pattern independent experience. Conditional statements (if a then b) are good examples in how humans can carry out different things from inference of their experiences. In contrast with classical control and ADP/RL architectures proposed in the literature [8,17,39], this approach aims to control the reference model instead of the real-world multi-agent linear system in order to provide a notion of how we expect the real-world system should behave.

For this application is difficult to separate the striatum and the neocortex learning systems because they are executed simultaneously. On the one hand, the striatum is modelled as an adaptive actor algorithm that relates both the hippocampus control policy and an inner feedback controller of the neocortex to guarantee experience inference. On the other hand, the neocortex learns to stabilize the unknown linear system by using the output of the striatum learning system.

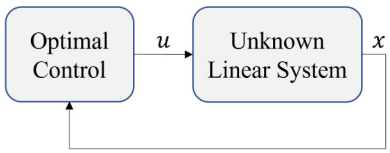


Fig. 2. Classical optimal control scheme.

Remark 1. The actor algorithm belongs to the striatum learning system because it relates two different sources of knowledge: the hippocampus control policy and an inner feedback controller.

The design of each learning system is discussed in the following sections.

3. Hippocampus learning system

To further motivate the proposed approach consider the control scheme of Fig. 2. This figure shows the basic control diagram that governs a wide variety of control architectures [40] in the literature.

To design an optimal control law we need prior knowledge of the linear system parameters [41,42], otherwise ADP/RL algorithms can be used to obtain the optimal or near optimal control policy iteratively [43,44]. However, two issues can be distinguished: (1) ADP/RL algorithms need to fulfil a PE condition to ensure parameters convergence and (2) biased control policies are obtained due to predefined assumptions and constraints.

The first issue is inherent to any identification [45] and adaptive algorithm which can be reduced by adding hippocampus memory methods such as experience replay [46–48] and eligibility traces [49,50] or prior knowledge of the system dynamics. The second issue is a hard problem since ADP/RL algorithms need to be trained in simulation in order to fulfil the PE condition without damaging the environment and the system itself [39]. So, the unknown system has to be modelled under certain assumptions such that the final control policy obtained from the simulation guarantees a desired closed-loop performance. However, this control policy is biased due to the assumptions and will not exhibit the same performance in the real-world system. This paper addresses the second problem by using the experience inference algorithm of Fig. 1.

The hippocampus learning system of Fig. 1 is constructed as i reference models associated to each agent that have the following structure

$$\dot{x}_r = A_r x_r + B_r u_r + D_r \tag{3}$$

where $A_r \in \mathbb{R}^{2\times 2}$, $B_r \in \mathbb{R}^2$, and $D_r \in \mathbb{R}^2$ are known constant matrices constructed from either prior knowledge, estimated values or a desired and stable model; $x_r \in \mathbb{R}^2$ and $u_r \in \mathbb{R}$ are the state and control input of the reference model, respectively. The matrices A_r, B_r and D_r of each agent are modelled as in (2) where the sign of b_r matches with the sign of element b. Define the hippocampus tracking error $\widetilde{x}_r \in \mathbb{R}^2$ as

$$\widetilde{x}_r = x_r - x^d. \tag{4}$$

The hippocampus error dynamics between (3) and the desired reference x^d is

$$\dot{\widetilde{x}}_r = A_r \widetilde{x}_r + B_r u_r + D_r + A_r x^d. \tag{5}$$

Define the hippocampus control policy as [51]

$$\begin{aligned} u_r &= u_1 + u_2, \\ &= u_1 - B_r^{\dagger}(A_r x^d + D_r), \end{aligned} \tag{6}$$

where $B_r^{\dagger}$ is the Moore–Penrose pseudoinverse of matrix B_r. Here u_2 corresponds to a feedforward term and u_1 is the optimal feedback term. Substituting (6) in (5) gives

$$\dot{\widetilde{x}}_r = A_r \widetilde{x}_r + B_r u_1. \tag{7}$$

The ADP algorithm is designed to minimize the infinite horizon value function [52]

$$V(x_r) = \int_t^{\infty} (\widetilde{x}_r^{\top} S \widetilde{x}_r + R u_1^2) d\tau \tag{8}$$

where $S = S^{\top} \geq 0 \in \mathbb{R}^{2\times 2}$ and $R > 0$ are predefined weight matrices for each agent i. The Hamiltonian of system (7) is

$$\begin{aligned} H(\widetilde{x}_r, u_1, \nabla V) = \; & \nabla^{\top} V(A_r \widetilde{x}_r + B_r u_1) + \widetilde{x}_r^{\top} S \widetilde{x}_r \\ & + R u_1^2 = 0. \end{aligned} \tag{9}$$

The optimal value function [53] can be expressed as

$$V^*(\widetilde{x}_r) = \widetilde{x}_r^{\top} P \widetilde{x}_r \tag{10}$$

where $P = P^{\top} > 0 \in \mathbb{R}^{2\times 2}$ is the solution to the following algebraic Riccati equation (ARE) [54,55]

$$A_r^{\top} P + P A_r - \frac{1}{R} P B_r B_r^{\top} P + S = 0. \tag{11}$$

The optimal hippocampus control policy u_1^* of agent i is computed by

$$\begin{aligned} u_1^* &= -K \widetilde{x}_r \\ &= -\frac{1}{2R} B_r^{\top} \nabla V^* = -\frac{1}{R} B_r^{\top} P \widetilde{x}_r, \end{aligned} \tag{12}$$

where $K \in \mathbb{R}^{1\times 2}$ is the optimal stabilizing gain of the hippocampus learning system of agent i. The ARE provides an off-line solution of the optimal control problem. An on-line version can be easily obtained by transforming the ARE to a matrix differential Riccati equation (MDRE) as

$$\dot{P}(t) = -A_r^{\top} P(t) - P(t) A_r + \frac{1}{R} P(t) B_r B_r^{\top} P(t) - S. \tag{13}$$

In contrast to the classical ARE, the MDRE is able to compute on-line the solution of the Riccati equation by solving a first-order differential equation. The following theorem shows the convergence and equivalence between the MDRE (13) and the ARE (11).

Theorem 1. *Consider the linear time invariant (LTI) system (3). The kernel matrix $P(t)$ of the MDRE (13) is input-state stable (ISS) and converges exponentially to the kernel matrix P of the ARE (11).*

Proof. Taking the difference between the MDRE (13) and the ARE (11) gives

$$\begin{aligned} \dot{P}(t) = & -A_r^{\top}(P(t) - P) - (P(t) - P) A_r + \frac{1}{R} P(t) B_r B_r^{\top} P(t) \\ & - \frac{1}{R} P B_r B_r^{\top} P \end{aligned} \tag{14}$$

$$\dot{\mathcal{P}}(t) = -\mathcal{A}^{\top} \mathcal{P}(t) - \mathcal{P}(t) \mathcal{A} + \frac{1}{R} \mathcal{P}(t) B_r B_r^{\top} \mathcal{P}(t), \tag{15}$$

where $\mathcal{A} = A_r - \frac{1}{R} B_r B_r^{\top} P$ and $\mathcal{P}(t) = P(t) - P$. Let $X^{-1}(t) = \mathcal{P}(t)$, then the following holds

$$\frac{d}{dt}\left(X(t) X^{-1}(t) = I_n\right),$$

$$\dot{X}(t) X^{-1}(t) + X(t) \frac{dX^{-1}(t)}{dt} = 0_n,$$

$$\dot{X}(t) = -X(t) \frac{dX^{-1}(t)}{dt} X(t) = -\mathcal{P}^{-1}(t) \dot{\mathcal{P}}(t) \mathcal{P}^{-1}(t).$$

In consequence,

$$\dot{X}(t) = X(t)\mathcal{A}^\top + \mathcal{A}X(t) - B_r R^{-1} B_r^\top \tag{16}$$

Notice that $\mathcal{A} = A_r - B_r K$ is a Hurwitz matrix, that is, the $\mathfrak{Re}[\lambda(\mathcal{A})] < 0$. The above differential equation can be written as

$$\dot{\mathcal{X}}(t) = \Lambda \mathcal{X}(t) - F, \tag{17}$$

where $\mathcal{X}(t) = \text{vec}(X(t))$, $\Lambda = [\mathcal{A} \otimes I_n + I_n \otimes \mathcal{A}]$, and $F = \text{vec}\left(\frac{1}{R} B_r B_r^\top\right)$. The solution of the above ODE is

$$\mathcal{X}(t) = \exp^{\Lambda t}\, \mathcal{X}(0) + \Lambda^{-1}(I_n - \exp^{\Lambda t})F. \tag{18}$$

When $t \to \infty$ the solution of the ODE (18) reduces to

$$\mathcal{X}(\infty) = \Lambda^{-1} F.$$

The above equality is equivalent to the following Lyapunov equation

$$\mathcal{P}^{-1}(\infty)\mathcal{A}^\top + \mathcal{A}\mathcal{P}^{-1}(\infty) - \frac{1}{R} B_r B_r^\top = 0. \tag{19}$$

The Lyapunov Eq. (19) is pre-multiplied and post-multiplied by $\mathcal{P}(\infty)$ such that the following Riccati equation is obtained

$$\mathcal{A}^\top \mathcal{P}(\infty) + \mathcal{P}(\infty)\mathcal{A} - \frac{1}{R}\mathcal{P}(\infty) B_r B_r^\top \mathcal{P}(\infty) = 0. \tag{20}$$

Moreover, when $t \to \infty$ the MDRE (15) is reduced to

$$-\dot{\mathcal{P}}(\infty) = \mathcal{A}^\top \mathcal{P}(\infty) + \mathcal{P}(\infty)\mathcal{A} - \frac{1}{R}\mathcal{P}(\infty) B_r B_r^\top \mathcal{P}(\infty). \tag{21}$$

From (20) and (21) it follows that $\dot{\mathcal{P}}(\infty) = \dot{P}(\infty) = 0$ which implies that $P(\infty)$ has a constant value. Since $\mathcal{P}(\infty) = P(\infty) - P$, then (21) can only be equal to zero if the kernel matrices of the MDRE and the ARE are equivalent, that is, $P(\infty) = P$. This completed the proof. □

The above result fills the gap of current adaptive programming algorithms that shows the equivalence between different formulations of the Riccati equation [4].

Therefore, the output of the hippocampus learning system is the optimal control policy and performance that feeds the neocortex/striatum for experience inference in the real multi-agent system.

4. Neocortex and striatum learning systems

These systems are designed simultaneously to control the unknown multi-agent linear system and force it to exhibit the same hippocampus performance. Define the neocortex tracking error $\widetilde{x} \in \mathbb{R}^2$ as

$$\widetilde{x} = x - x^d. \tag{22}$$

The tracking error dynamics between (1) and the desired reference x_d is

$$\dot{\widetilde{x}} = A\widetilde{x} + Bu + D + Ax^d. \tag{23}$$

Consider an ideal case where the matrices A, B, and D are known. The neocortex and striatum learning systems aim to design a control policy which is able to capture the information of the hippocampus policy to infer it to the real-world multi-agent linear system effectively. For this purpose, consider that the optimal control policy[2] of the striatum learning system is given by

$$u^* = K_1^\top \widetilde{x} + k_2 u_r, \tag{24}$$

where $K_1 \in \mathbb{R}^2$ and $k_2 \in \mathbb{R}$ are gain matrices. Notice that the striatum control policy connects the hippocampus control policy u_r and the inner neocortex control policy $K_1\widetilde{x}$. The unknown linear system (23) under the striatum control policy (24) is

$$\begin{aligned}\dot{\widetilde{x}} &= A\widetilde{x} + B(K_1^\top \widetilde{x} + k_2 u_r) + D + Ax^d \\ &= (A + BK_1^\top)\widetilde{x} + k_2 B u_r + D + Ax^d.\end{aligned} \tag{25}$$

The dynamics (25) is equivalent to the hippocampus reference model (5) if $A + BK_1^\top = A_r$ and $k_2 B = B_r$. Here it is more clear why it is difficult to separate the striatum and neocortex policies because the gains K_1 and k_2 relate the hippocampus and neocortex control policies but also K_1 correspond to the gain of the neocortex control policy.

Assumption 1. There exists a gain $k_3 > 0$ such that $B = k_3 B_r$ and therefore $k_2 = 1/k_3$.

The previous assumption is valid for the kind of linear systems focused on this paper. So, (25) is rewritten as

$$\dot{\widetilde{x}} = A_r \widetilde{x} + B_r u_r + D + Ax^d. \tag{26}$$

Therefore, (26) will exhibit a near optimal performance if u_r is designed as in (6). However, it is not possible to design K_1 and k_2 because matrices A and B are unknown and hence, an approximation of the control policy is required. Consider the following actor control policy

$$u = \widehat{K}_1^\top \widetilde{x} + \widehat{k}_2 u_r, \tag{27}$$

where $\widehat{K}_1 \in \mathbb{R}^2$ and $\widehat{k}_2 \in \mathbb{R}$ are estimates of K_1 and k_2, respectively. Define the reference error as

$$e = x - x_r. \tag{28}$$

The closed-loop error dynamics between (1) and (3) is

$$\begin{aligned}\dot{e} =& Ax + B(\widehat{K}_1^\top \widetilde{x} + \widehat{k}_2 u_r) + D - A_r x_r - B_r u_r - D_r, \\ =& A_r e + k_3 B_r(\widetilde{K}_1^\top \widetilde{x} + \widetilde{k}_2 u_r) + D - D_r - k_3 B_r K_1^\top x^d \\ =& A_r e + k_3 B_r(\widetilde{K}_1^\top \widetilde{x} + \widetilde{k}_2 u_r) + \eta\end{aligned} \tag{29}$$

where $\widetilde{K}_1 = \widehat{K}_1 - K_1 \in \mathbb{R}^2$ and $\widetilde{k}_2 = \widehat{k}_2 - k_2 \in \mathbb{R}$ denote the gain error matrices, $\eta = D - D_r - k_3 B_r K_1^\top x^d$ is an unknown disturbance vector.

The following theorem establishes the ultimate uniformly boundedness (UUB) [19] of the closed-loop error trajectories of the proposed experience inference algorithm.

Theorem 2. *Consider the unknown linear system (1) in closed-loop with the neocortex/striatum actor control policy (27), the hippocampus control policy (12), and the hippocampus reference model (3). Consider that there exists a matrix $Q = Q^\top > 0 \in \mathbb{R}^{2\times 2}$ that satisfies*

$$\lambda_{\min}(Q) > 2\lambda_{\max}(M)\|\eta\| + \rho, \tag{30}$$

where $M = M^\top > 0 \in \mathbb{R}^{2\times 2}$ is the solution of the following Lyapunov equation [56]

$$A_r^\top M + MA_r = -Q, \tag{31}$$

and $\rho > 0$. If the estimates $\widehat{K}_1$ and $\widehat{k}_2$ are updated as

$$\dot{\widehat{K}}_1 = \dot{\widehat{K}}_1 = -\Gamma \widetilde{x} e^\top M B_r, \tag{32}$$

$$\dot{\widehat{k}}_2 = \dot{\widehat{k}}_2 = -\gamma u_r e^\top M B_r, \tag{33}$$

where $\Gamma = \Gamma^\top > 0 \in \mathbb{R}^{2\times 2}$ and $\gamma > 0$ are the gains of the update rules; then the trajectories of (29) are uniformly ultimately bounded (UUB) with a practical bound $\mu = \frac{2\lambda_{\max}(M)}{\lambda_{\min}(Q)}\|\eta\|$ and the estimates $\widehat{K}_1$ and $\widehat{k}_2$ remain bounded. Therefore, the trajectories of (1) behave approximately as the states x_r of the hippocampus reference model (3).

[2] The optimal control policy of the striatum is the policy that ensures experience inference from the hippocampus' optimal performance to the real-system in a least-squares sense.

Proof. Consider the following Lyapunov function

$$W = e^\top Me + k_3 \text{tr}\{\widetilde{K}_1^\top \Gamma^{-1} \widetilde{K}_1\} + \frac{k_3}{\gamma}\widetilde{k}_2^2. \tag{34}$$

The time-derivative of (34) along the closed-loop error dynamics trajectories (29) and the update rules (32) and (33) is

$$\begin{aligned}\dot{W} =& e^\top (MA_r + A_r^\top M)e + 2k_3 e^\top MB_r(\widetilde{K}_1^\top \widetilde{x} + \widetilde{k}_2 u_r) + 2e^\top M\eta \\ &+ 2k_3 \text{tr}\{\widetilde{K}_1^\top \Gamma^{-1}\dot{\widetilde{K}}_1\} + 2\frac{k_3}{\gamma}\widetilde{k}_2\dot{\widetilde{k}}_2 \\ =& -e^\top Qe + 2e^\top M\eta \\ \leq& -\lambda_{\min}(Q)\|e\|^2 + 2\lambda_{\max}(M)\|e\|\|\eta\|,\end{aligned} \tag{35}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalues, respectively. Therefore, $\dot{W} \leq 0$ as long as

$$\|e\| > \frac{2\lambda_{\max}(M)}{\lambda_{\min}(Q)}\|\eta\| \equiv \mu. \tag{36}$$

Selecting the matrix Q such that (30) is satisfied ensures that the trajectories of the system (29) converge to a compact set Ω of radius μ, that is, $\|e\| \leq \mu$ and hence the trajectories of (29) are UUB.

Boundedness of e implies boundedness of x and x_r; hence the tracking error $\widetilde{x}$ and the control policy u are also bounded. In consequence $\|\dot{e}\|$ is bounded. Since e, u are bounded, then it is ensured boundedness of the gains error $\widetilde{K}_1$, $\widetilde{k}_2$ and hence $\widehat{K}_1$ and $\widehat{k}_2$. Finally, the practical bound μ implies that the error can be small enough such that the trajectories of the unknown linear system (1) almost converges to the hippocampus reference trajectories (3). This completes the proof. □

Remark 2. The practical bound μ depends of three elements: the Q matrix proposed by the user, the solution matrix M of the Lyapunov equation, and the disturbance vector η. The bound μ can be small enough by choosing a large enough Q matrix. However, the solution matrix M of the Lyapunov equation can also be increased depending in how the reference matrix A_r is designed. In other words, if A_r is small and Q is large then M will be large. Conversely, if A_r is large and Q is small then M will be small. Therefore, large M can excite the disturbance vector η which may lead to a large practical bound μ and policies with undamped oscillations. Moreover, small Q matrix can amplify the effect of η and consequently μ will be large. The effect of the disturbance vector η can be attenuated if the vector D_r of the reference model is close to the real values of the real-world interaction vector D, i.e. $D_r \approx D$.

5. Experimental studies

In this section, the experience inference algorithm is tested in two different problems: a DC servomechanism and a 4-DOF robot.

5.1. DC-servomechanism

First we test the approach using a single agent linear system given by the DC-servomechanism of Fig. 3. Details of the DC servomechanism can be found in [57]. The tachometer and encoder outputs are filtered by the following low-pass filter $G_1(s) = \frac{50}{s+50}$.

Data acquisition is performed by a Sensoray model 626 PCI data acquisition target endowed with inputs for optical encoders. The data card electronics increases four times the optical encoder resolution up to 10,000 pulses per revolution. The real parameters of the DC-servomechanism are obtained previously and are given by

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -0.1026 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 5.3996 \end{bmatrix}.$$



Fig. 3. DC-servomechanism prototype.

The desired position x_1^d is a square wave form function with amplitude 1 and frequency of 0.2 Hz. The low-pass filter $G_2(s) = \frac{10}{s+10}$ is used to smooth the non-differentiable parts of the wave function. The reference model of the hippocampus is proposed as the next stable linear system

$$A_r = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.2 \end{bmatrix}, \quad B_r = \begin{bmatrix} 0 \\ 5 \end{bmatrix}.$$

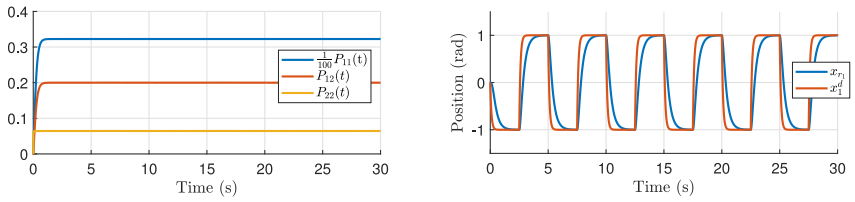
The reference model can be freely chosen and only require to have the same structure as in (1). The weight matrices of the MDRE/ARE are proposed as $S = \text{diag}\{100, 10\}$ and $R = 0.01$. The kernel matrix of the ARE (11) and the optimal control gain are

$$P = \begin{bmatrix} 32.2489 & 0.2 \\ 0.2 & 0.0644 \end{bmatrix},$$
$$K = \begin{bmatrix} 99.98 & 32.2089 \end{bmatrix}.$$

The hippocampus can be trained on-line or off-line since it is a completely known system. Fig. 4 shows the results of the hippocampus learning system. The elements of the kernel matrix $P(t)$ converge to the same values of the kernel matrix P. The trajectories of x_r exhibit the desired optimal performance that we want to inject to the DC servomechanism dynamics via the neocortex/striatum learning systems.

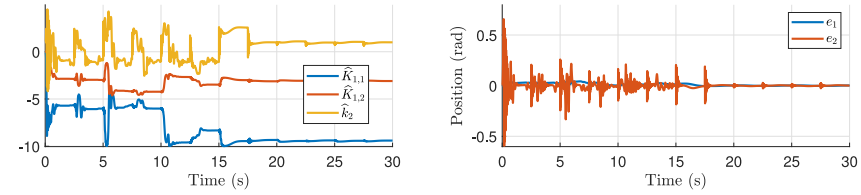
For the neocortex/striatum learning systems it is proposed a matrix $Q = \begin{bmatrix} 8 & 2 \\ 2 & 8 \end{bmatrix}$ to compute the kernel matrix M of the Lyapunov Eq. (31). The resulting matrix is $M = \begin{bmatrix} 248 & -4 \\ -4 & 22 \end{bmatrix}$. The optimal gains which guarantee an equivalent hippocampus reference model tracking are $K_1 = \begin{bmatrix} -0.0185 & -0.018 \end{bmatrix}$ and $k_2 = 0.926$. The gains of the update rules (32) and (33) are tuned manually until the best performance is achieved. The final gains are set to $\Gamma = \text{diag}\{3, 0.5\}$ and $\gamma = 3$.

The results of the experience inference algorithm are exhibited in Fig. 5. The gain estimates in Fig. 5(a) show stable and bounded values that correspond to one of the possible solutions of the inference problem. The gains converge to the following values: $\widehat{K}_1 = \begin{bmatrix} -9.398 & -3.1 \end{bmatrix}$ and $\widehat{k}_2 = 0.9961$. Notice that these values differ from the optimal gains because the main goal of the experience inference algorithm is to achieve the same or almost same performance of the hippocampus by minimizing the reference error e. Hence, there exists multiple gain combinations that achieve a similar hippocampus performance and that reduce the reference error e (see Fig. 5(b)). Figs. 5(c) and 5(d) demonstrate

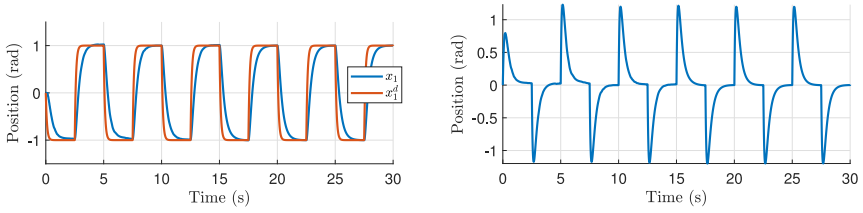


(a) Kernel matrix $P(t)$ components (b) States x_r trajectories

Fig. 4. Hippocampus results.

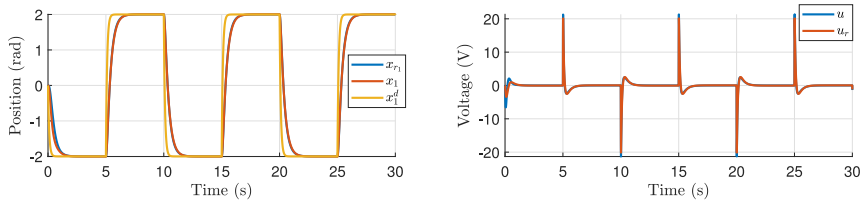


(a) Gains convergence (b) Reference error e trajectories



(c) States x trajectories (d) Tracking error $\widetilde{x}$

Fig. 5. Neocortex/Striatum results.



(a) State trajectories comparisons (b) Control Voltage comparisons

Fig. 6. Experience inference validation.

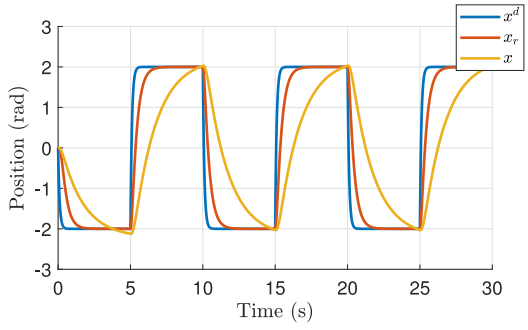


Fig. 7. Migration problem of biased control policies.

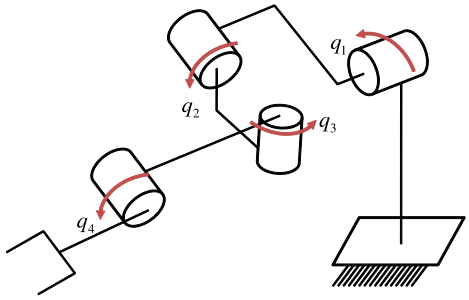
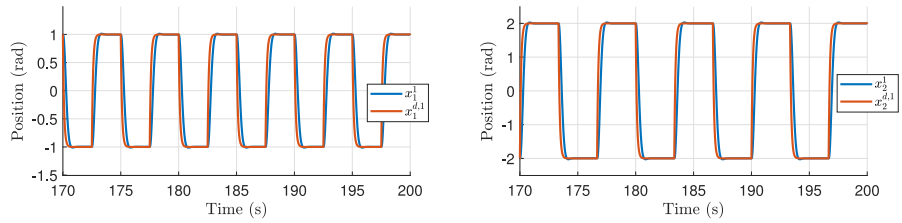
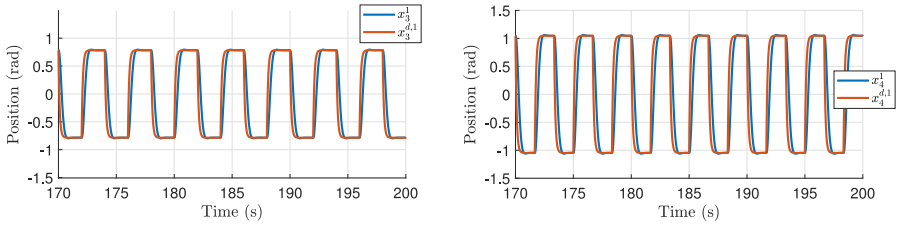


Fig. 8. 4-DOF robot diagram.

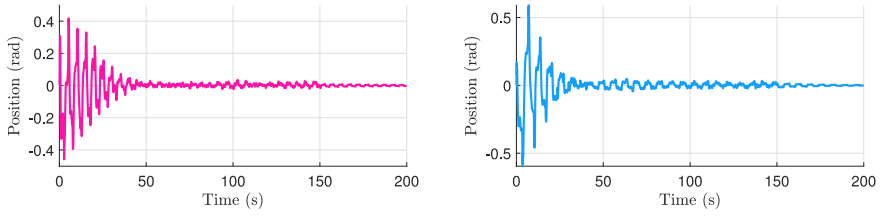


(a) Joint position x_1 (b) Joint position x_2

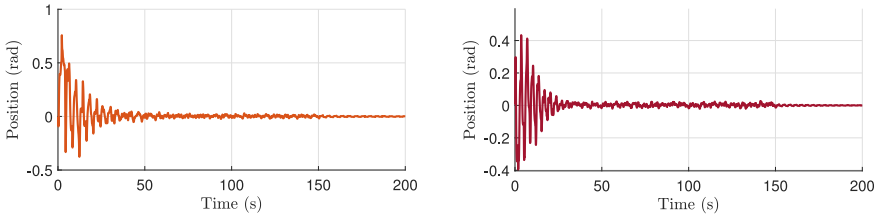


(c) Joint position x_3 (d) Joint position x_4

Fig. 9. Neocortex/Striatum results.



(a) Reference error e_1 (b) Reference error e_2



(c) Reference error e_3 (d) Reference error e_4

Fig. 10. Reference error results.

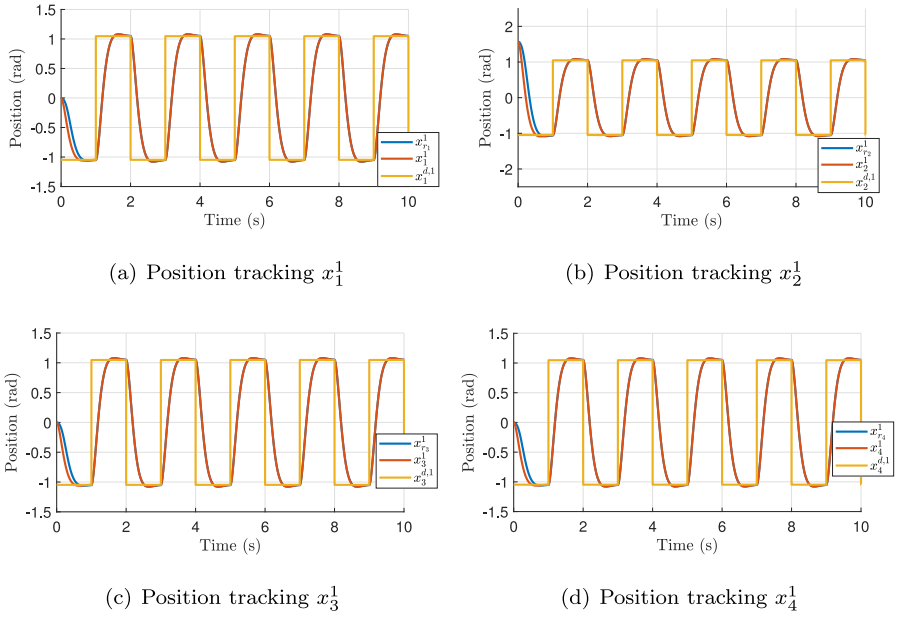
that the hippocampus' experience is inferred to real-world DC servomechanism trajectories x.

To further verify the proposed approach, we fix the gains of the actor control policy and modify the desired reference x^d to a square wave form of amplitude 2 and frequency 0.1 Hz. The results are shown in Fig. 6.

Although different gains $\widehat{K}_1$ and $\widehat{K}_2$ are obtained, the control voltage applied to the DC servomechanism presents almost the same performance (see Fig. 6(a)). The norm of each control input, $\bar{u}_r = \left\| u_r(t)\big|_{t=0}^{t=30} \right\|$ and $\bar{u} = \left\| u(t)\big|_{t=0}^{t=30} \right\|$, are computed as a performance metric to evaluate the hippocampus and the neocortex/striatum control policies. The results are: $\bar{u}_r = 168.981$ and $\bar{u} = 171.8122$ which permit concluding that the neocortex/striatum learning systems have almost the same hippocampus performance and hence a near optimal performance with respect to the proposed weight matrices S and R.

The control policy of the hippocampus is migrated directly to the real-world system to demonstrate the biased control policies problem. A Q-learning algorithm [44] is used to derive the optimal hippocampus control policy using $S = I_2$ and $R = 1$ as weight matrices and the same reference model. The results are shown in Fig. 7.

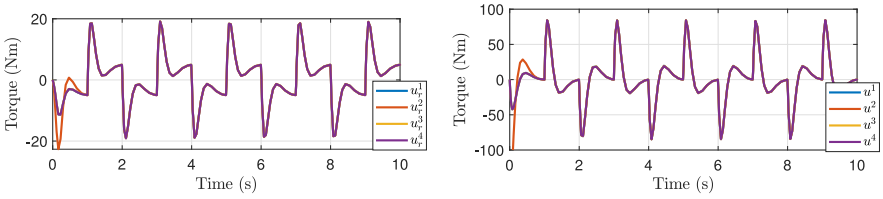
We can see that the states x_r of the hippocampus reference model track the desired reference x^d with high accuracy by using the solution of a least-squares algorithm in the Q-learning update rule. However, when the optimal control policy is migrated to the real-world system we can observe that the states x cannot follow the desired reference. The hippocampus control policy is biased to the reference model assumptions and hence cannot guarantee an accurate optimal performance in the real-world system. This issue is solved by the proposed experience inference algorithm as it is shown in Figs. 5 and 6.



(a) Position tracking x_1^1 (b) Position tracking x_2^1

(c) Position tracking x_3^1 (d) Position tracking x_4^1

Fig. 11. Experience inference validation.



(a) Hippocampus control policies (b) Neocortex/Striatum control policies

Fig. 12. Control policies comparison.

5.2. 4-DOF robot

The previous example shows good results for a single agent case. In this example, we further verify the approach for a multi-agent case. Consider the 4-DOF robot (see Fig. 8) used in [58].

Assume that the robot has a gearbox in each actuator such as the interaction dynamics between the robot and the actuator is reduced to a linear system with disturbances [59], that is,

$$H\ddot{q} + C\dot{q} + F = u, \tag{37}$$

where $H \in \mathbb{R}^{4\times4}$ defines a diagonal inertia matrix, $C \in \mathbb{R}^{4\times4}$ stands to a diagonal viscous friction matrix, $F \in \mathbb{R}^4$ models a disturbance vector, $u \in \mathbb{R}^4$ defines the input vector, and $q, \dot{q}, \ddot{q} \in \mathbb{R}^4$ define the joint positions, velocities, and acceleration vectors. We can decouple (37) into 4 linear systems with disturbances as

$$\dot{x}^i = A^i x^i + B^i u^i + D^i, \tag{38}$$

where $x^i = [q^i, \dot{q}^i]^\top$ and

$$A^i = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{C^i}{H^i} \end{bmatrix}, \quad B^i = \begin{bmatrix} 0 \\ \frac{1}{H^i} \end{bmatrix}, \quad D^i = \begin{bmatrix} 0 \\ -\frac{D^i}{H^i} \end{bmatrix}.$$

For this case, the unknown linear system of Fig. 1 is defined by the dynamic interaction between the 4-DOF robot and the actuators such as is not possible to compute analytically the optimal gains K_1 and k_2. An stable reference model is proposed and used for each agent i

$$\dot{x}_r = A_r x_r + B_r u_r + D_r,$$
$$= \begin{bmatrix} 0 & 1 \\ -10 & -4 \end{bmatrix} x_r + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u_r + \begin{bmatrix} 0 \\ 0.02 \end{bmatrix}.$$

The same weight matrices $S = \text{diag}\{100, 1\}$ and $R = 0.1$ are used for each agent to compute the optimal hippocampus feedback controller. For this case, the ARE (11) is used instead of the MDRE for a fast policy inference. The optimal kernel matrix and control gain are

$$P = \begin{bmatrix} 19.4948 & 1.349 \\ 1.349 & 0.2198 \end{bmatrix},$$
$$K = \begin{bmatrix} 27.0224 & 4.4026 \end{bmatrix}.$$

The Q matrix is set as in the DC-servomechanism example. The solution of the Lyapunov equation is

$$M = \begin{bmatrix} 2.9004 & -4 \\ -4 & 10.9912 \end{bmatrix}.$$

The desired position $x_1^{d,i}$ of each agent is given by a square waveform function which is filtered by the following low-pass filter $G(s) = \frac{10}{s+10}$. The gains used in the update rules (32) and (33), the final estimates of $\widehat{K}_1$ and $\widehat{k}_2$, and the amplitudes and frequencies of $x_1^{d,i}$ are summarized in Table 1.

Fig. 9 shows the tracking results of the experience inference algorithm. Notice that despite having an almost linear plant due to the gear-ratio presented in the actuators' gearbox we still have a nonlinear disturbance that should affect the tracking precision. However, that problem does not seems to occur since the main goal of the neocortex/striatum is to guarantee that the robot's

Table 1
Neocortex/striatum gains and estimates.

Agent	Amplitude	Frequency	Γ	γ	$\widehat{K}_1$	$\widehat{k}_2$
1	1	0.2 Hz	$\begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$	0.05	$\begin{bmatrix} -130.1 & -21.72 \end{bmatrix}$	−0.06734
2	2	0.15 Hz	$\begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$	0.01	$\begin{bmatrix} -128.8 & -21.61 \end{bmatrix}$	−0.01916
3	$-\frac{\pi}{4}$	0.25 Hz	$\begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$	0.05	$\begin{bmatrix} -129.9 & -21.74 \end{bmatrix}$	−0.05787
4	$\frac{\pi}{3}$	0.3 Hz	$\begin{bmatrix} 5 & 0 \\ 0 & 0.5 \end{bmatrix}$	0.05	$\begin{bmatrix} -128.7 & -21.66 \end{bmatrix}$	−0.01346

states converges to the hippocampus reference position, that is, $x \to x_r$. This fact is more clear in the reference error e^i of Fig. 10.

Nevertheless, the main problem of the proposed approach is that the gains $\widehat{K}_1$ and $\widehat{K}_2$ could increase to large values (see Table 1) in their quest to reduce the reference error e^i. This idea is very common in manual tuning controllers where the user increases the control gains such that the error is decreased but the control input is increased.

We use the gain estimates of Table 1 and the hippocampus control policy u_r to validate the obtained results. A square wave form of amplitude 1 and frequency 0.5 Hz is used in each agent. Fig. 11 shows the obtained results. Notice that each agent exhibits the same performance of their respective hippocampus reference model and therefore we can conclude that the neocortex/striatum learning systems are successful to infer the experience of the hippocampus. However, Fig. 12 demonstrates that their control policies increase because the estimates $\widehat{K}_1$ and $\widehat{k}_2$ are not regulated.

6. Conclusions

In this paper, an experience inference approach for safety critical control of multi-agent linear systems is proposed. The algorithm is based on the complementary learning of the hippocampus, the striatum, and neocortex. The hippocampus is designed as a reference model controlled by an ADP algorithm which defines the desired performance to be inferred to the neocortex/striatum learning systems. The latter is designed as an actor control policy algorithm that obtains the gains that relate the hippocampus and neocortex control policies such that the trajectories of the unknown agent behaves as the trajectories of the hippocampus reference model. Experimental and simulation studies were carried out to verify the approach. The results show that the experience inference algorithm is able to transfer the desired performance to the unknown multi-agent linear system and the final control policy is unbiased with respect to the model uncertainties. However, the estimates of the gains could increase to large values which is directly reflected in the control input values. This issue is topic of our future work. In addition, further work aims to increase the scope of the approach using game theory approaches in either centralized or distributed plants.

Declaration of competing interest

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

References

[1] Khalil H. Nonlinear systems. Prentice Hall; 2002.
[2] Chen C-T. Linear system theory and design. Oxford University Press; 1999.
[3] Perrusquía A, Flores-Campos JA, Torres-San-Miguel CR. A novel tuning method of PD with gravity compensation controller for robot manipulators. IEEE Access 2020;8:114773–83. http://dx.doi.org/10.1109/ACCESS.2020.3003842.
[4] Lewis FL. Optimal control. New York, NY, USA: Wiley; 2012.
[5] Perrusquía A, Flores-Campos JA, Torres-Sanmiguel CR, González N. Task space position control of slider-crank mechanisms using simple tuning techniques without linearization methods. IEEE Access 2020;8:58435–42. http://dx.doi.org/10.1109/ACCESS.2020.2981187.
[6] Flores-Campos JA, Perrusquía A, Hernández-Gómez LH, González N, Armenta-Molina A. Constant speed control of slider-crank mechanisms: A joint-task space hybrid control approach. IEEE Access 2021;9:65676–87. http://dx.doi.org/10.1109/ACCESS.2021.3073364.
[7] Perrusquía A, Yu W. Human-in-the-loop control using Euler Angles. J Intell Robot Syst 2020;97:271–85. http://dx.doi.org/10.1007/s10846-019-01058-2.
[8] Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL. Optimal and autonomous control using reinforcement learning: A survey. IEEE Trans Neural Netw Learn Syst 2018;29(6):2042–62. http://dx.doi.org/10.1109/TNNLS.2017.2773458.
[9] Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement learning and feedback control using natural decision methods to design optimal adaptive controllers. IEEE Conrol Syst Mag 2012;32(6):76–105. http://dx.doi.org/10.1109/MCS.2012.2214134.
[10] Wiering M, van Otterlo M. Reinforcement learning: State-of-art. Springer; 2012.
[11] Sutton R, Barto A. Reinforcement learning: An introduction. Cambridge, MA: MIT Press; 1998.
[12] Buşoniu L, Babuška R, De Schutter B, Ernst D. Reinforcement learning and dynamic programming using function approximators. CRC Press; 2010.
[13] Ma L, Zhu F, Zhang J, Zhao X. Leader-follower asymptotic consensus control of multiagent systems: An observer-based disturbance reconstruction approach. IEEE Trans Cybern 2021. http://dx.doi.org/10.1109/TCYB.2021.3125332.
[14] Perrusquía A, Yu W. Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning. Internat J Robust Nonlinear Control 2020;30(7):2920–36. http://dx.doi.org/10.1002/rnc.4911.
[15] Vrabie D, Lewis FL. Neural networks approach for continuous-time direct adaptive optimal control for partially unknown nonlinear systems. Neural Netw 2009;22:237–46. http://dx.doi.org/10.1016/j.neunet.2009.03.008.
[16] Kiumarsi B, Lewis FL. Actor-critic based optimal tracking for partially unknown nonlinear discrete- time systems. IEEE Trans Neural Netw Learn Syst 2015;26(1):140–51. http://dx.doi.org/10.1109/TNNLS.2014.2358227.
[17] Modares H, Lewis FL, Jiang Z-P. H_∞ Tracking control of completely unknown continuous-time systems via off-policy reinforcement learning. IEEE Trans Neural Netw Learn Syst 2015;26(10):2550–62. http://dx.doi.org/10.1109/TNNLS.2015.2441749.
[18] Gheibi A, Ghiasi A, Ghaemi S, Badamchizadeh M. Designing of robust adaptive passivity-based controller based on reinforcement learning for nonlinear port-Hamiltonian model with disturbance. Internat J Control 2020;93(8):1754–64. http://dx.doi.org/10.1080/00207179.2018.1532607.
[19] Lewis FL, Jagannathan S, Yeşildirek A. Neural network control of robot manipulators and nonlinear systems. Taylor & Francis; 1999.
[20] Perrusquía A, Yu W. Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning. IEEE Trans Cybern 2020;1–10. http://dx.doi.org/10.1109/TCYB.2020.3028988.
[21] Modares H, Lewis FL, Naghibi-Sistani M-B. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. Automatica 2014;50(1):193–202. http://dx.doi.org/10.1016/j.automatica.2013.09.043.
[22] Perrusquía A, Flores-Campos JA, Yu W. Optimal sliding mode control for cutting tasks of quick-return mechanisms. ISA Trans 2021. http://dx.doi.org/10.1016/j.isatra.2021.04.033.
[23] Ramírez J, Yu W, Perrusquía A. Model-free reinforcement learning from expert demonstrations: A survey. Artif Intell Rev 2022;3213–41. http://dx.doi.org/10.1007/s10462-021-10085-1.
[24] Perrusquía A, Yu W, Li X. Nonlinear control using human behavior learning. Inform Sci 2021;569:358–75. http://dx.doi.org/10.1016/j.ins.2021.03.043.

[25] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33. http://dx.doi.org/10.1038/nature14236.
[26] Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ. Building machines that learn and think like people. Behav Brain Sci 2017;40. http://dx.doi.org/10.1017/S0140525X16001837.
[27] Kumaran D, Hassabis D, McClelland JL. What learning systems do intelligent agents need? Complementary learning systems theory updated. Trends in Cognitive Sciences 2016;20(7):512–34. http://dx.doi.org/10.1016/j.tics.2016.05.004.
[28] O'Reilly RC, Bhattacharyya R, Howard MD, Ketz N. Complementary learning systems. Cogn Sci 2014;38(6):1229–48. http://dx.doi.org/10.1111/j.1551-6709.2011.01214.x.
[29] Blakeman S, Mareschal D. A complementary learning systems approach to temporal difference learning. Neural Netw 2020;122:218–30. http://dx.doi.org/10.1016/j.neunet.2019.10.011.
[30] McClelland JL, McNaughton BL, O'Reilly RC. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Psychol Rev 1995;102(3):419. http://dx.doi.org/10.1037/0033-295X.102.3.419.
[31] Vilà-Balló A, Mas-Herrero E, Ripollés P, Simó M, Miró J, Cucurell D, et al. Unraveling the role of the hippocampus in reversal learning. J Neurosci 2017;37(28):6686–97. http://dx.doi.org/10.1523/JNEUROSCI.3212-16.2017.
[32] Stachenfeld KL, Botvinick MM, Gershman SJ. The hippocampus as a predictive map. Nature Neurosci 2017;20(11):1643–53. http://dx.doi.org/10.1038/nn.4650.
[33] Ólafsdóttir HF, Bush D, Barry C. The role of hippocampal replay in memory and planning. Curr Biol 2018;28(1):R37–50. http://dx.doi.org/10.1016/j.cub.2017.10.073.
[34] Mattar MG, Daw ND. Prioritized memory access explains planning and hippocampal replay. Nature Neurosci 2018;21(11):1609–17. http://dx.doi.org/10.1038/s41593-018-0232-z.
[35] Schultz W, Apicella P, Scarnati E, Ljungberg T. Neuronal activity in monkey ventral striatum related to the expectation of reward. J Neurosci 1992;12(12):4595–610. http://dx.doi.org/10.1523/JNEUROSCI.12-12-04595.1992.
[36] Perrusquía A. Human-behavior learning: A new complementary learning perspective for optimal decision making controllers. Neurocomputing 2022;157–66. http://dx.doi.org/10.1016/j.neucom.2022.03.036.
[37] Luviano D, Yu W. Continuous-time path planning for multi-agents with fuzzy reinforcement learning. J Intell Fuzzy Systems 2017;33:491–501. http://dx.doi.org/10.3233/JIFS-161822.
[38] Perrusquía A, Yu W, Li X. Multi-agent reinforcement learning for redundant robot control in task-space. Int J Mach Learn Cybern 2021;12(1):231–41. http://dx.doi.org/10.1007/s13042-020-01167-7.
[39] Perrusquía A, Yu W, Soria A. Large space dimension reinforcement learning for robot position/force discrete control. In: 2019 6th International conference on control, decision and information technologies. IEEE; 2019, p. 91–6. http://dx.doi.org/10.1109/CoDIT.2019.8820575.
[40] Perrusquía A, Yu W. Continuous-time reinforcement learning for robust control under worst-case uncertainty. Internat J Systems Sci 2021;52(4):770–84. http://dx.doi.org/10.1080/00207721.2020.1839142.
[41] Kamalapurkar R, Walters P, Dixon W. Model-based reinforcement learning for approximate optimal regulation. Automatica 2016;64:94–104. http://dx.doi.org/10.1016/j.automatica.2015.10.039.
[42] Zhang H, Liu D, Luo Y, Wang D. Adaptive dynamic programming for control. London, U.K.: Springer-Verlag; 2013.
[43] Vamvoudakis K, Lewis FL. On-line actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 2010;46:878–88. http://dx.doi.org/10.1016/j.automatica.2010.02.018.
[44] Vamvoudakis KG. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. Systems Control Lett 2017;14–20. http://dx.doi.org/10.1016/j.sysconle.2016.12.003.
[45] Perrusquía A, Yu W. Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview. Neurocomputing 2021;438:145–54. http://dx.doi.org/10.1016/j.neucom.2021.01.096.
[46] Adam S, Buşoniu L, Babuška R. Experience replay for real-time reinforcement leaening control. IEEE Trans Syst Man Cybern 2012;42(2):201–12. http://dx.doi.org/10.1109/TSMCC.2011.2106494.
[47] Liu R, Zou J. The effects of memory replay in reinforcement learning. In: International conference on machine learning 2017 workshop on principled approaches to deep learning. 2017, http://dx.doi.org/10.1109/ALLERTON.2018.8636075.
[48] Zha D, Lai K-H, Zhou K, Hu X. Experience replay optimization. In: Proceedingds of the twenty-eighth international joint conference on artificial intelligence. 2019, arXiv:1906.08387.
[49] Perrusquía A, Yu W, Soria A. Position/force control of robot manipulators using reinforcement learning. Ind Robot: Int J Robot Res Appl 2019;46(2):267–80. http://dx.doi.org/10.1108/IR-10-2018-0209.
[50] Singh S, Sutton S. Reinforcement learning with replacing elegibility traces. Mach Learn 1996;22:123–58. http://dx.doi.org/10.1023/A:1018012322525.
[51] Perrusquía A, Yu W. Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning. IEEE Trans Neural Netw Learn Syst 2020;32(11):4879–89. http://dx.doi.org/10.1109/TNNLS.2020.3026010.
[52] Kiumarsi B, Lewis FL, Modares H, Karimpor A, Naghibi-Sistani M-B. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. Automatica 2014;50:1167–75. http://dx.doi.org/10.1016/j.automatica.2014.02.015.
[53] Wang C, Li Y, Sam Ge S, Heng Lee T. Optimal critic learning for robot control in time-varying environments. IEEE Trans Neural Netw Learn Syst 2015;26(10):2301–10. http://dx.doi.org/10.1109/TNNLS.2014.2378812.
[54] Al-Tamimi A, Lewis F, Abu-Khalaf M. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. IEEE Trans Syst, Man, Cybern Part B, Cybern 2008;38(4):943–9. http://dx.doi.org/10.1109/ADPRL.2007.368167.
[55] Palanisamy M, Modares H, Lewis FL, Aurangzeb M. Continuous-time Q-learning for infinite horizon-discounted cost linear quadratic regulator problems. IEEE Trans Cybern 2015;45(2):165–76. http://dx.doi.org/10.1109/TCYB.2014.2322116.
[56] Perrusquía A, Yu W. Neural $\mathcal{H}_2$ control using reinforcement learning for unknown nonlinear systems. In: 2020 International joint conference on neural networks. IEEE; 2020, p. 1–7. http://dx.doi.org/10.1109/IJCNN48605.2020.9207547.
[57] Perrusquía A. Robust state/output feedback linearization of direct drive robot manipulators: A controllability and observability analysis. Eur J Control 2022;64:100612.
[58] Yu W, Perrusquía A. Simplified stable admittance control using end-effector orientations. Int J Soc Robot 2020;12(5):1061–73. http://dx.doi.org/10.1007/s12369-019-00579-y.
[59] Perrusquía A, Garrido R, Yu W. Stable robot manipulator parameter identification: A closed-loop input error approach. Automatica 2022;141:110294. http://dx.doi.org/10.1016/j.automatica.2022.110294.



https://dspace.org/

Hippocampus experience inference for safety critical control of unknown multi-agent linear systems

Perrusquía, Adolfo; Guo, Weisi

2023-06-01
Elsevier
https://doi.org/10.1016/j.isatra.2022.12.011
Perrusquia A, Guo W. (2023) Hippocampus experience inference for safety critical control of unknown multi-agent linear systems. ISA Transactions, Volume 137, June 2023, pp. 646-655

Downloaded from a DSpace repository. https://dspace.org/