Neurocomputing 489 (2022) 157–166

Contents lists available at ScienceDirect

Neurocomputing

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

Survey paper

Human-behavior learning: A new complementary learning perspective for optimal decision making controllers

Adolfo Perrusquía *

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

ARTICLE INFO

Article history:
Received 9 July 2021
Revised 5 November 2021
Accepted 14 March 2022
Available online 17 March 2022
Communicated by Zidong Wang

Keywords:
Human-behavior learning
reinforcement learning
Cognitions
Neocortex
Hippocampus
Striatum

ABSTRACT

This paper reviews an almost new method for the design of optimal decision making controllers named as Human-Behavior learning. This new paradigm is inspired by the complementary learning that different areas of the human brain have to improve learning and experience transference. It is shown that independent and well identified sources of knowledge can enhance learning and facilitate the design of the optimal decision making controller. This interaction is modelled as a Markov Decision Process defined by a tuple of actions, cognitions, and emotions sets. Existing methods of both control and reinforcement learning theories are reviewed and connected to complete the behavior learning picture for a class of linear systems.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

This article gives an overview of a new approach named human-behavior learning [1] for the design of optimal decision making controllers [2]. As it is well known, humans are capable to learn new things and take decisions in accordance to their observations, experience, and curiosity. There exists several studies that seek to understand how human learns and which specific area or areas of the brain are responsible for carrying out the learning process [3,4]. In [5] is stated that the brain requires two different specialized learning and memory systems given by the hippocampus and the neocortex. On the one hand, the hippocampus is modelled as a sparse and pattern-separated system for fast learning episodic memories. On the other hand, the neocortex is modelled as a well distributed pattern-dependent system for efficient generalization that is slow in comparison to the hippocampus system [6].

There exist a strong connection between reinforcement learning (RL) and human-behavior learning since both are governed under a similar philosophy of operation, that is, obtain an optimal control policy/ optimal decision making control by minimizing a predefined utility function in a discounted or infinite horizon [7,8]. Throughout the paper, the differences between RL and human-behavior learning will be point out to exhibit the main advantages that offer the human-behavior learning approach.

Previous work [2] states that human-behavior is modelled as an interplay of three main sets: actions, cognitions, and emotions. The main differences between reinforcement learning and human-behavior learning lie in the cognitive and emotions sets that provide several tools to incorporate different sources of knowledge and establish their relation to achieve complex behaviors [2,9]. In fact, reinforcement learning (RL) is a cognitive model of the human-behavior approach. This paper aims to provide different cognitive models and relate them with three specialized learning systems: the neocortex, the hippocampus, and the striatum.

The emotions [10] set gives feedback of the performance of the controller in a more complex way than static reward functions due to its dependence on the cognitions set. However, they will not be discussed in this paper. Recall that adaptive dynamic programming (ADP)/RL [11] algorithms connect adaptive control [12] and optimal control [13,14] theories for the design of optimal adaptive controllers [15–18] by adjusting their actions based on reward or stimuli received from the environment. For sake of simplicity, the emotions set are constrained as the reward or utility function ADP and RL problems.

The human neocortex or neocortical system consist of well distributed, interconnected, and usually bidirectional structures which executes small updates to the connections among the structures involved in the learning process so that the activity in one

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

https://doi.org/10.1016/j.neucom.2022.03.036
0925-2312/© 2022 Elsevier B.V. All rights reserved.

region of the brain can give rise to activity in many other regions [5]. This updates are small in magnitude such that they not generate good pattern association in the short term. However, in the long-term the associations become stronger and provide the basis for correct performance in several tasks (e.g., semantic and episodic). In the context of human-behavior learning, the neocortex can be modelled as intelligent systems such as neural networks [19], ADP, and RL architectures [20–22]. These methods are well known data-dependent techniques that exhibit slow learning phase but good pattern association, adaptability, and generalization. However, the neocortex is not a completely independent learning system since it is supported by the hippocampus. This fact will be addressed throughout the article.

The human hippocampus or hippocampal system is referred to a system that is specialized for the rapid formation of new memories. There exists an extensive work that address the importance and functionalities of the hippocampus for decision making problems [23–26]. Furthermore, some authors [5] have suggested that the hippocampus is the teacher of the neocortex because provides of memory and reinstatement of the neocortex activity. There exists several memory architectures that enhance the performance of neural networks and ADP/RL algorithms such as: recurrent neural networks [27], eligibility traces [28], model reference adaptive controllers [29], and experience replay [2]. Furthermore, previous knowledge such as dynamical parameters and expert's data can be used for experience transference to avoid learning from scratch.

In view of the above, model-based and model-free ADP/RL algorithms show a good interplay between neocortex and hippocampus learning systems. Each ADP/RL method can be model-based [30] or model-free [31–33]. Temporal difference methods offer a friendly way to incorporate memory and previous knowledge, e.g., actor-critic algorithms [34,35], value and policy iteration algorithms, and Q-learning [36]. A complete overview of reinforcement learning methods can be found [14,29,37].

The use of function approximators [38,39] offer a simple way to express complex behaviors with a simple set of basis functions and weights. Convergence of the weights' of the function approximator can be achieved under the fulfilment of a persistency of excitation condition [40,41] which is provided by the hippocampus learning system. This excitation condition gives information of the performance of the system under certain inputs and states. This technique is equivalent to an exploration strategy [42] that serves to acquire new information and give more resources for learning [43].

The striatum or corpus striatum is a structure that is responsible for evaluating sensory inputs and different sources of knowledge for decision making [44]. In general, the striatum mediates cognition, reinforcement, and reward systems. In this context, the interaction between the hippocampus, neocortex, and striatum exhibits complementary learning properties [45,46] that enable complex behavior. Whilst the neocortex learns distributed representations for good generalization, the hippocampus provide of fast learning memories and skills that support the neocortex via the striatum. This paper posits that human-behavior learning is a complementary learning algorithm that enhance learning by combining different semi-independent cognitive models.

Throughout this paper, $\mathbb{N}, \mathbb{R}, \mathbb{R}_+, \mathbb{R}^n, \mathbb{R}^{n\times m}$ denote the spaces of natural numbers, real numbers, positive real numbers, real n-vectors, and real $n \times m$-matrices, respectively; $\otimes$ denotes the Kronecker product; where $x \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}$ and $n, m \in \mathbb{N}$.

2. Human-Behavior Learning Formulation

In the recent work [2] is stated that the human-behavior learning approach can be modelled as a Markov Decision Process (MDP) composed of a tuple of actions $\mathscr{A}$, cognitions $\mathscr{C}$, and emotions $\mathscr{E}$. Fig. 1 shows the interplay of the human-behavior sets.

The set of actions $\mathscr{A}$ is modelled as any signal that can be measured, for example, the internal states, the control input, the output, and exogenous disturbances (if possible) of a given system. The set of cognitions $\mathscr{C}$ is one of the central parts of the human-behavior learning approach since it is inspired in the complementary learning of the hippocampus, neocortex, and the striatum learning systems of the human brain [46]. In other words, the set $\mathscr{C}$ contains different and semi-independent sources of knowledge which enables fast learning, generalization, and experience transference. The set of emotions $\mathscr{E}$ gives feedback about the performance of the system in accordance to the actual actions and cognitive models. Notice that this definition is almost equivalent to the reward or utility function in reinforcement learning [13]. However, the emotions differ from rewards in the dependence on the cognitions set $\mathscr{C}$. For sake of simplicity, this paper assumes that the emotions are equivalent to reward functions such that dependence on the cognitive set is avoided.

2.1. Reinforcement Learning or Human-Behavior Learning?

A close relationship between reinforcement learning and human-behavior learning can be distinguished. Whilst reinforcement learning seeks the optimal control policy that minimizes a pre-defined utility function [37], the human-behavior learning approach seeks the optimal decision making controller (also known as control policy) that minimizes a pre-defined utility function by combining different sources of knowledge [2].

For this study the main difference between reinforcement learning and human-behavior learning relies in the cognitive set. It is evident that reinforcement learning is a cognitive model of the human-behavior learning approach. Therefore, human-behavior learning aims to provide a complete picture of how human learns and use reinforcement learning as a cognitive model to support this new perspective.

A cognitive model f_i is understood as any method that enables adaptability and establishes relations or rules based on previous data or experience, for example, neural networks [40] is a well known cognitive method that seeks the connection weights of multiple neurons to estimate/predict the performance of a system. A skill s_i is any method that gives useful information that improves and facilitates the learning activity of the cognitive model, for example, dynamic knowledge [27] helps the cognitive model not start from scratch and provides a good direction for updating.

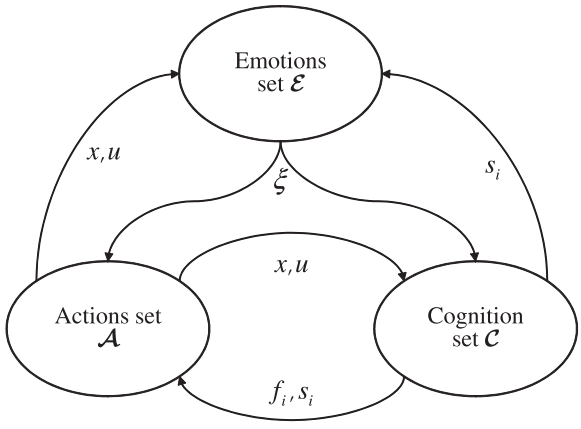

Fig. 1. Human-behavior learning diagram. It is composed of the three main sets: actions $\mathscr{A}$, the cognitions $\mathscr{C}$, and the emotions $\mathscr{E}$; which are connected in a "complex" fashion to achieve complex performances.

There can be as many cognitive models f_i and skills s_i in the cognitive set $\mathscr{C}$.

The cognitions set $\mathscr{C}$ can be divided in three main subsets: the neocortex, the hippocampus, and the striatum [5]. Each subset has specific cognitive models and skills that improves learning and decision making. Fig. 2 summarizes some of the most relevant examples of cognitive models and skills of the neocortex, hippocampus, and the striatum learning systems.

It is quite challenging to separate the functionalities of the neocortex, hippocampus, and striatum learning systems since they are closely related. In the next sections, each learning system is analyzed and is defined how they complement each other.

2.2. The lack of Adaptability of Dynamic Programming Algorithms: The Linear Quadratic Regulator case

The following linear system [47] is considered

$$\dot{x} = Ax + Bu, x(t_0) = x_0. \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ define the dynamics of (1), $x \in \mathbb{R}^n$ denote the states and $u \in \mathbb{R}^m$ is the control input. Notice that $(x, u) \in \mathscr{A}$. It is assumed that the pair (A, B) is controllable and known in advance.

The optimal control objective is to seek a control input that minimizes the following value function [48]

$$V(x) = \int_t^\infty \xi(x, u)d\tau, \tag{2}$$

where $\xi(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a predefined utility function (or reward function) from the emotions set $\mathscr{E}$. For the linear quadratic regulator (LQR) problem [13], the utility function is set to a quadratic function of the form $\xi(x, u) = x^\top Sx + u^\top Ru$ where $S = S^\top \geqslant 0 \in \mathbb{R}^{n\times n}$ and $R = R^\top > 0 \in \mathbb{R}^{m\times m}$ denote known weight matrices.

The LQR problem can be regarded as a hippocampus learning system due to its fast learning and pattern independent, that is, the optimal control is derived from the solution of the next algebraic Riccati equation (ARE) [49]

$$A^\top P + PA + S - PBR^{-1}B^\top P = 0, \tag{3}$$

for some kernel matrix $P = P^\top > 0 \in \mathbb{R}^{n\times n}$. Then, the optimal control policy (optimal decision making function) is

$$u^* = -R^{-1}B^\top Px. \tag{4}$$

Notice that the ARE (3) does not require measurements of the states and control input to derive the optimal control law. Therefore, the ARE is the first hippocampus model in the human-behavior learning perspective. However, the LQR solution has a poor performance against disturbances or parameter changes [50,51]. For this reason, other complementary techniques have to been used to achieve both adaptability and optimality of the final control policy. These techniques are analyzed in the next sections.

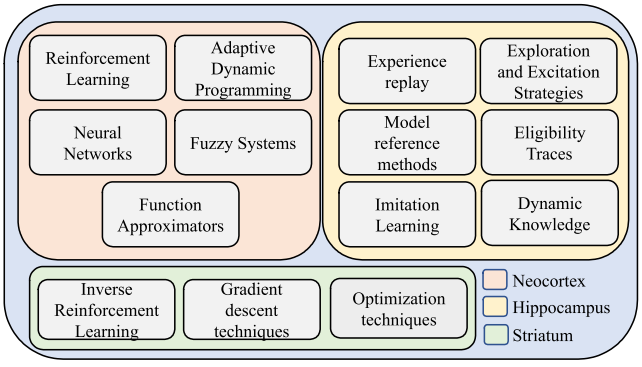

Fig. 2. Cognition set $\mathscr{C}$ elements. The set is divided in three subsets defined by the neocortex, hippocampus, and the striatum.

3. Neocortex learning system

The neocortex learning system defines one of the central parts for behavior learning. This system provides of different tools that enables learning by interacting with on-line data measured along the system trajectories [14]. The neocortex slowly learns distributed associations for efficient knowledge generalization. Some of the neocortex learning models are given by ADP/RL algorithms and function approximators [38], which are described below.

Remark 1. In this work, the ADP/RL algorithms are expressed in a general way and are not categorized as value and policy iteration algorithms. References are provided to complete the picture of each approach.

3.1. Function Approximators, Neural Networks, and Fuzzy Systems

Before explaining some of the cognitive models of the neocortex learning system, it is crucial to introduce function approximators [52] as the first cognitive skill.

Recall that a skill is a tool that human uses to facilitate learning by providing simpler relations or functions. So, function approximators is a skill of the neocortex learning system that provides a simpler basis of functions that can approximate the performance of a relatively more complex function [29].

The structure of a value function approximator (VFA) is [53]

$$V(x) = \phi^\top\theta + \varepsilon_V, \tag{5}$$

where $\phi = \phi(x) : \mathbb{R}^n \to \mathbb{R}^p$ is the vector of basis functions, $\theta \in \mathbb{R}^p$ is a vector of unknown parameters, and $\varepsilon_V \in \mathbb{R}$ denote the residual error of the approximator. Furthermore, the approximator can be written in terms of the well known action value function $Q(x, u)$ [54] as

$$Q(x, u) = \psi^\top\vartheta + \varepsilon_Q, \tag{6}$$

where $\psi = \psi(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ is the vector of basis functions that depends of both x and u, $\vartheta \in \mathbb{R}^p$ is the parameters vector, and $\varepsilon_Q \in \mathbb{R}$ is the residual error of the Q-function approximation. However since the parameters θ of (5) and ϑ of (6) are unknown, then the next value function and action value function approximators are used

$$\begin{aligned} \widehat{V}(x) &= \phi^\top\widehat{\theta} \\ \widehat{Q}(x, u) &= \psi^\top\widehat{\vartheta} \end{aligned} \tag{7}$$

where $\widehat{\theta}, \widehat{\vartheta} \in \mathbb{R}^p$ are estimates of θ and ϑ, respectively. In the context of machine learning, there exists a wide variety of basis functions that can be used for the design of function approximators such as: local models, radial basis functions (RBFs) [21], Gaussian mixture models (GMMs) [55], neighbors [56,50], linear models [15], among others. A complete survey and convergence properties of the approximators are given in [38].

In the context of optimal control, the basis functions are commonly quadratic [27], for example, let $x = [x_1, x_2]^\top$, for some $x_1, x_2 \in \mathbb{R}$, then the set of basis functions are $\phi(x) = \left[x_1^2, x_1x_2, x_2^2\right]^\top$. This operation is generally expressed by the symmetric Kronecker product $x \otimes x \in \mathbb{R}^{\frac{n(n+1)}{2}}$. In this case the residual error ε_V and ε_Q are

zero because a complete and exact set of basis-functions are provided [54].

It should be note that Fig. 2 states that the neocortex learning system contains function approximators, neural networks[49], and fuzzy systems [57,58] as separate cognitive skills. It is well known that both neural networks and fuzzy systems are a special kind of function approximators. However, the scope of these methods is deeper, that is, in accordance to the Weierstrass higher-order approximation theorem [53] there exists a complete independent basis set that uniformly approximate the value function as $p \to \infty$, and hence the residual $\varepsilon_V \to 0$ or $\varepsilon_Q \to 0$. These kind of approximators provide a feasible way to introduce adaptability and robustness in the final decision making function.

3.2. Adaptive Dynamic Programming (ADP)

Adaptive Dynamic Programming is the first cognitive model of the neocortex learning system and is commonly used if there exists dynamic knowledge provided by the hippocampus learning system. This techniques merge the capabilities of optimal and adaptive control theories in a single algorithm to achieve optimal performances and adaptability in the learning process [14]. In contrast with the ARE of previous section, ADP overcomes the instability problem and improves the performance of the decision making controller by adjusting the weights of the value function approximator. Two different ADP approaches can be distinguished: full model-based and partially model-based [37].

3.2.1. Full Model-based ADP

Full model-based ADP computes the optimal decision making controller by solving a Hamilton–Jacobi-Bellman (HJB) equation in an iterative fashion [22,59]. Let define three different HJB equations as

$$H(x,u) = x^\top P(Ax+Bu) + (Ax+Bu)^\top Px + \xi(x,u) = 0, \tag{8}$$

$$H(x,u) = \theta^\top \nabla\phi(Ax+Bu) + \xi(x,u) = \varepsilon_H, \tag{9}$$

$$\widehat{H}(x,u) = \widehat{\theta}^\top \nabla\phi(Ax+Bu) + \xi(x,u) = \delta_1, \tag{10}$$

where $\nabla = \frac{\partial}{\partial x}$ represents the nabla operator, $\varepsilon_H \in \mathbb{R}$ is the residual error caused by the approximation error ε_V, and $\delta_1 \in \mathbb{R}$ is known as the temporal difference (TD) error which is an approximation of the residual error ε_H. The TD error is inspired in the behaviour of phasic midbrain dopaminergic neurons which show some relationship to reward (utility function) and sensory input (action set $\mathscr{A}$). Although, dopaminergic neurons correspond to less than 1% of the total number of brains neurons, they innervate including motor behavior, motivation and working memory, that is, they play an important role in the reward system that controls the learning of many human behaviors and enables decision making [60,61].

The HJB Eq. (8) is the neocortex version of the ARE (3) of the hippocampus learning system, which takes into account measurements of the action space $\mathscr{A}$, that is, x and u. This is a general version of the ARE (3) which can be solved on-line by a differential Ricatti equation (DRE) of the form [49]

$$-\dot{P} = A^\top P + PA - PBR^{-1}B^\top P + S. \tag{11}$$

The HJB Eq. (9) is an approximation of (8) using any function approximator discussed in the previous section. Assume that a function approximator with estimation error ε_V is used. Then, the optimal decision making function is biased by a residual error $\varepsilon_H = -\nabla\varepsilon_V(Ax+Bu)$ which can be attenuated by improving the design of the approximator [53].

The HJB Eq. (10) is an approximation of (9) and is used for the design of the ADP algorithm. The main goal of any ADP algorithm is to minimize a quadratic cost in terms of the TD error [32], that is,

$$E = \frac{1}{2}\delta_1^2, \tag{12}$$

such that $\widehat{\theta} \to \theta$ and $\delta_1 \to \varepsilon_H$ as $t \to \infty$. To compute the estimates $\widehat{\theta}$, we need to use the striatum learning system which provide tools for updating and decision making. The standard solutions are based on a batch least-squares (LS) rule [32] and the gradient-descent technique [49] as

$$\widehat{\theta} = -\left(q_1 q_1^\top\right)^{-1} q_1 \xi(x,u), \tag{13}$$

$$\dot{\widehat{\theta}} = -\alpha q_1 \left(q_1^\top \widehat{\theta} + \xi(x,u)\right), \tag{14}$$

where $\alpha \in \mathbb{R}_+$ is the learning rate and $q_1 = \nabla\phi(Ax+Bu)$. For the batch LS rule, it is stored at least $l \geqslant p$ samples in order to estimate the parameter vector $\widehat{\theta}$. The decision making controller is computed at each iteration as

$$\widehat{u} = -R^{-1}B^\top \nabla^\top \phi \widehat{\theta}. \tag{15}$$

Here we can see the advantage of the neocortex over the hippocampal learning system. The weights $\widehat{\theta}$ takes into account the dynamic knowledge of the hippocampus and measurements of the action set $\mathscr{A}$, this complementary sources of knowledge gives the controller (15) adaptability to new changes in the system dynamics (1) and overcome possible instability problems or poor performances [50]. Despite the hippocampus learning system gives the ideal optimal controller, it lacks of robustness and may not present good performance in presence of exogenous and endogenous disturbances.

3.2.2. Partially Model-based ADP

Partially model-based ADP can be seen as a first approach for reinforcement learning [14]. In this case, matrix A is unknown [53,62]. To overcome this issue the Bellman equation form (also known as integral reinforcement learning)[63,64] is used in the value function definition as

$$0 = x^\top Px|_t^{t+T} - \int_t^{t+T} \xi(x,u)d\tau, \tag{16}$$

$$\varepsilon_T = \theta^\top q_2 - \int_t^{t+T} \xi(x,u)d\tau, \tag{17}$$

$$\delta_2 = \widehat{\theta}^\top q_2 - \int_t^{t+T} \xi(x,u)d\tau, \tag{18}$$

where $q_2 \in \mathbb{R}^p$ is a set of basis functions in terms of $x(t)$ and $x(t+T)$, $\varepsilon_T \in \mathbb{R}$ is the residual error of the function approximator $q_2^\top\theta$, and $\delta_2 \in \mathbb{R}$ is the TD error of the Bellman equation in terms of the estimates $\widehat{\theta}$. Notice that the above Bellman equations use only measurements of the action set $\mathscr{A}$ and do not require knowledge of the system dynamics A and B. The batch LS rule (13) and the gradient descent rule (14) are slightly modified as

$$\widehat{\theta} = \left(q_2 q_2^\top\right)^{-1} q_2 \int_t^{t+T} \xi(x,u)d\tau, \tag{19}$$

$$\dot{\widehat{\theta}} = -\alpha q_2 \left(q_2^\top \widehat{\theta} - \int_t^{t+T} \xi(x,u)d\tau\right), \tag{20}$$

The optimal decision making controller is computed by (15) which depends on the knowledge of the input matrix B. There exist other algorithms in the ADP framework, such as actor-critic algorithms [65] which use either the batch-least square rule or the gradient rule to estimate the value function and another approximator to compute the decision making controller. However, these techniques are commonly used to avoid a boundedness assumption in the control input [53].

3.3. Reinforcement Learning

Reinforcement learning (RL) has become a leading technique in both machine learning [66–68] and ADP techniques [37]. In contrast to ADP techniques, the philosophy of RL is to design completely model-free algorithms that achieve an adaptive optimal performance [27]. Traditional RL can be seen as a purely neocortex cognitive model since they learn from scratch both the optimal value function and the optimal decision making function. However, it is well known that RL requires of complementary techniques to guarantee parameter convergence and achieve the desired optimal performance.

The most used RL algorithm is Q-learning which uses an action value function $Q(x,u)$ instead of the standard value function $V(x)$. Here the Q-function satisfies [54]

$$Q(x,u) = V^*(x) + H(x,u), \tag{21}$$

where $V^* = x^\top Px$ is the optimal value function. The Q-function evaluated in the optimal decision making controller (4) is equivalent to the optimal value function, that is, $Q(x,u^*) = V^*(x)$. The Q-function has the following structure [48]

$$Q(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = Z^\top MZ, \tag{22}$$

where $Q_{xx} \in \mathbb{R}^{n\times n}, Q_{xu} \in \mathbb{R}^{n\times m}, Q_{uu} \in \mathbb{R}^{m\times m}, M \in \mathbb{R}^{(n+m)\times(n+m)}$, and $Z = [x^\top, u^\top]^\top$.

The Bellman equations are written in terms of the Q-function as

$$0 = Z^\top MZ|_t^{t+T} - \int_t^{t+T} \xi(x,u)d\tau \tag{23}$$

$$\varepsilon_M = \vartheta^\top q_3 - \int_t^{t+T} \xi(x,u)d\tau \tag{24}$$

$$\delta_3 = \widehat{\vartheta}^\top q_3 - \int_t^{t+T} \xi(x,u)d\tau \tag{25}$$

where q_3 is a set of basis functions for $Z(t)$ and $Z(t+T), \varepsilon_M \in \mathbb{R}$ is the residual error of the function approximator $q_3^\top\vartheta$, and δ_3 is the TD error of the Q-function Bellman equation in terms of the estimate vector $\widehat{\vartheta}$.

The batch LS update rule (19) and the gradient descent rule (20) hold for the Q-function approximator. The optimal decision making controller is computed as

$$\widehat{u} = -\widehat{Q}_{uu}^{-1}\widehat{Q}_{xu}^\top x, \tag{26}$$

where $\widehat{Q}_{uu}$ and $\widehat{Q}_{xu}$ are estimates of Q_{uu} and Q_{xu} which are obtained from the estimate vector ϑ. In contrast with the partially model-based ADP, the Q-learning algorithm does not require dynamic knowledge. However, the learning phase is slow in contrast with ADP methods.

Another RL approach uses a value function $V(x)$ with a batch LS rule to compute both the optimal value function and the optimal decision making function [48]. Two key ideas are required: (1) express the control as $u = -Kx$, where $K = R^{-1}B^\top P \in \mathbb{R}^{m\times n}$ is the stabilizing gain, and (2) initialize the learning with a stabilizing control gain K. The above conditions are not very restrictive and are easy to fulfil. The RL algorithm uses a quadratic parametrization of the Bellman Eq. (16) as

$$x^\top Px|_t^{t+T} - 2\int_t^{t+T} (u+Kx)^\top RKxd\tau = -\int_t^{t+T} x^\top(S + K^\top RK)xd\tau, \tag{27}$$

The above notation for the Bellman equation facilitates the estimation of both P and K using the batch-least squares rule (19). The main constraints of this approach are that can only be solved by a batch LS or a recursive least-squares method, and require an exact set of quadratic basis function in terms of x and u.

Other reinforcement learning algorithms in the machine learning framework [11] can also be used to achieve a local optimal decision making controller. However, they are not in the scope of the proposed human-behavior learning approach. In the next section, the cognitive models and skills of the hippocampus learning system are briefly described.

4. Hippocampus learning system

The hippocampus learning system provides of fast learning tools that improves significantly the performance of the neocortex learning system. Although they are opposites learning systems, they complement each other to achieve adaptive and optimal decision making controllers [6]. The hippocampus is directly related with short-term and long-term memories, that is, it incorporates previous knowledge and past experiences into the neocortex models to enhance the learning procedure in gradient update rules. Some of the hippocampus cognitive models and skills are described below.

4.1. Exploration/Excitation signals

The exploration/excitation signals are skills that provide enough information and excitation [69] of the system modes for accurate parameter estimation. There exists several exploration techniques in the context of machine learning, for example, ε-greedy exploration [52], Thompson sampling[7], Boltzmann exploration [38], upper confidence bounds, memory-based exploration [42], among others. A complete survey of these techniques are given in [42]. However, these techniques are not common in ADP/RL problems since they require additional assumptions in the control actions which yields to local optimal solutions that differ from the optimal control philosophy [50].

In the optimal control context, the exploration/excitation signals are designed to fulfil a persistency of excitation (PE) condition [70] to guarantee parameter convergence. The PE condition requires that the following condition [40] on the regressor matrix q (either q_1, q_2, q_3) is fulfilled

$$\beta_0 I_p \leqslant \int_t^{t+T} qq^\top d\tau \leqslant \beta_1 I_p, \tag{28}$$

for some constants $\beta_0, \beta_1 \in \mathbb{R}_+$ and time constant $T \in \mathbb{R}_+$. This condition means that the matrix $qq^\top \in \mathbb{R}^{p\times p}$ is full rank and its integral in a time interval $[t : t+T]$ is bounded. Some examples of PE signals v are given in Table 1 where a_i and ω_i denote the amplitude and frequency of the sinusoidal functions, κ is the exponential decay factor, and $\mathcal{N}(0,\sigma^2)$ denote a Gaussian distribution with mean zero and variance σ^2.

4.2. Eligibility traces

The first memory function that is discussed in this section is known as eligibility traces [7,71]. This technique is a long-short term memory that corrects the TD error by a change in time of the value function $V(x)$ or action value function $Q(x,u)$ [72]. Sup-

Table 1
Examples of PE signal v.

Sinusoidal functions	Exponential decaying signal	Gaussian noise
$\sum_{i=1}^N a_i \sin(\omega_i t)$	$\sum_{i=1}^N a_i e^{-\kappa t}\sin(\omega_i t)$	$\sim \mathcal{N}(0,\sigma^2)$

pose that we are analysing the TD error δ (either δ_1, δ_2, or δ_3) in a time $t_0 \in [t, \infty)$. Then all samples that occur in a time $t \leqslant t_0$ will be taken into account by the value function, that is,

$$\begin{aligned} V(x) &= \int_t^{\infty} \xi(x,u)d\tau \\ &= \int_t^{t_0} \xi(x,u)d\tau + \cancelto{0}{\int_{t_0}^{\infty} \xi(x,u)d\tau} \end{aligned} \tag{29}$$

Then a temporal profile $W(t)$ of the value function $V(x)$ can be defined as

$$W(t) = \begin{cases} 1 & \text{if } t \leqslant t_0 \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Recall that the TD error is linear with respect to the utility function $\xi(x,u)$ as it is shown in (10), (18), and (25), then the temporal profile can be modified as

$$\widehat{W}(t) = \begin{cases} \delta(t_0) & \text{if } t \leqslant t_0 \\ 0 & \text{otherwise}, \end{cases} \tag{31}$$

The key idea is to update the estimates of the gradient rule by weighting previous samples with the value of the actual TD-error.

$$\begin{aligned} \dot{\widehat{\theta}} &= -\alpha \textstyle\int_{-\infty}^{t_0} \widehat{W}(\tau) q(\tau) d\tau \\ &= -\alpha \delta(t_0) \textstyle\int_{-\infty}^{t_0} q(\tau) d\tau. \end{aligned} \tag{32}$$

Notice that the eligibility traces is in fact an integrator system which satisfies the following dynamic system for any time t

$$\begin{aligned} \dot{\widehat{\theta}} &= -\alpha \delta e_t \\ \dot{e}_t &= q \end{aligned} \tag{33}$$

The dynamic system (33) is not commonly used in applications since it is required that the eligibility trace be bounded in every time step. The classical eligibility traces approach [52] is based on the following discount value function

$$V(x) = \int_t^{\infty} e^{-\gamma(t-\tau)} \xi(x,u) d\tau, \tag{34}$$

where $\gamma \in \mathbb{R}_+$ is a discount factor. The discounted value function (34) is used to weight more recent time steps than past ones and ensure convergence [38]. With this slightly modification, the update rule (33) is changed to

$$\begin{aligned} \dot{\widehat{\theta}} &= -\alpha \delta e_t \\ \dot{e}_t &= -\lambda e_t + q \end{aligned} \tag{35}$$

where $\lambda \in (0, \gamma]$ is the trace decay factor [71]. By means of the discount factor, it is possible to add a stable dynamics to the trace such that it serves as a low pass filter with cutting frequency λ [47]. The eligibility trace dynamics has input-state stability (ISS) [52,73]. The above result hold for Q-function approximations. Refer to [7,52] for more details.

4.3. Experience replay

Experience replay [65,74,75] is the second memory function of the hippocampus learning system. Experience replay [68] also known as neuronal replay is one of the primary mechanisms thought to underlie the ability of the hippocampus to teach the neocortex. Indeed, neocortically and endogenously driven replay, to and within the hippocampus, may act together for the consolidation of newer memories and the protection of old ones [5].

Consider that the samples are stored in a memory vector of $\mathscr{M}$ dimension. Define q_k and ξ_k as data samples of q and ξ in a time instance k. These data are used as additional terms in the gradient method in order to evaluate multiple TD errors in the same update rule [65]. The gradient update rule is improved as

$$\dot{\widehat{\theta}} = -\alpha q \left(q^{\top} \widehat{\theta} + \xi(x,u) \right) - \alpha \sum_{k=1}^{\mathscr{M}} q_k \left(q_k^{\top} \widehat{\theta} + \xi_k(x,u) \right). \tag{36}$$

The experience replay has two main objectives: (1) add memory to the gradient update rule to enhance the performance of the neocortex learning system, and (2) reduce the exploration technique and achieve parameter convergence in less time. Indeed, it can be seen as an alternative to achieve both excitation and memory [76].

It is not difficult to prove that closed-loop system trajectories of the neocortex converges exponentially to zero. The next Lyapunov equation is considered

$$W = \frac{1}{2} \widetilde{\theta}^{\top} \alpha^{-1} \widetilde{\theta}, \tag{37}$$

where $\widetilde{\theta} = \widehat{\theta} - \theta$. The time-derivative of W along the trajectories of (36) is

$$\dot{W} = -\widetilde{\theta}^{\top} \left[q q^{\top} + \sum_{k=1}^{\mathscr{M}} q_k q_k^{\top} \right] \widetilde{\theta}. \tag{38}$$

Therefore, the trajectories of the system converges exponentially to zero under the fulfillment of the PE condition (28). Furthermore, the experience replay accelerate the convergence of the neocortex algorithm by taking into account previous TD errors in the same update rule. Refer to [65] for more details.

4.4. Imitation learning

Imitation learning [77] provides a wide number of techniques that enables the transference of knowledge from an expert to an apprentice. These techniques are very common in robotic applications such as learning by demonstrations [55,78] which uses several techniques such as Gaussian processes, dynamic time-warping [1], Lloyd's algorithm [79,80], among others.

For this review, a simpler approach is considered for the design of optimal decision making controllers. Assume that we have access to expert's trajectories $\bar{x}_e = [x_e(0) \cdots x_e((k-1)T)] \in \mathbb{R}^{n \times k}$ and $\bar{u}_e = [u_e(0) \ldots u_e((k-1)T] \in \mathbb{R}^{m \times k}$, where $x_e \in \mathbb{R}^n$ and $u_e \in \mathbb{R}^m$ denote the state and control input of the expert's trajectories in a time $t = kT$ with sampling period $T \in \mathbb{R}_+$ and $k \in \mathbb{N}$. The above data are obtained in accordance to an expert control policy $u_e = -K_e x_e$, where $K_e \in \mathbb{R}^{m \times n}$ is the expert's stabilizing control gain.

Using the above data it is possible to extract an approximate expert's control gain $\widehat{K}_e \in \mathbb{R}^{m \times n}$ as

$$\widehat{K}_e = -\bar{u}_e \bar{x}_e^{\top} \left(\bar{x}_e \bar{x}_e^{\top} \right)^{-1}. \tag{39}$$

The above control policy can be used as an initial control policy for the neocortex learning systems. However, its main advantage lies in the transference of knowledge in inverse reinforcement learning applications which is discussed in the next section.

5. Striatum learning system

The striatum learning system is responsible for evaluating different forms of information for decision making [44]. This system serves as a communication channel between the neocortex and the hippocampus learning systems. At this point it has been discussed some cognitive methods that uses the striatum to connect

the neocortex and the hippocampus, e.g., the batch LS and gradient rules, the addition of excitation signals in the control input and memory models in the update rule, the incorporation of dynamic knowledge to compute the decision making function, etc. As can be seen, it is difficult to identify the cognitive models and skills of the striatum because apparently they are part of either the neocortex or the hippocampus learning systems. However, the main difference is that the hippocampus and neocortex are only sources of information, meanwhile the striatum is the responsible to relate those information and make decisions. One interesting property of the striatum is that enables complementary learning (see Fig. 3) for experience transference [6,81]. This complementary property can be achieved by means of inverse optimal control (IOC)([82]) or inverse reinforcement learning (IRL) [83,77] which are discussed below.

5.1. Inverse Optimal Control/Inverse Reinforcement Learning

The IOC/IRL [82–84] problems seek the weights of the utility function $\xi(x,u)$ that are responsible for obtaining an optimal or expert decision making function. These methods are considered as a striatum cognitive models because they take cognitive models of both the neocortex and hippocampus for complementary learning and experience transference.

The main issue of any inverse problem is that there exists multiple solutions [85,86] that derive the same optimal decision making controller. So, one of the first main assumptions is that the weight matrix R is known in advance such that the matrix S is the only weight matrix to be estimated. Fig. 4 shows the complementary learning scheme of the IOC/IRL problems.

The scheme consists of the hippocampus model (39) that computes an estimation of the expert's control gain $\widehat{K}_e$. Assume that the weight matrix R of the expert is known in advance. The neocortex model uses any ADP/RL method and an initial weight matrix S_{i-1}, $i=1,2,\ldots$to compute the control gain K_i and kernel matrix P_i. Both K_i and P_i are solutions of the optimal control problem for a weight matrix S_{i-1}. The outputs of the hippocampus and the neocortex models feed the striatum model which gives a new improved weight matrix S_i that will be used by the neocortex model to derive a new and improved control gain K_{i+1} and kernel matrix P_{i+1}. The process continue until the weight matrix converges or satisfy an error tolerance, that is, $\|S_i - S_{i-1}\| \leqslant \varepsilon_S$ for a small constant $\varepsilon_S \in \mathbb{R}_+$.

The striatum computes a gain error $e_K \in \mathbb{R}^{m\times n}$ between the hippocampus and neocortex control gains as

$$\begin{aligned} e_K &= \widehat{K}_e - K_i, \\ &= -\bar{u}_e\bar{x}_e^\top\left(\bar{x}_e\bar{x}_e^\top\right)^{-1} - R^{-1}B^\top P_i \end{aligned} \tag{40}$$

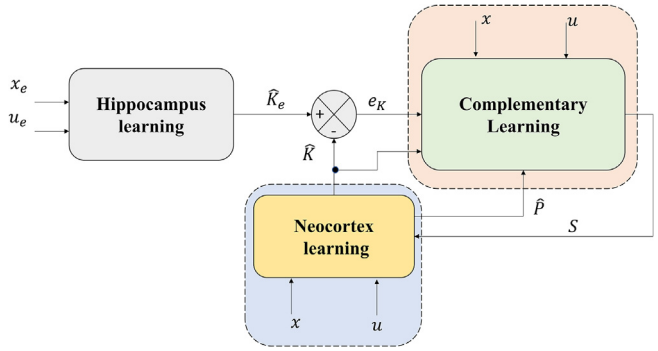

Fig. 3. Complementary learning scheme.

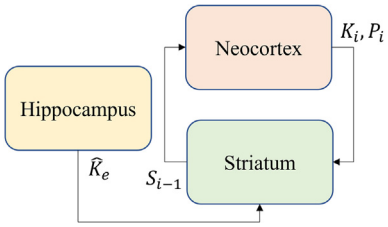

Fig. 4. Complementary Learning scheme of the IOC/IRL problems.

The main goal of the striatum is to minimize the gain error e_K. Define the cost index

$$E = \mathrm{tr}\{e_K^\top e_K\}. \tag{41}$$

Notice that the only degree of freedom (DOF) of (40) is given by the kernel matrix P_i. So, it is possible to obtain a new kernel matrix denoted by $\mathscr{P}_i \in \mathbb{R}^{n\times n}$ which minimizes (41). The best way to compute the kernel matrix $\mathscr{P}_i$ is by means of a gradient update rule because both K_i and P_i have fixed values in the striatum model. A one-step gradient rule is used as

$$\begin{aligned} \mathscr{P}_i = &\quad P_i - \beta\tfrac{\partial E}{\partial P_i} \\ = &\quad P_i + \beta\left(e_K^\top R^{-1}B^\top + BR^{-1}e_K\right), \end{aligned} \tag{42}$$

where $\beta \in \mathbb{R}_+$ is the learning rate. Notice that (42) finds a new and improved kernel matrix $\mathscr{P}_i$ which provide a new control gain which is close to the expert's control gain. Furthermore, (42) is used for the IOC problem that requires at least knowledge of matrix B. For the IRL case (42) is modified as

$$\mathscr{P}_i = P_i + \beta\left(e_K^\top K_i P_i^{-1} + P_i^{-1}K_i^\top e_K\right). \tag{43}$$

A new control gain $\mathscr{K}_i \in \mathbb{R}^{m\times n}$ can be computed from the kernel matrix $\mathscr{P}_i$ as

$$\mathscr{K}_i = R^{-1}B^\top\mathscr{P}_i, \tag{44}$$

or

$$\mathscr{K}_i = K_i P_i^{-1}\mathscr{P}_i. \tag{45}$$

Using the new kernel matrix $\mathscr{P}_i$ and the new control gain $\mathscr{K}_i$, then the new weight matrix S_i can be computed using the Bellman Eq. (27).

We can show that the weight matrix S_{i+1} converges $\lim_{i\to\infty} S_{i+1} = S_i$, by using the fact that the striatum satisfies the next ARE

$$-S_{i+1} = A^\top\mathscr{P}_i + \mathscr{P}_i A - \mathscr{P}_i BR^{-1}B^\top\mathscr{P}_i. \tag{46}$$

and also the neocortex satisfies the following ARE

$$-S_{i+1} = A^\top\widehat{P}_{i+1} + \widehat{P}_{i+1}A - \widehat{P}_{i+1}BR^{-1}B^\top\widehat{P}_{i+1}. \tag{47}$$

Substituting (47) and (42) in (46) gives

$$\begin{aligned} A^\top\widehat{P}_{i+1} + \widehat{P}_{i+1}A - \widehat{P}_{i+1}BR^{-1}B^\top\widehat{P}_{i+1} &= A^\top\widehat{P}_i + \widehat{P}_i A - \widehat{P}_i BR^{-1}B^\top\widehat{P}_i \\ &- \beta\left(A^\top\frac{\partial E_i}{\partial\widehat{P}_i} + \frac{\partial E_i}{\partial\widehat{P}_i}A\right) + \beta\left(\widehat{P}_i BR^{-1}B^\top\frac{\partial E_i}{\partial\widehat{P}_i} + \frac{\partial E_i}{\partial\widehat{P}_i}BR^{-1}B^\top\widehat{P}_i\right) \\ &- \beta^2\frac{\partial E_i}{\partial\widehat{P}_i}BR^{-1}B^\top\frac{\partial E_i}{\partial\widehat{P}_i}. \end{aligned} \tag{48}$$

The gradient descent rule (42) updates the kernel matrix $\mathscr{P}$ in each iteration i such that the error e_K is minimized, that is, $\widehat{K} \to \widehat{K}_e$, then $\lim_{i\to\infty}\frac{\partial E_i}{\partial\widehat{P}_i} = 0$ and $\lim_{i\to\infty}\mathscr{P}_i = \widehat{P}_i$ and hence $\lim_{i\to\infty} S_{i+1} = S_i$. Refer to [81] for more details.

Remark 2. The weight matrix S_i in most cases will not converge to the expert's weight matrix due to lack of constraints. However, the optimal or near optimal performance is guaranteed.

5.2. Optimization Techniques

The striatum possesses an important skill related to optimization of the variables that are independent to the action set $\mathscr{A}$, for example, hyperparameters [38] (learning rate, discount factor, number of basis functions, number of iterations, weight matrices, excitation frequencies, error tolerances, etc.).

The IOC/IRL exhibits an optimization procedure of the weight matrix S such that the same optimal behavior is obtained. However, there exists multiple solutions that derive the same performance and we require to add some constraints. The most common approaches to solve optimization problems under constraints use a convex optimization algorithm [84] based on linear or quadratic programming methods [83,50] to obtain the optimal variable or variables of interest.

Other techniques seeks to optimize the number of operations that any algorithm possess, for example, zeroing neural networks [87] which has shown interesting results in optimal control applications by formulating a set of equations and identify the number of zeros them.

Nevertheless, the variables of interest are usually hand-crafted and the optimization procedure has been put aside. This fact gives a strong area for future research that may improve the human-behavior learning approach.

6. An illustrative example

In the following example is exemplified the use of a human-behavior algorithm and the relations with different cognitive models of the neocortex and the hippocampus via the striatum.

The dynamics of a DC servomechanism [2,88] is described by the next matrices

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -0.1801 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 139.5 \end{bmatrix}. \tag{49}$$

The action set $\mathscr{A}$ is defined by the input state $x = [x_1, x_2]^\top$ and the control input voltage u, where x_1 and x_2 denote the position and velocity of the DC servomechanism. The main control objective is to stabilize the states trajectories x at the origin. The weights of the utility function were set to $S = I_2$ and $R = 150$. The neocortex was designed to fulfil the gradient rule (20). The hippocampus had two cognitive models: an exponential decaying signal and experience replay memory (36). The striatum was modeled by the addition of the hippocampus cognitive models in the Q-learning update rule. The learning rate was set to $\alpha = 5$ which exhibited the best estimation results and stable performance, and the memory replay was set to $\mathscr{M} = 5$.

The optimal M matrix of the Q-function (22) was

$$M = \begin{bmatrix} 2.0844 & 1.1563 & 12.2474 \\ 1.1563 & 1.2356 & 13.087 \\ 12.2474 & 13.087 & 150 \end{bmatrix}.$$

Fig. 5The performance of the Q-learning algorithm was tested with ($\mathscr{M} = 5$) and without ($\mathscr{M} = 0$) the experience replay cognitive model. The algorithm estimates five parameters. The estimate $\widehat{Q}_{uu}$ was not estimated since R is known in advance. The set of basis functions were

$$\psi = \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 & x_1u & x_2u \end{bmatrix}^\top.$$

Fig. 2 shows the position results of the servomechanism shaft. The plot shows that the incorporation of the memory cognitive

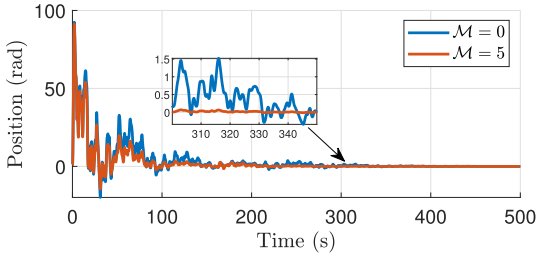

Fig. 5. Position x_1 of the DC Servomechanism.

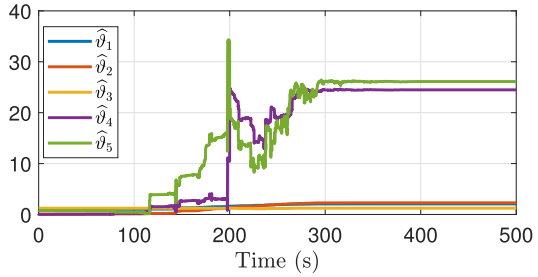

Fig. 6. Parameter estimates $\widehat{\vartheta}$ convergence.

model accelerates the stabilization of the position shaft at the origin by adding previous TD errors in the same update rule. The time evolution of the estimates $\widehat{\vartheta}$ are given in Fig. 6.

The estimates converge to the following value

$$\widehat{M} = \begin{bmatrix} \widehat{\vartheta}_1 & \frac{1}{2}\widehat{\vartheta}_2 & \frac{1}{2}\widehat{\vartheta}_4 \\ \frac{1}{2}\widehat{\vartheta}_2 & \widehat{\vartheta}_3 & \frac{1}{2}\widehat{\vartheta}_5 \\ \frac{1}{2}\widehat{\vartheta}_4 & \frac{1}{2}\widehat{\vartheta}_5 & R \end{bmatrix} = \begin{bmatrix} 2.0739 & 1.1459 & 12.2467 \\ 1.1459 & 1.2354 & 13.0585 \\ 12.2467 & 13.0585 & 150 \end{bmatrix}.$$

The Euclidean norm of the estimation error $\widetilde{M} = \widehat{M} - M$ was used as performance metric of the accuracy of the human-behavior algorithm. The result was $\|\widehat{M} - M\| = 0.0343$ which shows that near optimal estimates were obtained which can be improved using optimization techniques in: the learning rate, the memory size, the frequencies of the excitation signal, and the gradient descent rule.

7. Conclusions

This paper provides a brief overview of different techniques and methods that involve a new approach named as human-behavior learning for the design of optimal decision making controllers. This technique is modeled by three main sets: actions, cognitions, and emotions. The actions set constitute all sensory inputs such as states and controls. The emotions set is designed and constrained to be quadratic in terms of actions set to facilitate the optimal control design. The cognitive set is described by three main learning systems: the neocortex, the hippocampus, and the striatum which have complementary learning properties that achieve adaptability and optimal performances.

The neocortex is described as a pattern dependent and slow-learning system that acquire new information by interacting with the environment, such as ADP and RL algorithms. The hippocampus models a pattern-independent and fast-learning system that provides previous experiences and memories to the neocortex to enhance its performance, such as excitation signals, and replay memory. The striatum offers a communication channel between the neocortex and the hippocampus such that multiple sources of information are exploited to improve learning and the final deci-

sion making function, such as gradient update rules and optimization techniques.

Simulation studies show the advantages of the human-behavior approach by exploiting the complementary learning properties of the hippocampus, neocortex, and striatum. Further work, includes the adequate design and explainability of the emotions set to obtain realistic information that enhance the human-behavior algorithm. Furthermore, optimization techniques are an open problem in the context of optimal control theory and their applications.

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] J. Ramírez, W. Yu, Human behavior learning in joint space using dynamic time warping and neural networks, in: 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), IEEE, 2019, pp. 1–6, https://doi.org/10.1109/ICEEE.2019.8884501.

[2] A. Perrusquía, W. Yu, X. Li, Nonlinear control using human behavior learning, Information Sciences 569 (2021) 358–375, https://doi.org/10.1016/j.ins.2021.03.043.

[3] B.M. Lake, T.D. Ullman, J.B. Tenenbaum, S.J. Gershman, Building machines that learn and think like people, Behavioral and brain sciences 40. doi:10.1017/S0140525X16001837.

[4] H. Qiao, J. Chen, X. Huang, A survey of brain-inspired intelligent robots: Integration of vision, decision, motion control, and musculoskeletal systems, IEEE Transactions on Cybernetics.

[5] R.C. O'Reilly, R. Bhattacharyya, M.D. Howard, N. Ketz, Complementary learning systems, Cognitive science 38 (6) (2014) 1229–1248, https://doi.org/10.1111/j.1551-6709.2011.01214.x.

[6] S. Blakeman, D. Mareschal, A complementary learning systems approach to temporal difference learning, Neural Networks 122 (2020) 218–230, https://doi.org/10.1016/j.neunet.2019.10.011.

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

[8] X. Huang, W. Wu, H. Qiao, Computational modeling of emotion-motivated decisions for continuous control of mobile robots, IEEE Transactions on Cognitive and Developmental Systems 13 (1) (2020) 31–44.

[9] X. Huang, W. Wu, H. Qiao, Connecting model-based and model-free control with emotion modulation in learning systems, IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[10] T.M. Moerland, J. Broekens, C.M. Jonker, Emotion in reinforcement learning agents and robots: a survey, Machine Learning 107 (2) (2018) 443–480.

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

[12] C.-Y. Lee, J.-J. Lee, Adaptive control for uncertain nonlinear systems based on multiple neural networks, IEEE Transactions on Systems Man and Cybernetics Part B 34 (1) (2004) 325–333, https://doi.org/10.1109/TSMCB.2003.811520.

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

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

[15] I. Grondman, L. Buşoniu, R. Babuška, Model learning actor-critic algorithms: Performance evaluation in a motion control task, in: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), IEEE, 2012, pp. 5272–5277, https://doi.org/10.1109/CDC.2012.6426427.

[16] A. Perrusquía, W. Yu, X. Li, Redundant robot control using multi agent reinforcement learning, in: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), 2020, pp. 1650–1655, https://doi.org/10.1109/CASE48305.2020.9216774.

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

[18] Y. Jiang, J. Fan, T. Chai, J. Li, F.L. Lewis, Data-driven flotation industrial process operational optimal control based on reinforcement learning, IEEE Transactions on Industrial Informatics 14 (5) (2017) 1974–1989, https://doi.org/10.1109/TII.2017.2761852.

[19] S. Batilotti, A. De Santis, Robust output feedback control of nonlinear stochastic systems using neural networks, IEEE Transactions Neural Networks 14 (1) (2003) 103–116, https://doi.org/10.1109/TNN.2002.806609.

[20] H. Modares, F.L. Lewis, 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 26 (10) (2015) 2550–2562, https://doi.org/10.1109/TNNLS.2015.2441749.

[21] A. Perrusquía, W. Yu, A. Soria, Large space dimension reinforcement learning for robot position/force discrete control, in: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), 2019, pp. 91–96, https://doi.org/10.1109/CoDIT.2019.8820575.

[22] C. Wang, Y. Li, S. Sam Ge, T. Heng Lee, Optimal critic learning for robot control in time-varying environments, IEEE Transactions on Neural Networks and Learning Systems 26 (10) (2015) 2301–2310. doi:10.1109/TNNLS.2014.2378812.

[23] M.G. Mattar, N.D. Daw, Prioritized memory access explains planning and hippocampal replay, Nature neuroscience 21 (11) (2018) 1609–1617, https://doi.org/10.1038/s41593-018-0232-z.

[24] K.L. Stachenfeld, M.M. Botvinick, S.J. Gershman, The hippocampus as a predictive map, Nature neuroscience 20 (11) (2017) 1643–1653, https://doi.org/10.1038/nn.4650.

[25] H.F. Ólafsdóttir, D. Bush, C. Barry, The role of hippocampal replay in memory and planning, Current Biology 28 (1) (2018) R37–R50, https://doi.org/10.1016/j.cub.2017.10.073.

[26] 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 37 (28) (2017) 6686–6697, https://doi.org/10.1523/JNEUROSCI.3212-16.2017.

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

[28] A. Perrusquía, W. Yu, A. Soria, Position/force control of robot manipulators using reinforcement learning, Industrial Robot: the international jounral of robotics research and application 46 (2) (2019) 267–280, https://doi.org/10.1108/IR-10-2018-0209.

[29] I. Grondman, L. Buşoniu, G.A. Lopes, R. Babŭska, A survey of actor-critic reinforcement learning: standard and natural policy gradients, IEEE Transactions on Systems, Man, and Cybernetics, PART C 42 (6) (2012) 1291–1307, https://doi.org/10.1109/TSMCC.2012.2218595.

[30] R. Kamalapurkar, P. Walters, W. Dixon, Model-based reinforcement learning for approximate optimal regulation, Automatica 64 (2016) 94–104, https://doi.org/10.1016/j.automatica.2015.10.039.

[31] H. Modares, F.L. Lewis, Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning, IEEE Transactions on Automatic Control 59 (11) (2014) 3051–3056, https://doi.org/10.1109/TAC.2014.2317301.

[32] D. Vrabie, F.L. Lewis, Neural networks approach for continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks 22 (2009) 237–246, https://doi.org/10.1016/j.neunet.2009.03.008.

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

[34] A. Perrusquía, W. Yu, X. Li, Multi-agent reinforcement learning for redundant robot control in task-space, International Journal of Machine Learning and Cybernetics 12 (1) (2021) 231–241, https://doi.org/10.1007/s13042-020-01167-7.

[35] B. Kiumarsi, F.L. Lewis, Actor-critic based optimal tracking for partially unknown nonlinear discrete- time systems, IEEE Transactions on Neural Networks and Learning Systems 26 (1) (2015) 140–151, https://doi.org/10.1109/TNNLS.2014.2358227.

[36] B. Kiumarsi, F.L. Lewis, H. Modares, A. Karimpor, M.-B. Naghibi-Sistani, Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics, Automatica 50 (2014) 1167–1175, https://doi.org/10.1016/j.automatica.2014.02.015.

[37] B. Kiumarsi, K.G. Vamvoudakis, H. Modares, F.L. Lewis, Optimal and autonomous control using reinforcement learning: a survey, IEEE Transactions on Neural Networks and Learning Systems 29 (6) (2018) 2042–2062, https://doi.org/10.1109/TNNLS.2017.2773458.

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

[39] G. Cybenko, Approximation by superposition of sigmoidal activation function, Math, Control, Sig. Syst. 2 (1989) 303–314, https://doi.org/10.1007/BF02551274.

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

[41] A. Perrusquía, W. Yu, Neural $\mathscr{H}_2$ control using reinforcement learning for unknown nonlinear systems, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–7. doi:10.1109/IJCNN48605.2020.9207547.

[42] L. Weng, Exploration strategies in deep reinforcement learning, lilianweng.github.io/lil-log. url:https://lilianweng.github.io/lil-log/2020/06/07/exploration-strategies-in-deep-reinforcement-learning.html.

[43] A. Al-Tamimi, F. Lewis, M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof, IEEE Transactions on System, Man, and Cybernetics Part B, Cybernetics 38 (4) (2008) 943–949, https://doi.org/10.1109/ADPRL.2007.368167.

[44] W. Schultz, P. Apicella, E. Scarnati, T. Ljungberg, Neuronal activity in monkey ventral striatum related to the expectation of reward, Journal of neuroscience 12 (12) (1992) 4595–4610, https://doi.org/10.1523/JNEUROSCI.12-12-04595.1992.

[45] D. Kumaran, D. Hassabis, J.L. McClelland, What learning systems do intelligent agents need? complementary learning systems theory updated, Trends in

cognitive sciences 20 (7) (2016) 512–534, https://doi.org/10.1016/j.tics.2016.05.004.
[46] J.L. McClelland, B.L. McNaughton, R.C. O'Reilly, Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory, Psychological review 102 (3) (1995) 419, https://doi.org/10.1037/0033-295X.102.3.419.
[47] C.-T. Chen, Linear System Theory and Design, Oxford University Press, 1999.
[48] S.A.A. Rizvi, Z. Lin, Output feedback Q-learning control for the discrete-time linear quadratic regulator problem, IEEE transactions on neural networks and learning systems 30 (5) (2018) 1523–1536, https://doi.org/10.1109/TNNLS.2018.2870075.
[49] A. Perrusquía, W. Yu, Neural $\mathscr{H}_2$ control using continuous-time reinforcement learning, IEEE Transactions on Cybernetics (2020) 1–10, https://doi.org/10.1109/TCYB.2020.3028988.
[50] A. Perrusquía, W. Yu, Robust control under worst-case uncertainty for unknown nonlinear systems using modified reinforcement learning, International Journal of Robust and Nonlinear Control 30 (7) (2020) 2920–2936, https://doi.org/10.1002/rnc.4911.
[51] A. Gheibi, A. Ghiasi, S. Ghaemi, M. Badamchizadeh, Designing of robust adaptive passivity-based controller based on reinforcement learning for nonlinear port-Hamiltonian model with disturbance, International Journal of Control 93 (8) (2020) 1754–1764, https://doi.org/10.1080/00207179.2018.1532607.
[52] A. Perrusquía, W. Yu, Robot position/force control in unknown environment using hybrid reinforcement learning, Cybernetics and Systems 51 (4) (2020) 542–560, https://doi.org/10.1080/01969722.2020.1758466.
[53] K. Vamvoudakis, F.L. Lewis, On-line actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010) 878–888, https://doi.org/10.1016/j.automatica.2010.02.018.
[54] K.G. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, Systems & Control Letters (2017) 14–20, https://doi.org/10.1016/j.sysconle.2016.12.003.
[55] C.G. Atkeson, S. Schaal, Robot learning from demonstration, in: ICML, Vol. 97, Citeseer, 1997, pp. 12–20. doi:10.1007/springerreference_302487.
[56] J. Martín H., J. de Lope, D. Maravall, Robust high performance reinforcement learning through weighted k-nearest neighbors, Neurocomputing 74 (2011) 1251–1259. doi:10.1016/j.neucom.2010.07.027.
[57] W. Yu, X. Li, Fuzzy identification using fuzzy neural networks with stable learning algorithms, IEEE Transactions on Fuzzy Systems 12 (3) (2004) 411–420, https://doi.org/10.1109/TFUZZ.2004.825067.
[58] D. Luviano, W. Yu, Continuous-time path planning for multi-agents with fuzzy reinforcement learning, Journal of Intelligent & Fuzzy Systems 33 (2017) 491–501, https://doi.org/10.3233/JIFS-161822.
[59] H. Zhang, L. Cui, X. Zhang, Y. Luo, Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method, IEEE Transactions on Neural Networks 22 (12) (2011) 2226–2236, https://doi.org/10.1109/TNN.2011.2168538.
[60] S.J. Chinta, J.K. Andersen, Dopaminergic neurons, The international journal of biochemistry & cell biology 37 (5) (2005) 942–946.
[61] S. Lammel, B.K. Lim, R.C. Malenka, Reward and aversion in a heterogeneous midbrain dopamine system, Neuropharmacology 76 (2014) 351–359.
[62] Q. Xie, B. Luo, F. Tan, Discrete-time lqr optimal tracking control problems using approximate dynamic programming algorithm with disturbance, in: 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), IEEE, 2013, pp. 716–721, https://doi.org/10.1109/ICICIP.2013.6568166.
[63] M. Palanisamy, H. Modares, F.L. Lewis, M. Aurangzeb, Continuous-time Q-learning for infinite horizon-discounted cost linear quadratic regulator problems, IEEE Transactions on Cybernetics 45 (2) (2015) 165–176, https://doi.org/10.1109/TCYB.2014.2322116.
[64] A. Perrusquía, W. Yu, Continuous-time reinforcement learning for robust control under worst-case uncertainty, International Journal of Systems Science 52 (4) (2021) 770–784, https://doi.org/10.1080/00207721.2020.1839142.
[65] H. Modares, F.L. Lewis, M.-B. Naghibi-Sistani, Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, Automatica 50 (1) (2014) 193–202, https://doi.org/10.1016/j.automatica.2013.09.043.
[66] M. Sarmad, H.J. Lee, Y.M. Kim, Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5898–5907. doi:10.1109/CVPR.2019.00605.
[67] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, nature 518 (7540) (2015) 529–533. doi:10.1038/nature14236.
[68] A. Taleb Zadeh Kasgari, W. Saad, M. Mozaffari, H.V. Poor, Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication, arXiv e-prints (2019) arXiv–1911 doi:10.1109/TCOMM.2020.3031930.
[69] J. Young Lee, J.B. Park, 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 26 (5). doi:10.1109/TNNLS.2014.2328590.
[70] A. Perrusquía, W. Yu, Discrete-time $\mathscr{H}_2$ neural control using reinforcement learning, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–11, https://doi.org/10.1109/TNNLS.2020.3026010.
[71] K. Doya, Reinforcement learning in continuous time and space, Neural computation 12 (1) (2000) 219–245, https://doi.org/10.1162/089976600300015961.
[72] S. Singh, S. Sutton, Richard, Reinforcement learning with replacing elegibility traces, Machine learning 22 (1996) 123–158, https://doi.org/10.1023/A:1018012322525.
[73] H. Khalil, Nonlinear systems, Prentice Hall, 2002.
[74] D. Zha, K.-H. Lai, K. Zhou, X. Hu, Experience replay optimization, Proceedingds of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19) doi:arXiv:1906.08387.
[75] S. Adam, L. Buşoniu, R. Babuka, Experience replay for real-time reinforcement leaening control, IEEE Transactions on Systems, Man and Cybernetics 42 (2) (2012) 201–212. doi:10.1109/TSMCC.2011.2106494.
[76] R. Liu, J. Zou, The effects of memory replay in reinforcement learning, International Conference on Machine Learning 2017 Workshop on Principled Approaches to Deep Learning doi:10.1109/ALLERTON.2018.8636075.
[77] J. Ramírez, W. Yu, A. Perrusquía, Model-free reinforcement learning from expert demonstrations: a survey, Artificial Intelligence Review (2021) 1–29.
[78] E. Pignat, J. Silvério, S. Calinon, Learning from demonstration using products of experts: applications to manipulation and task prioritization, arXiv preprint arXiv:2010.03505 doi:10.1177/ToBeAssigne.
[79] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and autonomous systems 57 (5) (2009) 469–483, https://doi.org/10.1016/j.robot.2008.10.024.
[80] H. Ravichandar, A.S. Polydoros, S. Chernova, A. Billard, Recent advances in robot learning from demonstration, Annual Review of Control, Robotics, and Autonomous Systems 3. doi:10.1146/annurev-control-100819-063206.
[81] A. Perrusquía, A complementary learning approach for expertise transference of human-optimized controllers, Neural Networks 145 (2022) 33–41.
[82] N. Ab Aza, A. Shahmansoorian, M. Davoudi, From inverse optimal control to inverse reinforcement learning: A historical review, Annual Reviews in Control 50 (2020) 119–138, https://doi.org/10.1016/j.arcontrol.2020.06.001.
[83] A.Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 663–670. doi:10.2460/AJVR.67.2.323.
[84] H. El-Hussieny, J.-H. Ryu, Inverse discounted-based LQR algorithm for learning human movement behaviors, Applied Intelligence 49 (4) (2019) 1489–1501, https://doi.org/10.1007/s10489-018-1331-y.
[85] W. Yu, A. Perrusquía, Simplified stable admittance control using end-effector orientations, International Journal of Social Robotics 12 (5) (2020) 1061–1073, https://doi.org/10.1007/s12369-019-00579-y.
[86] A. Perrusquía, W. Yu, Human-in-the-loop control using Euler angles, Journal of Intelligent & Robotic Systems 97 (2020) 271–285, https://doi.org/10.1007/s10846-019-01058-2.
[87] L. Xiao, K. Li, M. Duan, Computing time-varying quadratic optimization with finite-time convergence and noise tolerance: a unified framework for zeroing neural network, IEEE Transactions on Neural Networks and Learning Systems 30 (11) (2019) 3360–3369, https://doi.org/10.1109/TNNLS.2019.2891252.
[88] R. Garrido, R. Miranda, DC servomechanism parameter identification: A closed loop input error approach, ISA Transactions 51 (2012) 42–49, https://doi.org/10.1016/j.isatra.2011.07.003.

Adolfo Perrusquía received the B.Eng. degree in Mechatronic Engineering from the National Polytechnic Institute (UPIITA-IPN) in 2014, and the M.Sc. 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 in Cranfield University and a UK-IC 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.

https://dspace.org/

Human-behavior learning: a new complementary learning perspective for optimal decision making controllers

Perrusquía, Adolfo

2022-03-17
Elsevier
https://doi.org/10.1016/j.neucom.2022.03.036
Perrusquía A. (2022) Human-behavior learning: a new complementary learning perspective for optimal decision making controllers, Neurocomputing, Volume 489, June 2022, pp. 157-166

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