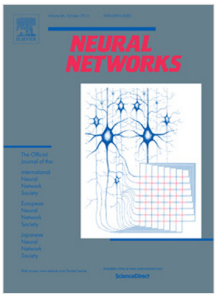
Journal Pre-proof



A complementary learning approach for expertise transference of human-optimized controllers

Adolfo Perrusquía

PII: S0893-6080(21)00400-7
DOI: https://doi.org/10.1016/j.neunet.2021.10.009
Reference: NN 5007

To appear in: *Neural Networks*

Received date : 20 June 2021
Revised date : 24 August 2021
Accepted date : 11 October 2021

Please cite this article as: A. Perrusquía, A complementary learning approach for expertise transference of human-optimized controllers. *Neural Networks* (2021), doi: https://doi.org/10.1016/j.neunet.2021.10.009.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Elsevier Ltd. All rights reserved.

Journal Pre-proof

Neural Networks, Volume 145, January 2022, pp. 33-41
DOI:10.1016/j.neunet.2021.10.009

A Complementary Learning Approach for Expertise Transference of Human-Optimized Controllers

Adolfo Perrusquía*

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

Abstract

In this paper, a complementary learning scheme for experience transference of unknown continuous-time linear systems is proposed. The algorithm is inspired in the complementary learning properties that exhibit the hippocampus and neocortex learning systems via the striatum. The hippocampus is modelled as pattern-separated data of a human optimized controller. The neocortex is modelled as a Q-reinforcement learning algorithm which improves the hippocampus control policy. The complementary learning (striatum) is designed as an inverse reinforcement learning algorithm which relates the hippocampus and neocortex learning models to seek and transfer the weights of the hidden expert's utility function. Convergence of the proposed approach is analysed using Lyapunov recursions. Simulations are given to verify the proposed approach.

Keywords: complementary learning, hippocampus and neocortex learning systems, Q-learning, inverse reinforcement learning, batch least squares, gradient-descent rule.

1. Introduction

Control of linear systems has been studied in the last decades with satisfactory results using either model-based or model-free controllers [1]. There are various techniques for control design and gain tuning [2, 3, 4]. However,

*Corresponding author
Email address: Adolfo.Perrusquia@cranfield.ac.uk (Adolfo Perrusquía)

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0).
The final published version (version of record) is available online at DOI:10.1016/j.neunet.2021.10.009. Please refer to any applicable publisher terms of use.

Journal Pre-proof

human intervention is required in the control implementation to guarantee a desired performance due to the presence of disturbances and restrictions that the human imposes, for example, error tolerances, control input bounds, output responses, etc.

Intelligent techniques have been used as expert systems to emulate the human expertise such as fuzzy systems [5], neural networks [6, 7], imitation learning (IL) [8, 9] and recently reinforcement learning (RL) [10, 11]. Fuzzy systems are used to model the gain tuning of linear controllers, for example, PD or PID control laws. Neural networks are used as a compensation of the system dynamics to reduce the steady state error of the closed-loop system and improve the control precision. Nevertheless, the performance of both fuzzy systems and neural networks is not optimal and lacks robustness, so intervention of the human is required.

Learning from Demonstration (LfD) [12, 13] or Programming by Demonstration (PbD) [14] is an IL algorithm that have been used in manufacturing robotics for robot movements programming [15, 16]. In an IL algorithm, a subject called apprentice learns to imitate a teacher's behavior when carrying out a task. On the other hand, the teacher or expert provides demonstrations of the desired behavior by performing the task. It is important to note that in IL, the teacher's behavior is assumed to be optimal, which may not be entirely accurate, since the teacher, who can be a human, naturally makes mistakes and might follow a lousy strategy when demonstrating a task, so the behavior learned by the apprentice is directly affected, thus being the teacher's behavior suboptimal. Notice that the trajectories of the demonstrations are not the same (in terms of velocities, shape, time) and hence, it is difficult or impossible to extract/infer the teacher's cost index. Furthermore, the apprentice learns the features of the trajectory but not the hidden intention of the teacher.

RL has become very popular in recent years due to the capability of emulating how human adjust their control actions based on stimuli received in response to the applied actions [17]. In control sense, this stimuli is an utility function (also known as reward) which we want to minimize or maximize in an

Journal Pre-proof

infinite horizon pre-defined cost index (also known as value function) [18, 19]. RL merges the capabilities of optimal control and adaptive control; hence two approaches can be distinguished: model-based and model-free. Model-based solves online a Hamilton-Jacobi-Bellman equation [20] to obtain the optimal control policy [21]; for example, the linear quadratic regulator (LQR) control [22, 23, 24]. Model-free schemes find online the optimal control policy using state and control measurements along the system trajectories; for example, Q-learning [25, 23] and actor-critic methods [26, 27, 28].

In our recent work [29, 30], it has been demonstrated that RL can obtain an optimal and robust performance [31] in presence of the worst-case uncertainty by modifying the optimal control problem into an optimization problem under restrictions. This modification is given at the utility function because is the most succinct, robust, and transferable definition of the task [32]; then the final control policy is robust against perturbations that are equal to or less than the worst-case uncertainty.

The design of the utility function is not a trivial problem and depends on the system and control goal. One approach that seeks to obtain the utility function is the inverse reinforcement learning (IRL) [33]. IRL finds the utility function from an expert control policy (provided by other controller, intelligent technique or the human) using a linear parameterization in terms of the states and actions (similarly to IL). The most famous IRL approaches solve a linear programming (LP) or quadratic programming (QP) problems [32] to obtain the adequate weights that fits the features of the utility function parameterization. However, a binary utility function is required to guarantee boundedness of the parameterization. Hence the basis functions are bounded between zero and one. In this sense, the main features of expert controllers (manual tuning controllers, fuzzy or neural based controllers, etc.) can be extracted and used to improve the performance and robustness of the control policy using any reinforcement learning technique [34].

Recently, a new approach called human-behavior learning was proposed [35] as a generalization of the humans decision-making process. This approach is

Journal Pre-proof

inspired in how human learns in accordance to the interaction of experiences, cognitions, and emotions [36]. This interaction can be related to three main learning systems of the brain cortex: the hippocampus, the neocortex, and the striatum. The hippocampus is responsible for providing memory and experiences [37, 38] to the brain cortex such that the human is capable to develop a certain task effectively. On the other hand, the neocortex is responsible for learning by interaction (human-environment) in order to relate, in well distributed representations [39, 40], the acquired knowledge. In this system the encoding of temporal difference (TD) error between dopaminergic neurons [39, 41] is realized (reinforcement learning). The striatum [42] relates the information provided by the hippocampus and the neocortex for decision making.

The hippocampus, neocortex, and the striatum are co-dependent and have different functionalities, that is, whilst the hippocampus is a fast and pattern-separated learning system [43] that teaches the neocortex the best way to execute a desired task [44], the neocortex is a slow-learning pattern-dependent system which updates in small rates [45] its connections and store the acquired knowledge in the hippocampus as memory/previous experience. Notice that the hippocampus and the neocortex cannot make decisions. Moreover, their co-dependence is given by the striatum, in other words, the striatum relates the information of both the hippocampus and the neocortex to obtain a complementary learning (CL)[46] that exploits fast and slow learning structures to achieve complex behaviours. In control context, CL serves as a connecting tool for the transference of experiences to an online learning algorithm. For sake of simplicity, the striatum will be denoted as the complementary learning system.

In this paper, a CL algorithm is proposed to solve the optimal control problem of unknown continuous-time linear systems. The CL algorithm is designed as an IRL problem which uses a relation between the hippocampus and neocortex models to extract the weights of the expert's utility function. Whilst the hippocampus is modelled as independent expert's data, the neocortex is designed as a Q-learning algorithm. The stability and convergence of the proposed approach is proven using Lyapunov recursions. Simulation studies were carried

Journal Pre-proof

out to verify the proposed approach.

The main contributions of this paper are: i) a model-free complementary learning algorithm for experience transference of unknown linear systems, ii) a new learning perspective based on the complementary functionalities of the hippocampus, neocortex, and the striatum applied in a control perspective, iii) the expert's control policy and utility function are improved to achieve the same performance with low control input values, iv) rigorous theoretical proofs and convergence of each learning system are analysed.

1.1. Problem Formulation

The next linear system is considered

$$\dot{x} = Ax + Bu, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the state and input matrices which are assumed to be unknown, $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control input. Assume the pair (A, B) is controllable.

The main concern of this paper is to design a feedback control law which achieves a desired optimal performance using two sources of knowledge: expert's data and on-line interaction data. This problem is immediately related to an optimization problem, where a wide variety of architectures have been developed such as the presented in [47, 48, 49]. However, in this work a new interesting perspective is used to solve the above control issue. The perspective consists in modelling the control problem as a human-behavior learning problem [35] that exploits the complementary learning properties of the hippocampus and the neocortex [39]. That is, whilst the hippocampus relies in fast learning pattern-separated information/memories (expert's data), the neocortex uses slow-learning pattern-dependent structures to generalize experiences (on-line interaction data).

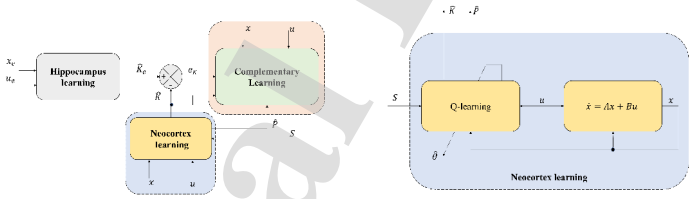
The objective of the expert's data is to provide information/examples of a good (not necessarily optimal) performance of the closed-loop system trajectories under an expert controller. This controller was designed and tuned by the

Journal Pre-proof

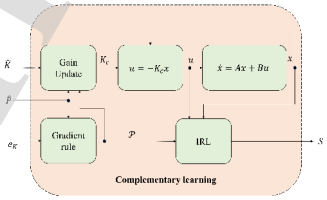
expert in accordance to a desired performance [2], e.g., minimize the tracking error, control input, overshoots, time, etc. The desired performances is constrained by an unknown utility function which is the most succinct, robust, and transferable definition of the task. Hence, it is essential to transfer the utility function of the expert's data (hippocampus) to an on-line learning algorithm (neocortex) to achieve the desired performance and improve it if possible.

Therefore, instead of solving a standard optimal control problem, we aim to solve an inverse optimal control (IOC) problem [50, 51, 52] and use the complementary learning qualities of the hippocampus and the neocortex learning models.

The complete block diagrams of the proposed approach is shown in Figure 1. The expertise learning scheme (see Figure 1(a)) relates the hippocampus and neocortex learning models via their complementary learning properties. Both the neocortex and complementary learning models learn their respective outputs in an iterative fashion as shown in Figure 1(b) and Figure 1(c).



(a) Expertise learning scheme (b) Neocortex learning scheme



(c) Complementary learning scheme

Figure 1: Block diagrams of the proposed approach

Throughout this paper, $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}^+$, $\mathbb{R}^n$, $\mathbb{R}^{n\times m}$ denote the spaces of

Journal Pre-proof

natural numbers, real numbers, positive real numbers, positive integers, real n-vectors, and real $n\times m$-matrices, respectively; $\text{vech}(A) \in \mathbb{R}^{\frac{1}{2}n(n+1)}$ and $\text{vec}(A) \in \mathbb{R}^{n^2}$ denote a half-vectorization and the complete vectorization of matrix A ; $\otimes$ and $\bar{\otimes}$ denote the symmetric and the complete Kronecker products; where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$ and $n, m \in \mathbb{N}$.

2. Hippocampus learning

Consider that we are provided with expert's data $\bar{x}_e = [x_e(0) \cdots x_e((k-1)T)] \in \mathbb{R}^{n\times k}$ and $\bar{u}_e = [u_e(0) \cdots u_e((k-1)T)] \in \mathbb{R}^{m\times k}$, 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 $T > 0$ and $k \in \mathbb{Z}^+$. These data can only be used for experiments with same conditions, for example, same time duration or initial conditions. Otherwise, the data cannot be used and will require intervention of the user to manipulate the data. The above data satisfy the dynamical equation

$$\dot{x}_e = Ax_e + Bu_e. \tag{2}$$

These data are the result of the minimization of the next value function $V_e(x_e) : \mathbb{R}^n \to \mathbb{R}$

$$V_e(x_e) = \int_t^\infty \rho_e(x_e, u_e)d\tau \tag{3}$$

where $\rho_e(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the expert's utility function [35]. In this paper, the main goal of the expert is to minimize a quadratic utility function in terms of the state and control input, that is, $\rho_e(x_e, u_e) = x_e^\top S_e x_e + u_e^\top R_e u_e$, where $S_e = S_e^\top \geq 0 \in \mathbb{R}^{n\times n}$ and $R_e = R_e^\top > 0 \in \mathbb{R}^{m\times m}$ are the weights of the expert utility function which are unknown. The optimal value function $V_e^*(x_e)$ can be written quadratic in the state as

$$V_e^*(x_e) = x_e^\top P_e x_e, \tag{4}$$

for some kernel matrix $P_e = P_e^\top > 0 \in \mathbb{R}^{n\times n}$. The Hamiltonian associated to (3) with respect to (2) and (4) is

$$H_e(x_e, u_e) = 2x_e^\top P_e(Ax_e + Bu_e) + x_e^\top S_e x_e + u_e^\top R_e u_e = 0. \tag{5}$$

Journal Pre-proof

The expert designs and tunes its control policy $u_e = -K_e x_e$ (for some stabilizing gain $K_e \in \mathbb{R}^{m\times n}$) such that (2) is optimized and fulfils the stationary condition $\frac{\partial H_e(x_e,u_e)}{\partial u_e} = 0$ [53]. Hence, the expert's optimal controller is

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

The application of (6) in the Hamiltonian (5) gives the next Algebraic Riccati equation (ARE) [23]

$$A^\top P_e + P_e A - P_e B R_e^{-1} B^\top P_e + S_e = 0. \tag{7}$$

As was previously mentioned, we do not have access to the control gain K_e, the kernel matrix P_e, and neither the weight matrices S_e and R_e. However, it is possible to derive an approximate control gain $\widehat{K}_e$ by means of a right least-squares (LS) algorithm as

$$\begin{aligned} \bar{u}_e &= -\widehat{K}_e \bar{x}_e \\ \bar{u}_e \bar{x}_e^\top &= -\widehat{K}_e \bar{x}_e \bar{x}_e^\top \\ \widehat{K}_e &= -\bar{u}_e \bar{x}_e^\top (\bar{x}_e \bar{x}_e^\top)^{-1}. \end{aligned} \tag{8}$$

The LS solution gives an approximate gain matrix $\widehat{K}_e$ because it depends on the amount of available data and the measurement noise. In the following sections the expertise learning algorithm is developed to discover the hidden weights S_e and R_e of the utility function.

3. Neocortex learning

In the previous section we address the hippocampus part of our algorithm, that is, we have access to expert's data that give information of how the system (1) should behave. In this section, the neocortex learning is developed.

Here on-line data measured along the system trajectories of (1) is used instead of expert's data [19]. Hence, the neocortex learning is equivalent to a reinforcement learning algorithm.

Journal Pre-proof

3.1. Model-based Dynamic Programming

First, let discuss the model-based dynamic programming algorithm [11]. Analogously to (3), the objective is to minimize the following value function $V(x) : \mathbb{R}^n \to \mathbb{R}$

$$V(x) = \int_t^\infty (x^\top Sx + u^\top Ru)d\tau, \tag{9}$$

where $S = S^\top > 0 \in \mathbb{R}^{n\times n}$ and $R = R^\top \geq 0 \in \mathbb{R}^{m\times m}$ are unknown weight matrices. Equivalently the optimal value function $V^*(x)$ can be written quadratic in the state as

$$V^*(x) = x^\top Px, \tag{10}$$

for some kernel matrix $P = P^\top > 0 \in \mathbb{R}^{n\times n}$. The Hamiltonian associated to (1) and the value function (9) and (10) is

$$H(x, u) = 2x^\top P(Ax + Bu) + x^\top Sx + u^\top Ru = 0. \tag{11}$$

The stationary condition $\frac{\partial H(x,u)}{\partial u} = 0$ is used to derive the optimal control law

$$u^* = -Kx = -R^{-1}B^\top Px, \tag{12}$$

for some stabilizing gain $K \in \mathbb{R}^{m\times n}$. Substituting (12) in the Hamiltonian (11) gives the following ARE

$$A^\top P + PA - PBR^{-1}B^\top P + S = 0. \tag{13}$$

It is important to make clear the differences between the hippocampus and neocortex models. The most evident difference is that the control gains K_e and K, the kernel matrices P_e and P, and the weight matrices S_e, S, R_e and R are different. The next and most important difference is that the hippocampus support the neocortex by providing experience in how to develop the task and fulfil the expert's requirements [39]. The interaction between the hippocampus and the neocortex is what we call complementary learning.

The action value function $Q(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is used to facilitate the design of the neocortex learning model. The Q-function satisfies the next

Journal Pre-proof

relation [54]

$$Q(x,u) = V^*(x) + H(x,u) \tag{14}$$

$$= \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} P + A^\top P + PA + S & PB \\ B^\top P & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

$$= \begin{bmatrix} x \\ u \end{bmatrix}^\top \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = z^\top M z \tag{15}$$

where $M = \begin{bmatrix} Q_{xx} & Q_{xu} \\ Q_{xu}^\top & Q_{uu} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$ and $z = \begin{bmatrix} x^\top & u^\top \end{bmatrix}^\top \in \mathbb{R}^{n+m}$. Notice that the optimal Q-function is equivalent to the optimal value function, that is, $Q^*(x,u^*) = V^*(x)$. The optimal control policy satisfies the stationary condition $\frac{\partial Q(x,u)}{\partial u} = 0$ and is given by

$$u^* = -Kx = -Q_{uu}^{-1} Q_{xu}^\top x. \tag{16}$$

3.2. Model-free Reinforcement Learning

The optimal control policy in terms of the Q-function avoids knowledge of the system dynamics by using an adequate function approximator. Consider the following parametrization

$$Q^*(x,u^*) := \theta^\top (z \otimes z), \tag{17}$$

where $\theta = \text{vech}(M) \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ and the non-diagonal terms of M are doubled because the matrix is symmetric. Since the optimal Q-function Q^* and the optimal control policy u^* are unknown, then the following approximation is considered

$$\widehat{Q}(x,u) = \widehat{\theta}^\top (z \otimes z), \tag{18}$$

where $\widehat{\theta} \in \mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)}$ are estimates of θ. The integral reinforcement learning [25] is used to derive the next Bellman equation in terms of the Q-function

$$\begin{aligned} Q^*(x(t),u^*(t)) = \; & Q^*(x(t+\mathcal{T}),u^*(t+\mathcal{T})) \\ & + \int_t^{t+\mathcal{T}} (x^\top S x + u^\top R u) d\tau \end{aligned} \tag{19}$$

Journal Pre-proof

for some fixed time $\mathcal{T}$. The approximation of (19) is

$$\begin{aligned}\widehat{Q}(x(t),u(t)) =&\widehat{Q}(x(t+\mathcal{T}),u(t+\mathcal{T}))\\ &+\int_t^{t+\mathcal{T}}(x^\top Sx+u^\top Ru)d\tau\\ \widehat{\theta}^\top(z(t)\otimes z(t)) =&\widehat{\theta}^\top(z(t+\mathcal{T})\otimes z(t+\mathcal{T}))\\ &+\int_t^{t+\mathcal{T}}(x^\top Sx+u^\top Ru)d\tau\end{aligned}$$

Define the temporal difference (TD) error [34] as

$$\begin{aligned}\delta(t)= &\left(z^\top(t+\mathcal{T})\otimes z^\top(t+\mathcal{T})-z^\top(t)\otimes z^\top(t)\right)\widehat{\theta}\\ &+\int_t^{t+\mathcal{T}}(x^\top Sx+u^\top Ru)d\tau.\end{aligned}\tag{20}$$

Let $\Phi := z(t+\mathcal{T})\otimes z(t+\mathcal{T})-z(t)\otimes z(t)$ and $\mathcal{R}(t)=\int_t^{t+\mathcal{T}}(x^\top Sx+u^\top Ru)d\tau$. The main objective of the neocortex learning method is to minimize the following cost index [6]

$$J=\frac{1}{2}\int\delta^2(t)dt.\tag{21}$$

The stationary condition $\frac{\partial J}{\partial\widehat{\theta}}=0$ is used to derive the update rule of the weights $\widehat{\theta}$. Using the inner product notation for the Lebesque integral as

$$\left\langle\frac{\partial\delta(t)}{\partial\widehat{\theta}},\delta(t)\right\rangle=0,\tag{22}$$

gives the following least-squares rule

$$\widehat{\theta}=-\langle\Phi,\Phi^\top\rangle^{-1}\langle\Phi,\mathcal{R}(t)\rangle.\tag{23}$$

The LS rule (23) is in fact a batch LS rule where the integral in (21) is spanned into $p\geq\frac{1}{2}(n+m)(n+m+1)$ independent linear equations. The following persistency of excitement (PE) condition is required to guarantee that the inner product $\langle\Phi,\Phi^\top\rangle$ is invertible [35].

Definition 1. *[55] The matrix $\Phi\in\mathbb{R}^{\frac{1}{2}(n+m)(n+m+1)\times p}$ is said to be persistent exciting if there exists constants $\beta_1,\beta_2>0$, and $T_1>0$ such that the following inequality is satisfied*

$$\beta_1 I\leq\int_t^{t+T_1}\Phi\Phi^\top d\tau\leq\beta_2 I.\tag{24}$$

Journal Pre-proof

The above definition means that the matrix $\Phi\Phi^\top$ does not lose rank. Furthermore, the PE signal belongs to the hippocampus learning system since it helps to get experience by exciting the modes of the system similarly to an exploration strategy in reinforcement learning [23].

Recall the elements of $\widehat{\theta}$ stands to $\widehat{Q}_{xx} = \widehat{\theta}\Big[1 : \frac{n(n+1)}{2}\Big]$, $\widehat{Q}_{xu} = \frac{1}{2}\widehat{\theta}\Big[\frac{n(n+1)+2}{2} : \frac{n(n+1)}{2} + nm\Big]$, and $\widehat{Q}_{uu} = \widehat{\theta}\Big[\frac{n(n+1)+2}{2} + nm : \frac{(n+m)(n+m+1)}{2}\Big]$. Then the approximate optimal solution is

$$\widehat{u}^* = -\widehat{K}x = -\widehat{Q}_{uu}^{-1}\widehat{Q}_{xu}^\top x. \tag{25}$$

Notice that from the Q-function properties [23], the following holds

$$\widehat{P} = \begin{bmatrix} I \\ -\widehat{K} \end{bmatrix}^\top \widehat{M} \begin{bmatrix} I \\ -\widehat{K} \end{bmatrix}, \tag{26}$$

where $\widehat{M} = \begin{bmatrix} \widehat{Q}_{xx} & \widehat{Q}_{xu} \\ \widehat{Q}_{xu}^\top & \widehat{Q}_{uu} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$.

Convergence of the least squares method using neural networks as approximator is discussed in [6]. For the proposed approach, we consider a complete set of basis functions such that the residual error is zero and hence $\widehat{\theta} \to \theta$ and $\widehat{P} \to P$ as the number of samples p are increased.

The gain matrix $\widehat{K}$ and the kernel matrix $\widehat{P}$ are used in the next section to develop the complementary learning algorithm.

4. Complementary Learning

In this section, the complementary learning relation between the hippocampus and neocortex learning methods is developed and discussed. The main goal of the CL is to seek the weights S_e and R_e of the utility function $\rho_e(\cdot)$ such that the control gain $\widehat{K}$ converges to a close approximation of the gain $\widehat{K}_e$.

4.1. Kernel matrix and Control gain update

First, we need to define how can we relate the expert's data and the online data. One way is by means the control gains K_e and K in (6) and (12),

Journal Pre-proof

respectively. On the other hand, notice that the gain $K = R^{-1}B^\top P$ depends on the weight matrix R and the kernel matrix P. For sake of simplicity, let assume that the weight matrices $R = R_e$ are known in advance, such that the gain K depends only of the kernel matrix P. Therefore, the CL obtains an improved weight matrix S which subsequently will be used by the neocortex learning model to achieve the performance of the expert's control policy.

Let the approximate gain (25) can be written as $\widehat{K} = R^{-1}B^\top \widehat{P} \in \mathbb{R}^{m\times n}$, where $\widehat{P}$ is defined as in (26). Then, we can define a control gain error $e_K \in \mathbb{R}^{m\times n}$ between (8) and (25) as

$$e_K = \widehat{K}_e - \widehat{K} = -\left(\bar{u}_e\bar{x}_e^\top(\bar{x}_e\bar{x}_e^\top)^{-1} + R^{-1}B^\top\widehat{P}\right). \tag{27}$$

Define the next cost index

$$E = \frac{1}{2}\mathrm{tr}\{e_K^\top e_K\}. \tag{28}$$

The gradient descent technique is used to obtain a new kernel matrix $\mathcal{P}$ which yields a close approximation of the expert's performance as

$$\begin{aligned} \mathcal{P} &= \widehat{P} - \alpha\frac{\partial E}{\partial \widehat{P}}, \\ &= \widehat{P} + \frac{1}{2}\alpha(e_K^\top R^{-1}B^\top + BR^{-1}e_K), \end{aligned} \tag{29}$$

where $\alpha \in \mathbb{R}^+$ is the learning rate.

Remark 1. *The gradient rule (29) needs to be expressed in quadratic form to guarantee a symmetric matrix $\mathcal{P}$.*

The update rule (29) cannot be used since it requires knowledge of matrix B. To avoid this issue, the equivalence $R^{-1}B^\top = \widehat{K}\widehat{P}^{-1}$ is used since $\mathrm{rank}(\widehat{P}) = n$. Hence, the gradient rule is rewritten as

$$\mathcal{P} = \widehat{P} + \frac{1}{2}\alpha(e_K^\top\widehat{K}\widehat{P}^{-1} + \widehat{P}^{-1}\widehat{K}^\top e_K). \tag{30}$$

Here the gradient rule is used as a correction term of $\widehat{P}$. Notice that the gain error e_K approaches to zero as $\widehat{K}$ tends to $\widehat{K}_e$ and hence the kernel matrices $\mathcal{P}$ converges to a kernel matrix $\widehat{P}$ which satisfies the expert's performance.

Journal Pre-proof

The new kernel matrix $\mathcal{P}$ is used to derive a new control gain $K_c = R^{-1}B^\top \mathcal{P} \in \mathbb{R}^{m\times n}$ as follows

$$\begin{aligned} K_c &= R^{-1}B^\top \mathcal{P} \\ &= \widehat{K}\widehat{P}^{-1}\mathcal{P}. \end{aligned} \tag{31}$$

Notice that $K_c = \widehat{K}$ if $\mathcal{P} = \widehat{P}$.

4.2. Inverse Reinforcement Learning

The kernel matrix $\mathcal{P}$ and the gain matrix K_c facilitate the formulation of the inverse reinforcement learning (IRL) algorithm [56] to derive and update the new weight matrix S. The IRL algorithm is designed as a Lyapunov recursion algorithm [19].

The Hamiltonian (12) written in terms of the ARE (13) and the kernel matrix $\mathcal{P}$ is

$$\begin{aligned} x^\top(A^\top\mathcal{P} + \mathcal{P}A - \mathcal{P}BR^{-1}B^\top\mathcal{P} + S)x = 0 \\ (Ax + Bu_c)^\top\mathcal{P}x + x^\top\mathcal{P}(Ax + Bu_c) + x^\top Sx \\ -x^\top K_c R K_c x - 2u_c^\top R K_c x = 0 \\ \frac{d}{dt}(x^\top\mathcal{P}x) + x^\top Sx - x^\top K_c^\top R K_c x - 2u_c^\top R K_c x = 0. \end{aligned} \tag{32}$$

Integrating (32) in a time-interval of $[t : t + \mathcal{T}]$ gives

$$\begin{aligned} \int_t^{t+\mathcal{T}} x^\top S x d\tau =& x^\top(t)\mathcal{P}x(t) - x(t+\mathcal{T})\mathcal{P}x(t+\mathcal{T}) \\ &+ \int_t^{t+\mathcal{T}} x^\top K_c^\top R K_c x d\tau + 2\int_t^{t+\mathcal{T}} u_c^\top R K_c x d\tau \end{aligned} \tag{33}$$

Note that (33) is a scalar equation which is linear in terms of the unknown matrix S. Since each component of matrix S are unknown, then there are more unknowns than the number of equations. So, a system of l independent equations are developed by performing l window integrals with length $\mathcal{T} =$

Journal Pre-proof

Algorithm 1 Expertise learning algorithm

Require: Weight matrix $R > 0$, expert's data $\bar{x}_e$ and $\bar{u}_e$, initial weight matrix $S_0 \geq 0$, and learning rate α.

1: Obtain the expert's gain matrix $\widehat{K}_e$ of the hippocampus learning using (8). Set $i = 0$ and $\widehat{K}_0^0 = \widehat{K}_e$. Apply control input $u_0^0 = -\widehat{K}_0^0 x$.
2: **repeat**
3: Set $j = 0$.
4: **repeat**
5: Update the weights θ_i^j using least-squares

$$\widehat{\theta}_i^j = \left\langle (-\Phi\Phi^\top)^{-1}\Phi, \int_t^{t+\mathcal{T}} (x^\top S_0 x + (u_i^j)^\top R u_i^j) d\tau \right\rangle \tag{34}$$

6: Update the control input $u_i^j = -\widehat{K}_i^j x$ using (25).
7: Set $j = j + 1$.
8: **until** $\|\widehat{\theta}_i^j - \widehat{\theta}_i^{j-1}\| \leq \varepsilon_\theta$
9: Obtain $\widehat{P}_i$ using (26).
10: Calculate the gain error $(e_K)_i$ using (27) and E_i using (28).
11: Obtain the new kernel matrix $\mathcal{P}_i$ using the gradient descent rule (30).
12: Calculate the new matrix gain $(K_c)_i$ using (31)
13: Calculate S_i using least-squares

$$\begin{aligned} \mathrm{vec}(S_{i+1}) = \; & (I_{xx}^\top I_{xx})^{-1} I_{xx}^\top (W\mathrm{vec}(\mathcal{P}_i) + 2I_{xu}\mathrm{vec}(R(K_c)_i)) \\ & \mathrm{vec}((K_c)_i^\top R(K_c)_i) \end{aligned} \tag{35}$$

14: Set $i = i + 1$ and $\widehat{K}_i^0 = (K_c)_i$.
15: **until** $\|\mathrm{vec}(S_i) - \mathrm{vec}(S_{i-1})\| \leq \varepsilon_S$ and $E_i \leq \varepsilon_K$.

$t_i - t_{i-1}$ for $i = 1, \cdots, l$. The following matrices are defined

$$\begin{aligned} I_{xx} &= \left[\int_{t_0}^{t_1} x\bar{\otimes}x d\tau, \int_{t_1}^{t_2} x\bar{\otimes}x d\tau, \cdots, \int_{t_{l-1}}^{t_l} x\bar{\otimes}x d\tau\right]^\top, \\ I_{xu} &= \left[\int_{t_0}^{t_1} x\bar{\otimes}u_c d\tau, \int_{t_1}^{t_2} x\bar{\otimes}u_c d\tau, \cdots, \int_{t_{l-1}}^{t_l} x\bar{\otimes}u_c d\tau\right]^\top, \\ W &= -\left[x(t)\bar{\otimes}x(t)\Big|_{t_0}^{t_1}, x(t)\bar{\otimes}x(t)\Big|_{t_1}^{t_2}, \cdots, x(t)\bar{\otimes}x(t)\Big|_{t_{l-1}}^{t_l}\right]^\top. \end{aligned}$$

Journal Pre-proof

where $I_{xx} \in \mathbb{R}^{l \times n^2}$, $I_{xu} \in \mathbb{R}^{l \times nm}$, and $W \in \mathbb{R}^{l \times n^2}$. Then, the batch least-squares rule for the next linear system of equations based in (33) is

$$\begin{aligned} I_{xx}\text{vec}(S) = &\quad W\text{vec}(\mathcal{P}) + I_{xx}\text{vec}(K_c^\top RK_c) + 2I_{xu}\text{vec}(RK_c), \\ \text{vec}(S) = &\quad (I_{xx}^\top I_{xx})^{-1} I_{xx}^\top (W\text{vec}(\mathcal{P}) + 2I_{xu}\text{vec}(RK_c)) \\ &+\text{vec}(K_c^\top RK_c) \end{aligned} \tag{36}$$

The last sections provide the complete picture of the proposed approach. However, they need to be written in a recursive form to guarantee adaptability to new changes of the weight matrix S, kernel gain $\widehat{K}$ and kernel matrix $\widehat{P}$. Algorithm 1 shows the recursive algorithm of the proposed complementary learning approach, where $\varepsilon_\theta, \varepsilon_S, \varepsilon_K \in \mathbb{R}^+$ are some small error tolerances proposed by the user.

The following theorem establishes the convergence of the weight matrix S to the expert's weight S_e using the proposed complementary learning algorithm.

Theorem 1. *The weight matrix S_i obtained by Algorithm 1 converges to a new improved weight matrix S_{i+1} such that the neocortex gain $\widehat{K}$ converges to $\widehat{K}_e$ as the number of iterations i increases.*

Proof 1. *See Appendix A.* □

5. Simulation studies

5.1. Example 1

The F16 aircraft dynamics used in [27] was considered. The following dynamics were considered

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -20.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 20.2 \end{bmatrix}.$$

Journal Pre-proof

Consider an expert controller which was designed with the following weight matrices: $S_e = 10I_3$ and $R_e = 1$. The optimal kernel matrix and gain were

$$P_e = \begin{bmatrix} 13.6596 & 11.0594 & -0.0294 \\ 11.0594 & 13.6855 & -0.0360 \\ -0.0294 & -0.0360 & 0.1148 \end{bmatrix}$$
$$K_e = \begin{bmatrix} -0.5938 & -0.7274 & 2.3185 \end{bmatrix}.$$

States and control input data were stored as expert's data for the first 3 seconds of simulation time with a sample time of $\mathcal{T} = 0.05$ s and the Dormand-Prince solver. The data collection contains 67 data points for each state and control input. Equation (8) was used to compute the control gain of the hippocampus model as

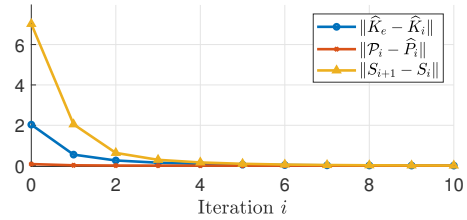
$$\widehat{K}_e = \begin{bmatrix} -0.5938 & -0.7274 & 2.3185 \end{bmatrix}.$$

For this case, the hippocampus gain $\widehat{K}_e$ is equivalent to the expert's gain K_e. For the neocortex model it is considered the following initial weight matrices: $S_0 = I_3$ and $R = 1$. A PE signal composed of a sum of sinusoidal functions with different frequencies was added to the control input to guarantee parameter convergence.

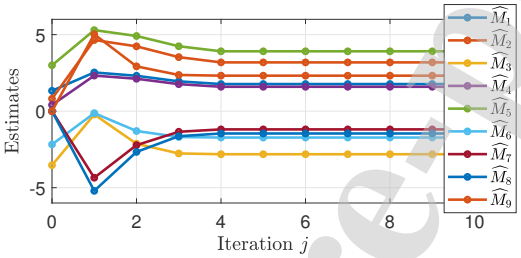
For the complementary learning part, the learning rate was set to $\alpha = 0.002$. Large learning rates were tested in different simulations, however they caused instability in the learning trajectories. The term $\widehat{Q}_{uu}$ was not estimated because it is equivalent to the known matrix R. The neocortex learning model used $p = 20$ stored data points measured online to compute the batch LS algorithm (34). For the IRL, $l = 6$ stored data points were used for the batch LS algorithm of (35). The convergence results were evaluated by taking the norm of the control gain error and the difference between previous iterations, that is, $\|\widehat{K}_e - \widehat{K}\|$, $\|\mathcal{P}_i - \widehat{P}_i\|$, and $\|S_{i+1} - S_i\|$. Figure 2 shows the convergence results of the IRL and the neocortex learning model.

As shown in Figure 2(a), if the control gain $\widehat{K}$ converges to $\widehat{K}_e$, then the kernel matrix $\widehat{P}$ converges to $\widehat{P}$ and hence the weight matrix S converges. The

Journal Pre-proof



(a) Convergence of the IRL model



(b) Convergence of the neocortex learning model

Figure 2: Convergence of the complementary learning approach

numerical results were

$$\mathcal{P} = \widehat{P} = \begin{bmatrix} 1.4235 & 1.1662 & -0.0295 \\ 1.1662 & 1.4318 & -0.0361 \\ -0.0295 & -0.0361 & 0.1148 \end{bmatrix},$$

$$S = \begin{bmatrix} 1.3315 & 0.4064 & -1.7652 \\ 0.4064 & 1.4983 & -2.1747 \\ -1.7652 & -2.1747 & 10 \end{bmatrix},$$

$$\widehat{K} = \begin{bmatrix} -0.5938 & -0.7274 & 2.3185 \end{bmatrix}$$

The states trajectories under the optimal control $\widehat{u}^*$ are shown in Figure 3. Notice that both $\widehat{P}$ and S were different to the expert's kernel matrix P_e and S_e. This is because the inverse problem has multiple solutions for the kernel matrix P and the weight matrix S that derive the same control gain $\widehat{K}_e$. However, the obtained results can be seen as an optimization of the weight matrix S and the kernel matrix $\widehat{P}$. To see this fact, Figure 4 shows the comparison of the

Journal Pre-proof

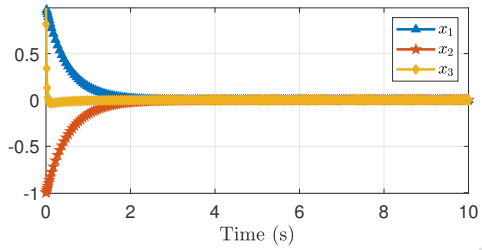


Figure 3: States trajectories.

optimal value function (4) and the optimal value function approximation, that is, $\widehat{V}^* = x^\top \widehat{P} x$.

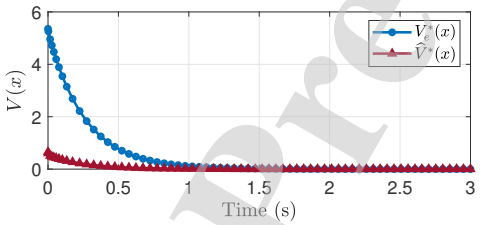


Figure 4: Value function $V(x)$ comparisons.

The figure shows that the new weight matrix S minimizes the kernel matrix $\widehat{P}$ and hence the value function $\widehat{V}$ was also minimized. This result shows that the proposed complementary approach is capable to achieve the expert's performance and simultaneously seeks the weight matrix S that optimizes the kernel matrix $\widehat{P}$.

5.2. Example 2

To further illustrate the effectiveness of the approach and validate the facts that were stated in the previous example, we modify the initial weight matrix and set it to

$$S_0 = \begin{bmatrix} 1.4235 & 1.1662 & -0.0295 \\ 1.1662 & 1.4318 & -0.0361 \\ -0.0295 & -0.0361 & 0.1148 \end{bmatrix}.$$

Journal Pre-proof

The main goal of this second example is to verify that the proposed approach can handle with a weight matrix which is not diagonal and if the complementary learning algorithm can estimate and improve the weight matrix S as it was shown in the previous example.

The same expert's data was used and hence the same hippocampus learning model. The same hyperparameters and data points were used in this example. The convergence results are shown in Figure 5.

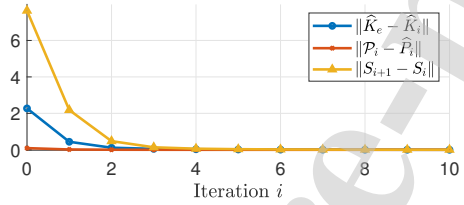


Figure 5: Convergence of the IRL model.

The results show that the complementary learning algorithm converged to the expert's control gain. For this case, the kernel matrix $\mathcal{P}$ and the weight matrix S converged to the following values

$$\mathcal{P} = \widehat{P} = \begin{bmatrix} 2.8100 & 2.7685 & -0.0294 \\ 2.7685 & 2.8640 & -0.0360 \\ -0.0294 & -0.0360 & 0.1148 \end{bmatrix},$$

$$S = \begin{bmatrix} 1.5258 & 1.3374 & -1.4787 \\ 1.3374 & 1.6892 & -1.9176 \\ -1.4787 & -1.9176 & 10.0000 \end{bmatrix}.$$

Figure 6 shows the comparison of the expert's optimal value function $V_e^*(x)$, and the optimal value function approximators $\widehat{V}_1(x)$ and $\widehat{V}_2(x)$ obtained in example 1 and example 2, respectively.

The above figure shows that the value function was improved again by considering a non-diagonal weight matrix. The norm of each value function was used as performance metric, the results were: $\|V^*(x)\| = 14.1834$, $\|\widehat{V}_1^*(x)\| = 1.5027$, and $\|\widehat{V}_2^*(x)\| = 0.4749$. Here it is evident that the proposed approach improves

Journal Pre-proof

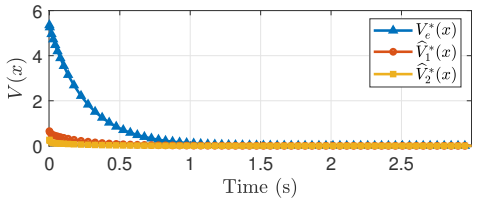


Figure 6: Value function $V(x)$ comparisons.

the expert's performance. Furthermore, the initial weight matrix S_0 in the neocortex model plays a fundamental role in the update direction of the next weight matrix. In other words, the complementary learning algorithm will converge to one solution of the inverse optimal control problem depending on the initialization of the weight matrix S_0.

6. Conclusions

In this paper a complementary learning algorithm is proposed for the expertise transference of human-optimized controllers. The algorithm takes advantage of the complementary learning properties of the hippocampus and neocortex learning systems. The hippocampus is designed as independent pattern separated data which provides of expert's knowledge in how the task has to been developed. The neocortex is designed as a Q-reinforcement learning algorithm to gain new information by interacting online with new data. The complementary learning part is designed as an inverse reinforcement learning problem which computes one of the solutions of the inverse optimal control problem which achieves the desired expert's performance.

Simulation results show that the proposed approach learns the expert's control policy and optimizes its value function by minimizing the weight matrix of the utility function and the kernel matrix. Furthermore, the solution of the IRL depends on the initialization of the utility function weight matrix.

Further work will analyse how to constraint the IRL problem such that the number of possible solutions is reduced. Moreover, the work will be spanned in

Journal Pre-proof

the human-behavior learning context and use other learning techniques which can complement and improve the proposed approach. In addition, experience transference for nonlinear system is a more challenging future work problem due to the nonlinear nature of the control policy.

Appendix A. Proofs

Proof 2. *First, it is important to recall that the Lyapunov recursions method of the IRL part of Algorithm 1 is equivalent to the following ARE*

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

Substituting (31) in (A.1) gives

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

Recall that the neocortex learning model satisfies the next ARE

$$-S_{i+1} = A^\top \widehat{P}_{i+1} + \widehat{P}_{i+1} A - \widehat{P}_{i+1} B R^{-1} B^\top \widehat{P}_{i+1}. \tag{A.3}$$

Substituting (A.3) in (A.2) gives

$$\begin{aligned} & A^\top \widehat{P}_{i+1} + \widehat{P}_{i+1} A - \widehat{P}_{i+1} B R^{-1} B^\top \widehat{P}_{i+1} \\ & = A^\top \widehat{P}_i + \widehat{P}_i A - \widehat{P}_i B R^{-1} B^\top \widehat{P}_i - \alpha \left(A^\top \frac{\partial E_i}{\partial \widehat{P}_i} + \frac{\partial E_i}{\partial \widehat{P}_i} A \right) \\ & \quad + \alpha \left(\widehat{P}_i B R^{-1} B^\top \frac{\partial E_i}{\partial \widehat{P}_i} + \frac{\partial E_i}{\partial \widehat{P}_i} B R^{-1} B^\top \widehat{P}_i \right) \\ & \quad - \alpha^2 \frac{\partial E_i}{\partial \widehat{P}_i} B R^{-1} B^\top \frac{\partial E_i}{\partial \widehat{P}_i}. \end{aligned} \tag{A.4}$$

The gradient descent rule (32) updates the kernel matrix $\mathcal{P}$ in each iteration i such that the error e_K is minimized, that is, $\widehat{K} \to \widehat{K}_e$, then

$$\lim_{i \to \infty} \frac{\partial E_i}{\partial \widehat{P}_i} = 0. \tag{A.5}$$

From the above result it follows that

$$\lim_{i \to \infty} \mathcal{P}_i = \widehat{P}_i.$$

Journal Pre-proof

Therefore, for an infinite number of iterations i we have that (A.4) is simplified to

$$\lim_{i\to\infty} S_{i+1} = \lim_{i\to\infty} S_i, \tag{A.6}$$

which implicitly implies that $\lim_{i\to\infty} \widehat{P}_{i+1} = \widehat{P}_i$. *This completes the proof.* □

References

[1] A. Perrusquía, W. Yu, Human-in-the-loop control using Euler angles, Journal of Intelligent & Robotic Systems 97 (2020) 271–285. doi:10.1007/s10846-019-01058-2.

[2] X. Li, W. Yu, A systematic tunning method of PID controller for robot manipulators, 2011 9th IEEE International Conference on Control and Automation (ICCA) (2011) 274–279doi:10.1109/ICCA.2011.6138081.

[3] W. Yu, A. Perrusquía, Simplified stable admittance control using end-effector orientations, International Journal of Social Robotics 12 (5) (2020) 1061–1073. doi:10.1007/s12369-019-00579-y.

[4] A. Perrusquía, J. A. Flores-Campos, C. Torres-San-Miguel, A novel tuning method of PD with gravity compensation controller for robot manipulators, IEEE Access 8 (2020) 114773–114783. doi:10.1109/ACCESS.2020.3003842.

[5] D. Luviano, W. Yu, Continuous-time path planning for multi-agents with fuzzy reinforcement learning, Journal of Intelligent & Fuzzy Systems 33 (2017) 491–501. doi:10.3233/JIFS-161822.

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

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

Journal Pre-proof

[8] C. G. Atkeson, S. Schaal, Robot learning from demonstration, in: ICML, Vol. 97, Citeseer, 1997, pp. 12–20. doi:10.1007/springerreference_302487.

[9] P. Kormushev, S. Calinon, D. G. Caldwell, Imitation learning of al and force skills demonstrated via kinesthetic teaching and haptic input, Advanced Robotics 25 (5) (2011) 581–603. doi:10.1163/016918611X558261.

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

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

[12] 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. doi:10.1016/j.robot.2008.10.024.

[13] 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 (2020). doi:10.1146/annurev-control-100819-063206.

[14] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Handbook of robotics chapter 59: Robot programming by demonstration, Handbook of Robotics. Springer (2008). doi:10.1007/978-3-540-30301-5_60.

[15] A. Segre, G. DeJong, Explanation-based manipulator learning: Acquisition of planning ability through observation, in: Proceedings. 1985 IEEE International Conference on Robotics and Automation, Vol. 2, IEEE, 1985, pp. 555–560. doi:10.1109/ROBOT.1985.1087311.

[16] T. Lozano-Perez, Robot programming, Proceedings of the IEEE 71 (7) (1983) 821–841. doi:10.1109/proc.1983.12681.

Journal Pre-proof

[17] 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. doi:10.1016/j.neucom.2021.01.096.

[18] B. Kiumarsi, G. V. Kyriakos, 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. doi:10.1109/TNNLS.2017.2773458.

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

[20] B. Luo, H.-N. Wu, Huand-Tingwen, D. Liu, Reinforcement learning solution for HJB equation arising in constrained optimal control problem, Neural Networks 71 (2015) 150–158. doi:10.1016/j.neunet.2015.08.007.

[21] A. Perrusquía, W. Yu, Discrete-time $\mathcal{H}_2$ neural control using reinforcement learning, IEEE Transactions on Neural Networks and Learning Systems (2020). doi:10.1109/TNNLS.2020.3026010.

[22] 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. doi:10.1109/TAC.2014.2317301.

[23] A. Perrusquía, W. Yu, A. Soria, Position force/control of robot manipulators using reinforcement learning, Industrial Robot 46 (2) (2019) 267–280. doi:10.1108/IR-10-2018-0209.

[24] Q. Xie, B. Luo, F. Tan, Discrete-time LQR optimal tracking control problems using approximate dynamic programming algorithm with disturbance,

Journal Pre-proof

Fourth International Conference on Intelligent Control and Information Processing (ICICIP) (2013) 716–721doi:10.1109/ICICIP.2013.6568166.

[25] 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. doi:10.1109/TCYB.2014.2322116.

[26] K. Vamvoudakis, D. Vrabie, F. L. Lewis, Online policy iteration based algorithms to solve the continuous-time infinite horizon optimal control problem, 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (2009). doi:10.1109/ADPRL.2009.4927523.

[27] K. G. Vamvoudakis, F. L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010) 878–888. doi:10.1109/IJCNN.2009.5178586.

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

[29] 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. doi:10.1002/rnc.4911.

[30] A. Perrusquía, W. Yu, Continuous-time reinforcement learning for robust control under worst-case uncertainty, International Journal of Systems Science (2020) 1–15doi:10.1080/00207721.2020.1839142.

[31] J. A. 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.

Journal Pre-proof

[32] A. Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, in: in Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 663–670. doi:10.2460/AJVR.67.2.323.

[33] P. Abbeel, A. Y. Ng, Apprenticeship learning via inverse reinforcement learning, Proceedings of the twenty-first International Conference on Machine Learning (2004). doi:10.1145/1015330.1015430.

[34] A. Perrusquía, W. Yu, Neural $\mathcal{H}_2$ control using continuous-time reinforcement learning, IEEE Transactions on Cybernetics (2020). doi:10.1109/TCYB.2020.3028988.

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

[36] 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 (2017). doi:10.1017/S0140525X16001837.

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

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

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

[40] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, S. Ahmad, A framework for intelligence and cortical function based on grid cells in the neocortex, Frontiers in neural circuits 12 (2019) 121. doi:10.3389/fncir.2018.00121.

Journal Pre-proof

[41] 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. doi:10.1523/JNEUROSCI.12-12-04595.1992.

[42] 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. doi:10.1016/j.tics.2016.05.004.

[43] 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. doi:10.1523/JNEUROSCI.3212-16.2017.

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

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

[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. doi:10.1037/0033-295X.102.3.419.

[47] Y. Liang, F. He, X. Zeng, 3d mesh simplification with feature preservation based on whale optimization algorithm and differential evolution, Integrated Computer-Aided Engineering (Preprint) (2020) 1–19. doi:10.3233/ICA-200641.

Journal Pre-proof

[48] H. Li, F. He, Y. Chen, Y. Pan, Mlfs-ccde: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution, Memetic Computing 13 (1) (2021) 1–18. doi:10.1007/s12293-021-00328-7.

[49] Y. Chen, F. He, H. Li, D. Zhang, Y. Wu, A full migration bbo algorithm with enhanced population quality bounds for multimodal biomedical image registration, Applied Soft Computing 93 (2020) 106335. doi:10.1016/j.asoc.2020.106335.

[50] Y. Park, Inverse optimal and robust nonlinear attitude control of rigid spacecraft, Aerospace Science and Technology 28 (1) (2013) 257–265. doi:10.1016/j.ast.2012.11.006.

[51] M. Johnson, N. Aghasadeghi, T. Bretl, Inverse optimal control for deterministic continuous-time nonlinear systems, in: 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 2906–2913. doi:10.1109/CDC.2013.6760325.

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

[53] J. Y. Lee, J. B. Park, Y. H. Choi, Integral q-learning and explorized policy iteration for adaptive optimal control of continuos-time linear systems, Automatica 48 (2012) 2850–2859. doi:10.1016/j.automatica.2012.06.008.

[54] K. G. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, Systems & Control Letters 100 (2017) 14–20. doi:10.1016/j.sysconle.2016.12.003.

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

[56] 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. doi:10.1016/j.arcontrol.2020.06.001.

Declaration of interests

☒ 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.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Journal Pre-proof

https://dspace.org/

A complementary learning approach for expertise transference of human-optimized controllers

Perrusquía, Adolfo

2021-10-21
Elsevier
https://doi.org/10.1016/j.neunet.2021.10.009
Perrusquia A. (2022) A complementary learning approach for expertise transference of human-optimized controllers. Neural Networks, Volume 145, January 2022, pp. 33-41

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