2021 60th IEEE Conference on Decision and Control (CDC), 14-17 December 2021, Austin, Texas, USA
10.1109/CDC45484.2021.9683719

Human-Behavior Learning for Infinite-Horizon Optimal Tracking Problems of Robot Manipulators

Adolfo Perrusquía, Wen Yu

***Abstract*— In this paper, a human-behavior learning approach for optimal tracking control of robot manipulators is proposed. The approach is a generalization of the reinforcement learning control problem which merges the capabilities of different intelligent and control techniques in order to solve the tracking task. Three cognitive models are used: robot and reference dynamics and neural networks. The convergence of the algorithm is achieved under a persistent exciting and experience replay fulfillment. The algorithm learns online the optimal decision making controller according to the proposed cognitive models. Simulations were carry out to verify the approach using a 2-DOF planar robot.**

I. Introduction

Human-behavior learning [1] has become a popular research topic in different communities, specially in video games applications [2]. These kind of applications extract special features of the human performance while playing a video game using a deep reinforcement learning architecture known as Deep Q Network (DQN) [3]. The main components of this algorithm are: a deep neural network (designed with convolutional and fully connected layers [4]), a reinforcement learning update rule (Q-learning rule [5]–[7]), a ε-greedy strategy for exploration-exploitation [8], [9] of the state-action space and an experience replay memory [10]. To our knowledge, human-behavior learning has been applied only on video games applications in a discrete-time domain [11], that is, the system has discrete actions. Therefore, human-behavior learning design for control applications is an open problem and main contribution of this work.

In this paper, the human-behavior learning is analyzed in a new perspective which is more similar to how human learns and fills the current gap in control applications. Whilst reinforcement learning (RL) is modeled by a tuple of states, actions, state-transition functions, and rewards, human-behavior learning is modeled by a tuple of actions, cognitions, and emotions. The main difference between these two approaches lies in the cognitions where different models and functions are used to extract experiences [12] and any previous knowledge [13], [14] that facilitates obtaining the solution of the desired task in an optimal way. Some examples of cognitive models are knowledge of the system and environment dynamics [9], [15] or any intelligent model/expert system [3], [16] like neural networks [17]–[19], function approximators [20]–[24], fuzzy systems [25], deep models [26], among others. The emotions defines a complex set and its topic for future work. For simplicity, this work assumes that emotions are equivalent to rewards.

Control of linear systems has been studied in [4], [6], [27], [28] using model-based and model-free RL controllers [24]. Furthermore, nonlinear systems control has been assessed in [18], [23] using actor-critic structures and partial knowledge of the system dynamics. Despite having good results, the above approaches do not consider past experiences and previous knowledge as in a human-behavior learning approach. In this work, robot manipulators are analyzed as a special case of nonlinear systems. However, the proposed approach can also be extended to any stabilizable nonlinear system for both regulation and tracking tasks.

The main goal of the human-behavior learning approach is to find an optimal decision making function (known as policy in RL framework) which achieves an optimal solution of the desired task by minimizing/maximizing the emotions in an infinite horizon. For nonlinear systems, this function can be regarded as the solution of a Hamilton-Jacobi-Bellman (HJB) equation [29]. Nevertheless, it is an almost impossible task to obtain a solution of the HJB equation for nonlinear systems even we have full dynamics knowledge [30].

In this paper, the human-behavior learning for video games is modified into a control problem to achieve the above goal. Three cognitive models are used: robot and reference dynamics and a neural network approximator. The ε-greedy strategy is changed into a persistent exciting (PE) condition [31] to guarantee convergence of a model-based neural reinforcement learning algorithm. Experience replay is generally used to store some samples and use them as targets for the deep learning architecture [10]. In this new approach, the experience replay is used to store past experiences and use them at the reinforcement learning update rule, similarly to eligibility traces [32], [33]. This modification improves the convergence of the algorithm and the accuracies of the neural estimates. Simulation studies verify our approach using a 2-DOF planar robot.

II. Preliminaries

Human behavior is composed of three main sets (see Fig. 1): actions $\mathcal{A}$, cognitions $\mathcal{C}$ and emotions $\mathcal{E}$. Actions $\mathcal{A}$ denote everything that can be observed from physiological sensors, for example, any movement of the body; these actions can take place on different time scales, for example, reading a book, food consumption, sleep, etc. Cognitions $\mathcal{C}$ describe thoughts, skills and knowledge, for example, how to use a broom, computer or a cellphone application.

Adolfo Perrusquía is with the School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford MK43 0AL, UK.

Wen Yu is with the Departamento de Control Automatico, CINVESTAV-IPN (National Polytechnic Institute), Av.IPN 2508, Mexico City, 07360, Mexico. yuw@ctrl.cinvestav.mx

Emotions $\mathcal{E}$ describes a brief conscious experience that are not characterized as a result from either reasoning or a cognitive knowledge. Emotions are defined in a relative scale, that is, from positive emotions (happiness, excited, pleasurable) to negative emotions (angry, sad, unpleasant).

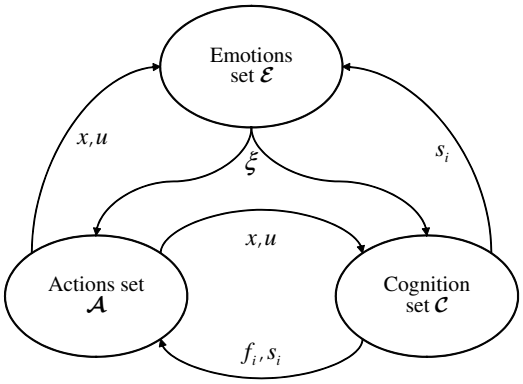


Fig. 1. Human-Behavior Learning components

Mathematically, human behavior can be written as a Markov decision process (MDP) given by the tuple $(\mathcal{A}, \mathcal{C}, \mathcal{E})$. Each MDP component are described as follows:

• The actions set $\mathcal{A}$ contains two subspaces: the set of states X and the set of controls U, so $\mathcal{A} = X \cup U$. X and U define all the possible states and controls of the system. The actions set contains a decision making function from states to controls denoted by $h : X \to U$, which describes which controls have to been applied in certain states in accordance to the cognition set $\mathcal{C}$ and the received emotion ξ.

• The cognition set $\mathcal{C}$ contains all previous knowledge, learning methods or skills that the human possess. Each previous knowledge i can be defined by an application from an action pair to a state value, for example, a state-transition function $f_i : X \times U \to X$ which describes how the state $x \in X$ changes as a result of applying a control $u \in U$. The skills $s_i : X \times U \to X$ are methods that the human uses to facilitate the execution of a task in accordance to its experience or a stimuli.

• The emotion set $\mathcal{E}$ provides feedback stimuli about the interaction between $\mathcal{A}$ and $\mathcal{C}$. This feedback can be interpreted as an application denoted by $\xi : X \times U \to \mathbb{R}$. Also the stimuli can be modified according to the cognitive skills s_i.

Remark 1: The main difference between reinforcement learning and human-behavior learning is the cognitive model which contains as many state-transition functions f_i of skills or knowledge that the human acquires while learning.

III. Human-Behavior Learning Control Design

The dynamics of a n-degree of freedom (DOF) robot manipulator (without friction) [34] is

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau, \tag{1}$$

where $M(q) \in \mathbb{R}^{n\times n}$ denotes the symmetric inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal forces matrix, $G(q) \in \mathbb{R}^n$ denotes the gravitational torques vector, $\tau \in \mathbb{R}^n$ is the control torque and $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the joint positions, velocities and accelerations, respectively. The robot dynamics (1) can be written as

$$\begin{aligned}\dot{x}_t &= \begin{bmatrix} x_{n+1} \\ \vdots \\ x_{2n} \\ -M^{-1}(q)\left[C(q,\dot{q})\dot{q} + G(q)\right] \end{bmatrix} + \begin{bmatrix} 0_{n\times n} \\ M^{-1}(q) \end{bmatrix} u_t \\ &= f(x_t) + g(x_t)u_t,\ x_{t_0} = x_0,\ t \geq t_0 \end{aligned} \tag{2}$$

where $f(x_t) \in \mathbb{R}^{2n}$ and $g(x_t) \in \mathbb{R}^{2n\times n}$ define the non-linear robot dynamics; with $x_{1,t} = q_{1,t}, \cdots, x_{n,t} = q_{n,t}$, $x_{n+1,t} = \dot{q}_{1,t}, \cdots, x_{2n,t} = \dot{q}_{n,t}$. Define $x_t = [x_{1,t}, \cdots x_{n,t}, x_{n+1,t}, \cdots, x_{2n,t}]^\top$ and $u_t = \tau$.

The robot is regarded as the acting human. The actions set $\mathcal{A}$ is given by all robot states $x_t \in \mathcal{A} \subset \mathbb{R}^{2n}$ (joint positions and joint velocities) and the torque inputs $u_t \in \mathcal{A} \subset \mathbb{R}^n$. When a control torque u_t is applied at the robot joints, then the states x_t are modified to a new state according to the robot state-transition function (2). The state-transition function (2) belongs to the cognition set $\mathcal{C}$.

Humans have the skill to explore all their actions possibilities in order to determine the best decision making. This skill belong to the cognition set $\mathcal{C}$ and can be designed with any exploration technique, for example, ε-greedy exploration [9], softmax [26] or a persistent exciting (PE) condition [31]. In this paper a PE condition is used as exploration skill.

For a tracking control problem, the objective is to force the robot states x_t to track a smooth desired trajectory $x_t^d \in \mathbb{R}^n$ [35]. The desired trajectory dynamics is regarded as a cognitive model for the human-behavior learning, which satisfies the following nonlinear reference model

$$\dot{x}_t^d = \varphi(x_t^d), \tag{3}$$

with known initial condition. Here $\varphi(\cdot)$ denotes the dynamics of the reference model. The tracking error is defined as $e_t = x_t - x_t^d$. The closed-loop dynamics between the cognitive model (2) and (3) is

$$\dot{e}_t = f(x_t) + g(x_t)u_t - \varphi(x_t^d). \tag{4}$$

Humans are capable to store past experiences as previous knowledge to facilitate the decision making. This also represents a skill for the cognitive set $\mathcal{C}$ which will be denoted as memory vector $\mathcal{M}$. So, the robot chooses controls from the actions set $\mathcal{A}$ according to the following decision making control

$$u_t = h(e_t; PE, \mathcal{M}) = h(x_t, x_t^d; PE, \mathcal{M}). \tag{5}$$

Notice that the decision making control depends on the states provided by the state-transition functions (2) and (3); the PE condition and experience replay skills. The decision making controller is composed of a feedforward control $u_{1,t}$ and feedback optimal control $u_{2,t}$ as

$$\begin{aligned} u_t &= u_{1,t} + u_{2,t} \\ u_{1,t} &= g^{-1}(x_t)\left(\varphi(x_t^d) - f(x_t)\right). \end{aligned} \tag{6}$$

The closed-loop dynamics (4) under the decision making control (6) is

$$\dot{e}_t = g(x_t)u_{2,t}. \tag{7}$$

Both $\mathcal{A}$ and $\mathcal{C}$ sets receive the scalar emotion stimuli $\xi_t \in \mathcal{E}$ in accordance to a utility function in terms of the state x_t and control u_t. This paper uses a quadratic function of the form

$$\xi_t = \xi(e_t, u_{2,t}) = e_t^\top S e_t + u_{2,t}^\top R u_{2,t}, \tag{8}$$

where $S = S^\top \geq 0 \in \mathbb{R}^{2n\times 2n}$ and $R = R^\top > 0 \in \mathbb{R}^n$ are weight matrices. The main goal is to find an optimal decision making controller that minimizes the negative emotion stimuli from any initial state x_0. This can be achieved using the concept of value function in reinforcement learning (RL) framework. The infinite-horizon value function [6] of a decision making control is given by

$$V^h(e_t) = \int_t^\infty \xi(e_\tau, u_{2,\tau})d\tau. \tag{9}$$

Taking the time derivative of (9) gives the Hamiltonian of system (7)

$$\begin{aligned} H(e_t, u_{2,t}, \nabla V) &= \dot{V}(e_t) + \xi_t = 0 \\ &= (\nabla V)^\top (g(x_t)u_{2,t}) + \xi_t = 0 \end{aligned} \tag{10}$$

where $\nabla \equiv \frac{\partial}{\partial e_t}$. The optimal value function can be obtained by the Bellman optimality principle as

$$V^*(e_t) = \min_{u_{2,t}\in U} \left(\int_t^\infty \xi(e_\tau, u_{2,\tau})d\tau \right), \tag{11}$$

which satisfies the Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \min_{v\in U} \left[H(e_t, v, \nabla V^*) \right]. \tag{12}$$

Therefore, the optimal decision making satisfies

$$\begin{aligned} u_{2,t} &= h^*(e_t, PE, \mathcal{M}) \\ &= \arg\min_{v\in U} \left[\xi_t + (\nabla V^*)^\top (g(x_t)v) \right]. \end{aligned} \tag{13}$$

Notice that the PE and $\mathcal{M}$ skills are not explicitly observed in (13) because the cognitive model (2) serves only as previous knowledge and it cannot acquired new valuable information. To see the benefits of the PE and $\mathcal{M}$ skills, we need to give a solution for the optimal value function (11) which can be obtained using reinforcement learning.

IV. Learning Human-Behavior

The cognitive set $\mathcal{C}$ gives numerous models and skills that are useful to get the optimal decision making (13). Human-behavior is supported by many intelligent techniques such as: reinforcement learning [8], [32], deep learning [17], [26], machine learning techniques [21], function approximators [5], [8], and so on; with the aim of providing an ability to learn by interacting with the actions (x_t, u_t) to achieve the control task.

The first cognitive model is given by the state transition function (2) and serves as previous knowledge for the other cognitive models. Humans use relations between concepts and ideas as cognitive model in such a way that it facilitates the development of a task. Neural networks fit this cognitive model by using pre-defined basis functions [14] to estimate a desired performance. In this paper, neural networks are used to estimate the value function (9).

The neural cognitive model is given by a value function approximation (VFA). Consider the following neural approximation [30]

$$V(e_t) = \theta^\top \phi(e_t) + \varepsilon(e_t) \tag{14}$$

where $\phi_t = \phi(e_t) : \mathbb{R}^{2n} \to \mathbb{R}^p$ is the activation function vector with p neurons in the hidden layer, $\theta \in \mathbb{R}^p$ is a weight vector and $\varepsilon_t = \varepsilon(e_t)$ is an approximation error. The gradient of (14) satisfies

$$\nabla V = \nabla \phi^\top(e_t)\theta + \nabla\varepsilon(e_t). \tag{15}$$

Using the neural cognitive model (14) in the Hamiltonian (10) gives

$$H(e_t, u_{2,t}, \theta) = \theta^\top \nabla\phi_t(g(x_t)u_{2,t}) + \xi_t = \varepsilon_{H_t} \tag{16}$$

where the residual error due to the approximation error ε_t is

$$\varepsilon_{H_t} = -\left(\nabla\varepsilon_t\right)^\top (g(x_t)u_{2,t}). \tag{17}$$

The value function approximation is

$$\widehat{V}(e_t) = \widehat{\theta}^\top \phi(e_t) \tag{18}$$

where $\widehat{\theta}$ are estimates of the neural weights θ. The Hamiltonian in terms of (18) is

$$\widehat{H}(e_t, u_{2,t}, \widehat{\theta}_t) = \widehat{\theta}_t^\top \nabla\phi_t(g(x_t)u_{2,t}) + \xi_t = \delta_t \tag{19}$$

where δ_t is the temporal difference (TD) error due to the value function approximation (18) [14]. The difference between (19) and (16) is

$$\delta_t = \widetilde{\theta}_t^\top \nabla\phi_t(g(x_t)u_{2,t}) + \varepsilon_{H_t} \tag{20}$$

where $\widetilde{\theta}_t = \widehat{\theta}_t - \theta$. The main objective of the neural approximator is to minimize the squared TD error [23]

$$E = \frac{1}{2}\delta_t^2$$

such that $\widehat{\theta}_t \to \theta$ and $\delta_t \to \varepsilon_{H_t}$. It is used the normalized gradient descent algorithm proposed in [23] for the neural weights tuning as

$$\dot{\widehat{\theta}}_t = -\alpha \frac{\partial E}{\partial \widehat{\theta}_t} = -\alpha \frac{q_t}{(1 + q_t^\top q_t)^2} \left[q_t^\top \widehat{\theta}_t + \xi_t \right] \tag{21}$$

where $q_t = \nabla\phi(x_t)(g(x_t)u_{2,t})$. This normalized version guarantees boundedness of the activation functions.

Remark 2: The neural cognitive model emulates how humans use different tools to perform the same task. Since it is difficult to obtain the optimal value function (11), then an approximation is used in terms of simpler functions.

The decision making control is obtained by solving the stationary condition $\partial\widehat{H}/\partial u_{2,t} = 0$ under the best neural weights $\widehat{\theta}_t^*$. The optimal decision making control is

$$u_{2,t} = h^*(e_t, \widehat{\theta}_t^*; PE, \mathcal{M}) = \arg\min_{v\in U} \left[\xi_t + \widehat{\theta}_t^{*\top} q_t \right]. \tag{22}$$

The neural cognitive model (21) gives an approximate solution of (11) such that the optimal decision making control (13) is obtained. To achieve this goal, the approximation needs an exploration term which, in this case, is given by a PE exciting condition.

Remark 3: Convergence of the neural-cognitive model weights is achieved under an enough exploration of the action set $\mathcal{A}$. This exploration is satisfied if q_t fulfills a persistently exciting (PE) condition [31]. So, the PE skill is within the neural approximator (21), that is, $q_t = q_t(x_t, u_{2,t}; PE)$.

The neural model (21) updates the weights by considering only one sample [19]. The key idea of the experience replay is to store more samples in a memory vector and add them to the gradient descent update rule [12], such that it minimizes simultaneously the actual TD error δ_t and the TD error of previous samples. The experience replay skill is based on memory, that is, humans store past experiences as knowledge in order to facilitate decision making. Consider that the samples are stored in a memory vector of $\mathcal{M}$ dimension. Denote q_k and ξ_k as samples of q and ξ in time instance t_k, that is,

$$q_k = \nabla\phi(x_{t_k})(g(x_{t_k})u_{t_k}) \tag{23}$$

$$\xi_k = \xi(x_{t_k}, u_{t_k}). \tag{24}$$

The TD error δ_k at time instance t_k is

$$\delta_k = \widehat{\theta}_t^\top q_k + \xi_k. \tag{25}$$

The neural cognitive model works together with the PE and experience replay skills and modifies the gradient descent algorithm as

$$\begin{aligned}\dot{\widehat{\theta}}_t = \; & -\alpha \frac{q_t}{(1+q_t^\top q_t)^2}\left[q_t^\top \widehat{\theta}_t + \xi_t\right] \\ & -\alpha \sum_{k=1}^{\mathcal{M}} \frac{q_k}{(1+q_k^\top q_k)^2}\left[q_k^\top \widehat{\theta}_t + \xi_k\right].\end{aligned} \tag{26}$$

Theorem 1: If the learning input $q_t/(1+q_t^\top q_t)$ and the sequence $q_k/(1+q_k q_k^\top)$ in (26) are PE, then the neural weights $\widehat{\theta}_t$ of the neural-cognition model converges exponentially to a small bounded set; and hence $\widehat{\theta}_t$ remain bounded and are closed to their optimal values.

Proof: Consider that $\varepsilon_{H_t} = 0$. The update (26) can be written in terms of the weights error $\widetilde{\theta}_t$ as

$$\dot{\widetilde{\theta}}_t = -\alpha \left[\frac{q_t q_t^\top}{(1+q_t^\top q_t)^2} + \sum_{k=1}^{\mathcal{M}} \frac{q_k q_k^\top}{(1+q_k^\top q_k)^2}\right]\widetilde{\theta}_t. \tag{27}$$

The term in brackets of (27) is a positive definite matrix and hence it is easy to show that $\dot{\widetilde{\theta}}_t = -A_t\widetilde{\theta}_t$ has an exponential solution of the form $\widetilde{\theta}_t = e^{-A_t(t-t_0)}\widetilde{\theta}_{t_0}$. On the other hand, when $\varepsilon_{H_t} \neq 0$, then the cognitive-neural network has the form of $\dot{\widetilde{\theta}}_t = -A_t\widetilde{\theta}_t + B_t\varepsilon_{H_t}$, for some bounded time-varying matrix B_t. Since the unforced system is exponentially stable, then the forced system trajectories are uniformly ultimate bounded (UUB) [15] and converges exponentially to a small bounded zone if $q_t/(1+q_t q_t^\top)$ and $q_k/(1+q_k q_k^\top)$ are persistently exciting. Subsequently the neural weights remain bounded and are closed to the real weights of the optimal decision making control. ■

Here the experience replay skill improves the exponential convergence in a similar way as human-learning [2], i.e., there are used past experiences as previous knowledge to update the neural weights.

V. Simulation studies

In this section, the performance of the human-behavior learning was assessed using a 2-DOF planar robot [36]. The matrices of the 2-DOF robot were defined as

$$M(q) = \begin{bmatrix}(m_1+m_2)l_1^2 + w_1 + 2w_2 + J_1 & w_1 + w_2 \\ w_1 + w_2 & w_1\end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix}\dfrac{\partial w_2}{\partial q_2}\dot{q}_2 & \dfrac{\partial w_2}{\partial q_2}(\dot{q}_1+\dot{q}_2) \\ -\dfrac{\partial w_2}{\partial q_2}\dot{q}_1 & 0\end{bmatrix}$$

$$G(q) = \begin{bmatrix}(m_1+m_2)gl_1\cos(q_1) + w_3 \\ w_3\end{bmatrix}$$

where q_1 and q_2 define the joint angles of the 2-DOF robot, m_i, J_i and l_i stand for the mass, inertia and length of each link $i = 1, 2$, $g = 9.81$ m/s^2 is the gravity acceleration and $w_1 = m_2 l_2^2 + J_2$, $w_2 = m_2 l_1 l_2 \cos(q_2)$, $w_3 = m_2 g l_2 \cos(q_1 + q_2)$. The links were modeled as thin bars with $J_i = \frac{1}{12}m_i l_i^2$. The robot parameters were $m_1 = m_2 = 0.5$ kg and $l_1 = l_2 = 0.6$ m.

The activation functions were selected as the quadratic vector in the state components as

$$\phi(e_t) = \left[e_1^2 \; e_1e_2 \; e_1e_3 \; e_1e_4 \; e_2^2 \; e_2e_3 \; e_2e_4 \; e_3^2 \; e_3e_4 \; e_4^2\right].$$

So there were 10 neural weights θ. The weight matrices were set as $S = I_{4\times4}$ and $R = I_{2\times2}$. The learning rate was set to $\alpha = 10$. These hyper-parameters were obtained via a grid search until the best performances was achieved. The desired joint space trajectory was

$$q_1^d = \frac{\pi}{3}\sin\left(\frac{\pi}{6}t\right), \quad q_2^d = \frac{\pi}{3}\cos\left(\frac{\pi}{6}t\right),$$

so $x_t^d = [q_1^d,\; q_2^d,\; \dot{q}_1^d,\; \dot{q}_2^d]^\top$. Three different experiments were executed using different memory size, that is, $\mathcal{M} = 0$, $\mathcal{M} = 5$, and $\mathcal{M} = 10$. A PE signal with small amplitude and high frequency was used.

Fig. 2 and Fig. 3 show the tracking results for the proposed cases. All cases achieve a correct realization of the desired trajectory by considering the PE signal, the excitation of the desired trajectory and the experience replay skill. When $\mathcal{M} = 0$, the human-behavior algorithm needs that the PE signal be sufficient rich in order to guarantee convergence of the estimates and hence the learning time increases. When $\mathcal{M} = 5$ and $\mathcal{M} = 10$ the convergence of the solution was improved.

A large memory $\mathcal{M}$ speed up convergence of the human-behavior algorithm to the optimal/near optimal solution. However, fast convergence could imply that the algorithm converges to a local minima, which can be seen as the trade-off between exploration and exploitation in RL algorithms.

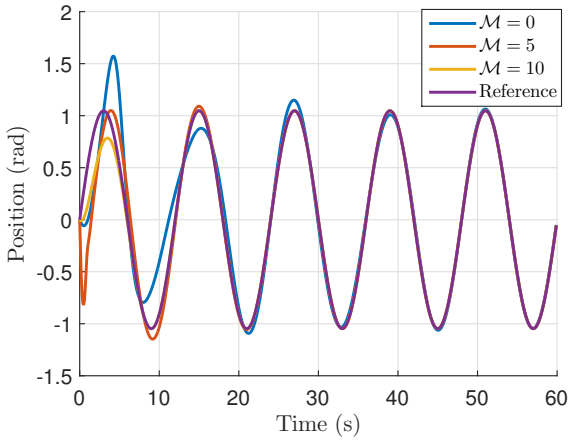


Fig. 2. Joint position q_1

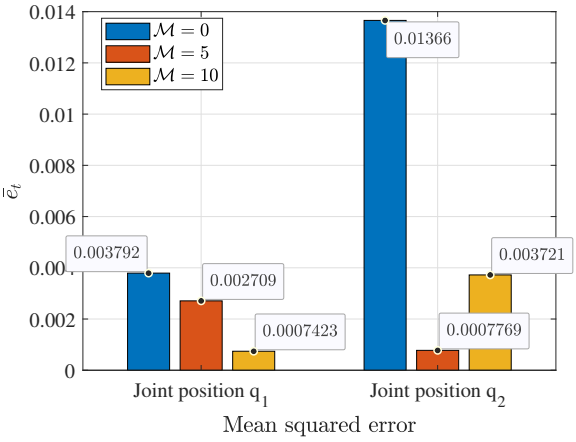


Fig. 4. Mean squared error $\bar{e}_t$

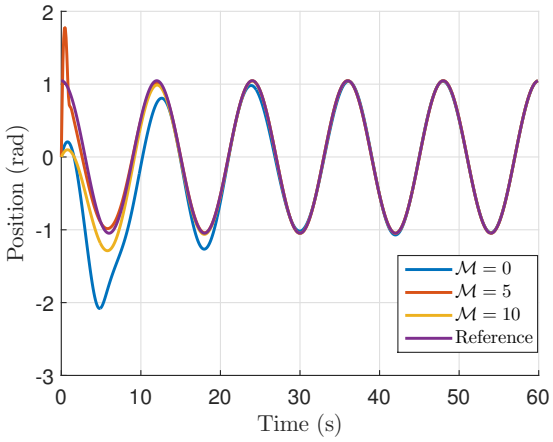


Fig. 3. Joint position q_2

In order to see this issue, the mean squared error (MSE) $\bar{e}_t = \sum_{t=0}^{N} e_t^2$, was used as performance metric. Fig. 4 shows the bar plot of the MSE. The plot show that the human-behavior algorithm without experience replay skill has large MSE since it converges in more time steps. On the other hand, when $\mathcal{M} = 10$ shows the better MSE performance for the joint position q_1 but not for joint position q_2 because it finds a local minima. So for control design, there is a trade-off between the size of the memory in order to balance the exploration and exploitation of the knowledge that was acquired.

Fig. 5 shows the phase diagram between the joint position states x_1 and x_2 of the 2-DOF robot. Here is more evident the convergence improvement of the human-learning method using the experience replay skill. Since the proposed method needs bounded control input, some authors use an actor-critic structure [12], [23] to overcome this issue. However it increases the complexity of the neural cognitive model by adding other neural network and extra hyper-parameters, which is relatively undesirable due to the computational resources. A large enough weight matrix R helps to normalize the input dynamics $g(x_t)$ such that the control input remain bounded.

VI. Conclusions

In this paper, the control of nonlinear systems using a human-behavior learning is presented. The proposed learning method finds online the solution of a HJB equation and the optimal decision making method. The effectiveness of the proposed human-behavior learning lies in the use of different cognitive models and skills which help to find the best decision making control in less time. Three cognitive models composed by the nonlinear dynamics, desired reference and a neural network model; and two skills given by a persistent exciting condition and a memory vector; are used to facilitate obtaining the solution of the optimization problem.

The use of experience replay skill improves the convergence of the neural network algorithm by taking into account more experiences at the update rule. Simulations studies verify our approach with satisfactory results.

References

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

[2] V. Mnih, K. Kabukcuoglu, S. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattle, A. Sadik, H. Antonouglu, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control thorugh deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015.

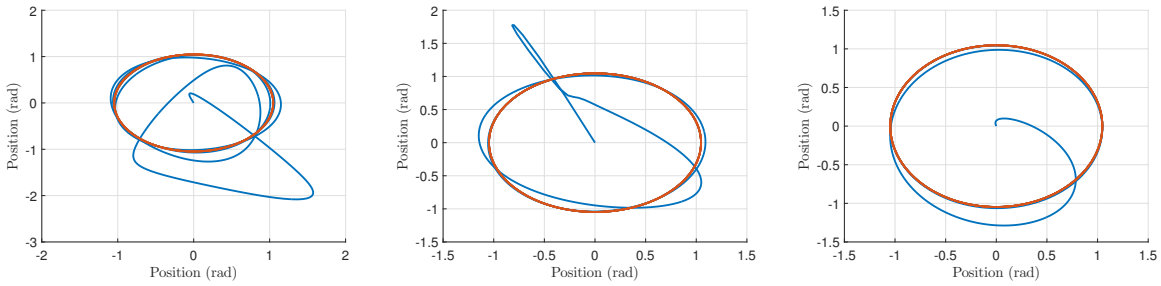
[3] J. W. Kim, B. J. Park, H. Yoo, J. H. Lee, and J. M. Lee, "Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear system," *9th IFAC Symposium on Robust Control Design ROCOND*, vol. 51, no. 25, pp. 257–262, 2018.

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

[5] A. Perrusquía, W. Yu, and A. Soria, "Large space dimension reinforcement learning for robot position/force discrete control," *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2019, paris, France.

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

[7] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, pp. 1167–1175, 2014.



(a) $\mathcal{M} = 0$ (b) $\mathcal{M} = 5$ (c) $\mathcal{M} = 10$

Fig. 5. Phase diagram trajectories

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

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

[10] D. Zha, K.-H. Lai, K. Zhou, and X. Hu, "Experience replay optimization," *Proceedingds of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.

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

[12] H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, pp. 193–202, 2014.

[13] B. Kiumarsi, G. V. Kyriakos, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

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

[15] W. Yu and A. Perrusquía, "Simplified stable admittance control using end-effector orientations," *International Journal of Social Robotics*, 2019.

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

[17] E. de la Rosa and W. Yu, "Randomized algorithms for nonlinear system identification with deep learning modification," *Information Sciences*, vol. 364-365, pp. 197–212, 2016.

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

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

[20] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, no. 1, pp. 219–245, 2000.

[21] A. Perrusquía and W. Yu, "Robust control under worst-case uncertianty for unknown nonlinear systems using modified reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 30, no. 7, pp. 2920–2936, 2020.

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

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

[24] A. Perrusquía, W. Yu, and X. Li, "Multi-agent reinforcement learning for redundant robot control in task-space," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 1, pp. 231–241, 2021.

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

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

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

[28] A. Perrusquía and W. Yu, "Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview," *Neurocomputing*, vol. 438, pp. 145–154, 2021.

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

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

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

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

[33] S. Singh and S. Sutton, Richard, "Reinforcement learning with replacing elegibility traces," *Machine learning*, vol. 22, pp. 123–158, 1996.

[34] A. Perrusquía and W. Yu, "Human-in-the-loop control using Euler angles," *Journal of Intelligent & Robotic Systems*, vol. 97, pp. 271–285, 2020.

[35] W. Yu, "Multiple recurrent neural networks for stable adaptive learning," *Neurocomputing*, vol. 70, pp. 430–444, 2006.

[36] A. Perrusquía, J. A. Flores-Campos, and C. R. Torres-San-Miguel, "A novel tuning method of PD with gravity compensation controller for robot manipulators," *IEEE Access*, vol. 8, pp. 114 773–114 783, 2020.

2022-02-01

Human-behavior learning for infinite-horizon optimal tracking problems of robot manipulators

Perrusquía, Adolfo

IEEE

Perrusquía A, Yu W. (2022) Human-behavior learning for infinite-horizon optimal tracking problems of robot manipulators. In: 2021 60th IEEE Conference on Decision and Control (CDC), 14-17 December 2021, Austin, Texas, USA

https://doi.org/10.1109/CDC45484.2021.9683719