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

Adolfo Perrusquia, 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 carried 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, especially in video games applications [2]. These kinds 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 \( \varepsilon \)-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 cognition 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 define 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 \( \varepsilon \)-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.
Emotions $E$ 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).

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

- The actions set $A$ contains two subspaces: the set of states $X$ and the set of controls $U$, so $A = X \cup U$. $X$ 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 $h : X \rightarrow U$, which describes which controls have to be applied in certain states in accordance to the cognition set $C$ and the received emotion $\xi$.

- The cognition set $C$ 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 $f : X \times U \rightarrow X$ which describes how the state $x \in X$ changes as a result of applying a control $u \in U$. The skills $s_i : X \times U \rightarrow X$ are methods that the human uses to facilitate the execution of a task in accordance to its experience or a stimuli.

- The emotion set $E$ provides feedback stimuli about the interaction between $A$ and $C$. This feedback can be interpreted as an application denoted by $\xi : X \times U \rightarrow \mathbb{R}$. 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

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

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

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

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 $x_{1,t} = q_{1,t}, \ldots, x_{n,t} = q_{n,t}, \ x_{n+1,t} = \dot{q}_{1,t}, \ldots, x_{2n,t} = \dot{q}_{n,t}$. Define $x_t = [x_{1,t}, \ldots, x_{n,t}, x_{n+1,t}, \ldots, x_{2n,t}]^T$ and $u_t = \tau$.

The robot is regarded as the acting human. The actions set $A$ is given by all robot states $x_t \in A \subset \mathbb{R}^{2n}$ (joint positions and joint velocities) and the torque inputs $u_t \in A \subset \mathbb{R}^n$. 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 $C$.

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 $C$ and can be designed with any exploration technique, for example, $\epsilon$-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 $x^d_t \in \mathbb{R}^n$ [35]. The desired trajectory dynamics is regarded as a cognitive model for the human-behavior learning, which satisfies the following nonlinear reference model

$$
\dot{x}^d_t = \varphi(x^d_t),
$$

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

$$
\dot{e}_t = f(x_t) + g(x_t)u_t - \varphi(x^d_t).
$$

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

$$
u_t = h(e_t; PE, M) = h(x_t, x^d_t; PE, M).
$$

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

$$
\begin{align*}
u_t & = u_{1,t} + u_{2,t} \\
u_{1,t} & = g^{-1}(x_t) \left( \varphi(x^d_t) - f(x_t) \right).
\end{align*}
$$
The closed-loop dynamics (4) under the decision making control (6) is
\[ \dot{x}_t = g(x_t)u_{2,t}. \] (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
\[ \xi_t = x_t^TS_t + u_{2,t}^TRu_{2,t}, \] (8)
where $S = S^T \geq 0 \in \mathbb{R}^{2n \times 2n}$ and $R = R^T > 0 \in \mathbb{R}^n$ 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
\[ V^b(e_t) = \int_t^\infty \xi(e_\tau, u_{2,\tau})d\tau. \] (9)

Taking the time derivative of (9) gives the Hamiltonian of system (7)
\[ H(e_t, u_{2,t}, \nabla V) = \dot{V}(e_t) + \xi_t = 0 = (\nabla V)^T(g(x_t)u_{2,t}) + \xi_t = 0 \] (10)
where $\nabla \equiv \frac{\partial}{\partial e_t}$. The optimal value function can be obtained by the Bellman optimality principle as
\[ V^*(e_t) = \min_{u_{2,t} \in U} \left( \int_t^\infty \xi(e_\tau, u_{2,\tau})d\tau \right), \] (11)
which satisfies the Hamilton-Jacobi-Bellman (HJB) equation
\[ 0 = \min_{e_t \in U} [H(e_t, v, \nabla V^*)]. \] (12)

Therefore, the optimal decision making satisfies
\[ u_{2,t} = h^*(e_t, PE, \mathcal{M}) = \arg \min_{v \in U} \left[ \xi_t + (\nabla V^*)^T(g(x_t)v) \right]. \] (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]
\[ V(e_t) = \theta^T \phi(e_t) + \varepsilon(e_t) \] (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
\[ \nabla V = \nabla \phi^T(e_t)\dot{\theta} + \nabla \varepsilon(e_t). \] (15)

Using the neural cognitive model (14) in the Hamiltonian (10) gives
\[ H(e_t, u_{2,t}, \theta) = \theta^T \nabla \phi_t(g(x_t)u_{2,t}) + \xi_t = \varepsilon_{H_t}, \] (16)
where the residual error due to the approximation error $\varepsilon_t$ is
\[ \varepsilon_{H_t} = - (\nabla \varepsilon_t)^T(g(x_t)u_{2,t}). \] (17)
The value function approximation is
\[ \hat{V}(e_t) = \hat{\theta}^T \phi(e_t) \] (18)
where $\hat{\theta}$ are estimates of the neural weights $\theta$. The Hamiltonian in terms of (18) is
\[ \hat{H}(e_t, u_{2,t}, \hat{\theta}_t) = \hat{\theta}_t^T \nabla \phi_t(g(x_t)u_{2,t}) + \xi_t = \delta_t \] (19)
where $\delta_t$ is the temporal difference (TD) error due to the value function approximation (18) [14]. The difference between (19) and (16) is
\[ \delta_t = \theta_t^T \nabla \phi_t(g(x_t)u_{2,t}) + \varepsilon_{H_t}, \] (20)
where $\hat{\theta}_t = \theta_t - \theta$. The main objective of the neural approximator is to minimize the squared TD error [23]
\[ E = \frac{1}{2} \delta_t^2 \]
such that $\hat{\theta}_t \to \theta$ and $\delta_t \to \varepsilon_{H_t}$. It is used the normalized gradient descent algorithm proposed in [23] for the neural weights tuning as
\[ \hat{\theta}_t = -\alpha \frac{\partial E}{\partial \hat{\theta}_t} = -\alpha \frac{q_t}{(1 + q_t^2)} [q_t^T \hat{\theta}_t + \xi_t]. \] (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 $\partial H/\partial u_{2,t} = 0$ under the best neural weights $\hat{\theta}_t^*$. The optimal decision making control is
\[ u_{2,t} = h^*(e_t, \hat{\theta}_t^*; PE, \mathcal{M}) = \arg \min_{v \in U} \left[ \xi_t + \hat{\theta}_t^* q_t \right]. \] (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 $A$. 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 $\delta_t$ 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 $M$ dimension. Denote $q_k$ and $\xi_k$ as samples of $q$ and $\xi$ in time instance $t_k$, that is,

$$q_k = \nabla_\phi(x_{t_k})(g(x_{t_k})u_{t_k})$$

(23)

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

(24)

The TD error $\delta_k$ at time instance $t_k$ is

$$\delta_k = \tilde{\theta}_t^T q_k + \xi_k.$$

(25)

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

$$\dot{\tilde{\theta}}_t = -\alpha \frac{q_t}{1 + q_t^T q_t} \left[ q_t^T \tilde{\theta}_t + \xi_t \right]$$

$$-\alpha \sum_{k=1}^{M} \frac{q_k}{1 + q_k^T q_k} \left[ q_k^T \tilde{\theta}_t + \xi_k \right].$$

(26)

Theorem 1: If the learning input $q_k / (1 + q_k^T q_k)$ and the sequence $q_k / (1 + q_k^T q_k)$ in (26) are PE, then the neural weights $\tilde{\theta}_t$ of the neural-cognition model converges exponentially to a small bounded set; and hence $\tilde{\theta}_t$ 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 as

$$\dot{\tilde{\theta}}_t = -\alpha \left[ \frac{q_t q_t^T}{1 + q_t^T q_t} \right] \tilde{\theta}_t - \alpha \sum_{k=1}^{M} \frac{q_k q_k^T}{1 + q_k^T q_k} \tilde{\theta}_t.$$

(27)

The term in brackets of (27) is a positive definite matrix and hence it is easy to show that $\dot{\tilde{\theta}}_t = -A_t \tilde{\theta}_t$ has an exponential solution of the form $\tilde{\theta}_t = e^{-A_t (t-t_0)} \tilde{\theta}_{t_0}$. On the other hand, when $\varepsilon_{H_t} \neq 0$, then the cognitive-neural network has the form $\dot{\tilde{\theta}}_t = -A_t \tilde{\theta}_t + B_t \varepsilon_{H_t}$, 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 $q_t / (1 + q_t^T q_t)$ and $q_k / (1 + q_k^T q_k)$ 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

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

$$C(q, q') = \begin{bmatrix} \frac{\partial w_2}{\partial q_2} & \frac{\partial w_2}{\partial q_2} (q_1 + q_2) \\ -\frac{\partial w_2}{\partial q_1} & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} (m_1 + m_2)g_1 \cos(q_1) + w_3 \\ w_3 \end{bmatrix}$$

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 $i = 1, 2$, $g = 9.81 \text{ m/s}^2$ 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 $J_1 = \frac{1}{12} m_i l_i^2$. The robot parameters were $m_1 = m_2 = 0.5 \text{ kg}$ and $l_1 = l_2 = 0.6 \text{ m}$.

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

$$\phi(e_t) = [e_1^2 e_1 e_2 e_1 e_3 e_1 e_4 e_2^2 e_2 e_3 e_2 e_4 e_3^2 e_3 e_4 e_4^2].$$

So there were 10 neural weights $\theta$. The weight matrices were set as $S = I_{4 \times 4}$ and $R = I_{2 \times 2}$. 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

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

so $x_t^d = [q_1^d q_2^d q_1^d q_2^d]^T$. Three different experiments were executed using different memory size, that is, $M = 0$, $M = 5$, and $M = 10$. 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 $M = 0$, 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 $M = 5$ and $M = 10$ the convergence of the solution was improved.

A large memory $M$ 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.
In order to see this issue, the mean squared error (MSE) $\epsilon_t = \sum_{t=0}^{N} \epsilon_t^2$, 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 $M = 10$ 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

**Fig. 5. Phase diagram trajectories**

- **(a) M = 0**
- **(b) M = 5**
- **(c) M = 10**

---


2022-02-01

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

Perrusquía, Adolfo

IEEE

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

Downloaded from Cranfield Library Services E-Repository