A Deep Reinforcement Learning Approach for Robust Control in Robotic Applications

Camilo Andrés Manrique Escobar
MEID4 Academic Spin-Off, University of Salerno
Fisciano (SA), Italy
camilo.manrique@outlook.com

Carmine Maria Pappalardo
DIIN, University of Salerno
Fisciano (SA), Italy
cpappalardo@unisa.it

Domenico Guida
DIIN, University of Salerno
Fisciano (SA), Italy
guida@unisa.it

***Abstract*—A methodology for developing robust control systems using Deep Reinforcement Learning (DRL) is proposed in this work. To this end, a numerical simulation of the cart-pole swing-up problem is used as a case study. The agent is trained with the Deep Deterministic Policy Gradient (DDPG). Then, its sensibility is evaluated by modifying the parameters of the physical system. Subsequently, the presence of dry friction between the cart and the horizontal plane is considered for testing the robustness of the agent after employing the post-training method proposed. The numerical results found demonstrates the broad potential of the methodology considered in this paper.**

***Index Terms*—Deep reinforcement learning, DDPG, robust control, robotics, mechatronics**

I. Introduction

The development of controllers for nonlinear systems is extremely challenging, time-consuming, and in many cases, an infeasible task. The traditional engineering approach consists of analytically deriving the system governing equations and manually adjusting the parameters of the control system to fit some measured physical parameters. Advanced expertise in applied mathematics, dynamical systems theory, computational methods, mathematical modeling, optimization frameworks, and operator-assisted algorithmic tuning of control parameters is then required [1]. Unfortunately, this approach results to be restrictive for highly uncertain or difficult to model systems. Because of the reality gap caused by the model bias, this often yields an unfeasible control system. Deep Reinforcement Learning (DRL) is a data-driven approach useful for the development of control systems. Given the generalization capability of Artificial Neural Networks (ANN), this framework has the potential to overcome the problems mentioned before. By following the methodology proposed in this work, the use of ANN also allows for controlling systems with parameters uncertainty.

II. Methodology

The reinforcement learning framework consists of learning based on interaction instead of mathematically modeling the system to be controlled. For any time t, the agent collects an observation $\mathbf{s}$ and generates action $\mathbf{a}$. The state of the environment changes to $\mathbf{s}'$ and produces a scalar reward signal r. This cycle is repeated until the final instant of time t_f is reached. Recently, RL combined with state-of-art deep learning techniques has attracted much attention to robotics [2]. In this context, the agent is the controller, and the robot and its surroundings are the environment [3]. The latter provides the agent with the instantaneous state and reward. The goal of the agent is to find the policy that optimizes the cumulative reward obtained during a complete episode. The present work uses the DDPG [4] algorithm to train an agent with an actor-critic architecture. That is, the agent is composed of two neural networks, the actor $\boldsymbol{\pi}_\phi$, and the critic $\mathbf{Q}_\theta$. The task of $\boldsymbol{\pi}_\phi$ is to map any $\mathbf{S}$ to an action $\mathbf{a}$. On the other hand, the critic addresses to learn the optimal Q-value function of the environment. That is, a function predicting the cumulative reward of the agent for a given policy, $\mathbf{s}$, and $\mathbf{a}$. This is done with a supervised learning approach, employing the dataset of experiences generated by the interaction of the agent and the environment. Each of its entries has the form $[\mathbf{s}, \mathbf{a}, r, \mathbf{s}', \mathbf{a}']$. The loss function L used is the mean squared Bellman error.

$$L = \frac{1}{D}\sum_{i=1}^{D}\left(\mathbf{y}_i - \mathbf{Q}_\theta\left(\mathbf{s}_i, \mathbf{a}_i\right)\right)^2 \tag{1}$$

where D is the size of a randomly sampled mini-batch of the dataset, and y is the target value function defined as

$$y_i = r_i + \gamma \mathbf{Q}_{\theta'}\left(\mathbf{s}_i', \pi_{\phi'}\left(\mathbf{s}_i'\right)\right) \tag{2}$$

Then, after updating the critic $\mathbf{Q}_\theta$, this is differentiated with respect to ϕ, the parameters of $\boldsymbol{\pi}_\phi$ in order to compute the optimal gradient for the parameters $\boldsymbol{\nabla}_{\pi_\phi}$.

$$\boldsymbol{\nabla}_{\pi_\phi} = \frac{1}{D}\sum_{i=1}^{D}\boldsymbol{\nabla}_{\pi_\phi}\mathbf{Q}_\theta\left(\mathbf{s}_i,\, \pi_\phi\left(\mathbf{s}_i\right)\right)\nabla_\phi \boldsymbol{\pi}_\phi\left(\mathbf{s}_i\right) \tag{3}$$

Both $\mathbf{Q}_\theta$ and $\boldsymbol{\pi}_\phi$ are updated by Polyak averaging with the precedent parameter values. The scope of this work is to test the robustness of the trained actor neural network $\boldsymbol{\pi}_\phi$ when modifying the environment properties. Then, to improve the robustness, a post-training of the agent is proposed and tested. That is, a subsequent training of $\boldsymbol{\pi}_\phi$ to expand the network optimal behavior space is performed.

III. Numerical Results

The environment employed to test the methodology is the cart-pole shown in Fig. 1. The goal of the agent is to find

Identify applicable funding agency here. If none, delete this.

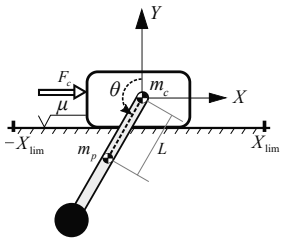

Fig. 1. The cart-pole environment.

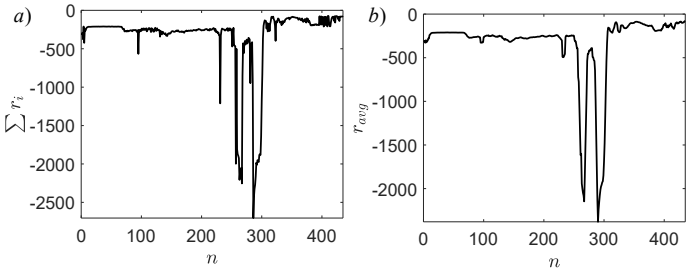

Fig. 2. Learning curves of the agent; a) cumulative reward per episode of training; and b) The five episodes moving window average cumulative reward.

the optimal policy capable of swinging up the pole in a given interval of time by applying a lateral force F_c to the cart. The system is trained with the fixed values $\mu = 0$, $m_c = 1(kg)$, $m_p = 1(kg)$, and $L_p = 0.1(m)$. The force F_c is constraint to be in the interval $[-20,\ 20]\ (N)$. The optimal policy is found after 435 episodes of training, and the respective learning curve is shown in Fig. 2. The response of the trained agent in the modified environments is shown in Fig. 3. The agent can perform the swing-up task in the presence of equal increment in the masses and length up to 25% and equal decrement up to 35%, proving to be robust as is. The agent can perform the swing-up task in the presence of joint increment in the masses and length up to 25% and a joint decrement up to 35%, proving to be robust as well. However, dry friction $(\mu < 0)$ is deleterious for the performance, resulting in an incapable agent as shown in Case 4 of Fig. 4. We call the initial

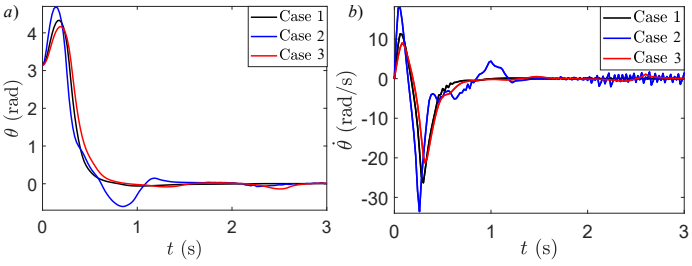

Fig. 3. a) Pole angular position during the swing-up task; and b) Pole angular velocity during the swing-up task. **Case 1**: values equal to the training; **Case 2**: 25% increment in all the parameter values except friction; **Case 3**: 35% decrement in all the parameters except friction.

agent, Agent A. Then, including dry friction, a second training process with the environment is carried out. The resulting agent is called Agent B. Table I reports the behavior of agents A and B in different configurations of the environment. The higher the cumulative reward $\sum r$ for the agent, the better its performance. Fig. 4 shows the angular position and angular velocity of the pole during an episode interaction. It can be seen that agent B performs the task under both environments with and without dry friction. This proves the approach to be feasible to develop robust control systems.

TABLE I
CASES OF ANALYSIS FOR THE PERFORMANCE OF AGENTS A AND B IN PRESENCE OF DRY FRICTION.

Case	Agent	μ	m_c (kg)	m_p (kg)	L_p 1 (m)	$\sum r$
4	A	0.3	1	1	0.1	-341.48
5	B	0.3	1	1	0.1	-74.71
6	B	0	1	1	0.1	-81.51
7	B	0.8	1	1	0.1	-78.55

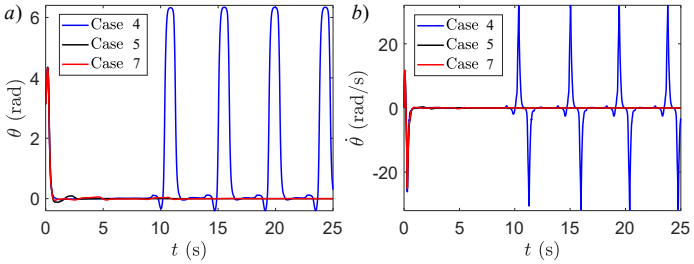

Fig. 4. Pole angular position and angular velocity for cases 4, 5, and 7; a) Pole angular position during the swing-up task; and b) Pole angular velocity during the swing-up task.

IV. CONCLUSIONS

In this work, the sensitivity of a controller developed with the DDPG algorithm was studied. Additionally, a post-training methodology was proposed to increase the robustness of the said control system. The cart-pole system was considered as the case study. The sensitivity analysis shows that the controller can perform the swing-up task for magnitute modifications of physical parameters as high as 35%. The presence of dry friction proves to be detrimental to the controller. However, the proposed methodology results in a new controller capable of operating effectively in environments with or without dry friction. Therefore, the present methodology is promising for future developments in robotics and mechatronics research.

REFERENCES

[1] J. Morimoto and K. Doya, "Robust Reinforcement Learning," Neural Comput., vol. 17, no. 2, pp. 335–359, Feb. 2005, doi: 10.1162/0899766053011528.

[2] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An Introduction to Deep Reinforcement Learning," Found. Trends® Mach. Learn., vol. 11, no. 3–4, pp. 219–354, 2018, doi: 10.1561/2200000071.

[3] P. Wawrzyński, "Control Policy with Autocorrelated Noise in Reinforcement Learning for Robotics," Int. J. Mach. Learn. Comput., vol. 5, no. 2, pp. 91–95, Apr. 2015, doi: 10.7763/IJMLC.2015.V5.489.

[4] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," Sep. 2015, ArXiv:1509.02971 [Cs, Stat].