Does Reinforcement Learning outperform PID in the control of FES-induced elbow flex-extension?

1st Davide Di Febbo
NearLab, Deib
Politecnico di Milano
Milan, Italy

2nd Emilia Ambrosini
NearLab, Deib
Politecnico di Milano
Milan, Italy

3rd Matteo Pirotta
SequeL Team
Inria
Lille - Nord Europe, France

4th Eric Rojas
NearLab
Politecnico di Milano
Milan, Italy

5th Marcello Restelli
AIRLab, Deib
Politecnico di Milano
Milan, Italy

6th Alessandra L. G. Pedrocchi
NearLab, Deib
Politecnico di Milano
Milan, Italy

7th Simona Ferrante
NearLab, Deib
Politecnico di Milano
Milan, Italy

***Abstract*—Functional electrical stimulation (FES) is an effective technology in post-stroke rehabilitation of the upper limbs. Because of the complexity of the system, traditional linear controllers are still far to drive accurate and natural movements. In this work, we apply reinforcement learning (RL) to design a nonlinear controller for an upper limb FES system combined with a passive exoskeleton. RL methods learn by interacting with the environment and, to efficiently use the collected data, we simulated large numbers of experience episodes through artificial neural network (ANN) models of the electrically stimulated arm muscles. The performance of the novel control solution was compared to a PID controller on five healthy subjects during planar reaching tasks. Both controllers correctly drove the arm at the target position, with a mean absolute error $< 1^{\circ}$. The RL control significantly outperformed the PID in terms of setting time, position accuracy and smoothness. Future trials are needed to confirm these promising results.**

***Index Terms*—Functional Electrical Stimulation; Reinforcement Learning; Hybrid Robotic Systems; Neuroprosthetics.**

I. Introduction

Functional Electrical Stimulation (FES) is a technology used to artificially activate hemiparetic muscles to produce functional movements. A recent Cochrane review reported evidence that goal-oriented repetitive movement therapy improves arm and hand functions after stroke [1]. In the past few years, the combined use of FES and robotic technologies have been proposed to improve the rehabilitation outcomes [2], [3]. Passive exoskeletons for upper limbs are well indicated to support patients as they remove the gravitational load but give the subject the role to execute the tasks. On the other hand, it has been recently shown that FES has a positive effect on upper limb activity compared with both no intervention and training alone [4].

However, considerable difficulties are encountered in the development of control systems for hybrid robotic systems since the dynamics of the electrically stimulated human arm is highly nonlinear and its physiological properties are not completely known [5]. Classical controllers, based on linear assumptions, are reliable, but they have limited performance because they rely on the model accuracy [6]. Nonlinear controllers can perform better [7], but they become time-consuming to fine-tune on different subjects. Moreover, they do not take into account the time-variations of the human arm system due, for example, to muscle fatigue or muscle strengthening [8].

Reinforcement Learning (RL) has been recently proposed to control a simulated upper limb FES system [9]. RL uses artificial intelligence methods to make an agent learn the best way to act on a system, namely a policy, from experience collected by interacting with it. RL represents a nonlinear and adaptive control solution and it does not require prior knowledge about the system. However, large numbers of experience episodes are often needed to achieve good performance. Thus, RL control algorithms are usually trained in a simulated environment, before being applied to the real system.

In this work, we investigated the feasibility of RL to control an upper limb FES system combined with a passive robotic device for weight relief in a real environment. To reach this aim, we defined a simple control problem consisting of planar elbow flex-extension movements. We chose the Proximal Policy Optimization (PPO) algorithm [10], which has been shown to outperform several state-of-the-art algorithms in continuous control tasks. To overcome the issue of continued interaction with the system, we trained the algorithm in a simulated environment consisting in a subject-specific artificial neural network (ANN) model of the electrically stimulated human arm. Finally, the performance of the PPO algorithm was compared to that of a PID controller, which was chosen because it represents the closed-loop system most reliable and used in FES applications.

II. Methods

A. Apparatus

We used a lightweight passive exoskeleton to support the right arm, characterized by 3 degrees of freedom (DoF): elbow flex-extension, shoulder rotation in the frontal plane and shoulder elevation, see Fig. 1. Each DoFs can be independently locked by activating electromagnetic brakes.

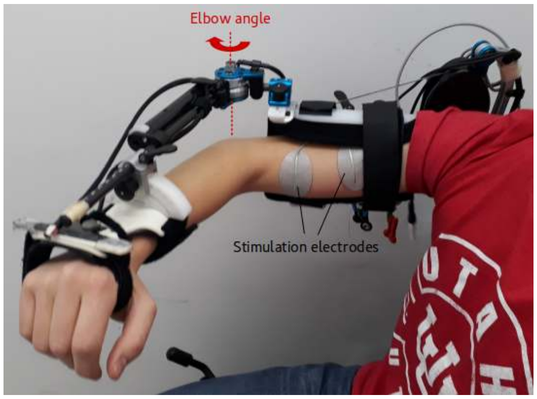

Fig. 1. The right arm passive exoskeleton.

The elbow angular position was measured by a goniometric sensor (Vert-X 13 E, ConTelec AG) embedded in the joint. The gravity compensation module consisted of a carbon fiber tube with springs inside, whose pretension was adjustable. A current-controlled stimulator (RehaMovePro®, Hasomed GmbH), which provided biphasic pulses through surface self-adhesive electrodes (Pals®) was used. Two stimulation channels were simultaneously connected: one to the biceps and one to the triceps muscle. The exoskeleton and the stimulator were controlled by an embedded processor (BeagleBoneBlack™) in which the I/O communication was set at 25 Hz.

B. Selection of the Motor Tasks

In our control problem, we considered target-reaching tasks in the horizontal plane. The elbow angle $\phi(\cdot)$ was allowed to move from 50° (complete flexion) to 180° (complete extension). Starting from an initial angle ϕ_s, identified as the subjects' relaxed arm position, the control system had to properly stimulate the two arm muscles to reach a prescribed target angle ϕ^t. For each subject, we defined two flexion or two extension tasks. The target angles of the flexion tasks were selected in the range $(50°, \phi_s)$, while those of the extension tasks, in the range $(\phi_s, 180°)$. We referred to the target closer to the initial angle as "low target" ϕ^t_l, and to the other one as "high target" ϕ^t_h.

C. RL Formalization

We start formalizing the mathematical framework used to describe the system and the optimal control problem. A sequential decision-making problem can be formalized as a Markov Decision Process (MDP) $M = \{\}$ where S is the continuous state space, A is the continuous action space, P is a Markovian transition kernel where $P(\boldsymbol{s}'|\boldsymbol{s}, a)$ defines the transition density from state $\boldsymbol{s}$ to $\boldsymbol{s}'$ under action a, R is the reward function $r(\boldsymbol{s}, a) = \mathbb{E}[r|\boldsymbol{s}, a]$, $\gamma \in (0, 1)$ is the discount factor and μ is the initial state distribution. The agents behaviour is modelled as a policy π, where $\pi(\cdot|\boldsymbol{s})$ is the density distribution over A in state $\boldsymbol{s}$. The goal of the agent is to find a policy $\pi(a|\boldsymbol{s})$ which maximizes the sum of collected discounted rewards:

$$J(\pi) = \mathbb{E}_\pi\left[\sum_{k=0}^{+\infty} \gamma^k r_k \middle| \boldsymbol{s}_0 \sim \mu, M\right]. \quad (1)$$

We consider episodic MDPs with effective horizon I. In this setting, we can limit our attention to episodes of length I. An episode is a sequence of states, actions and rewards $(\boldsymbol{s}_0, a_0, r_0, \boldsymbol{s}_1, a_1, r_1, \ldots, \boldsymbol{s}_{I-1}, a_{I-1}, r_{I-1})$ observed by following a policy (i.e., $a_i \sim \pi(\cdot|\boldsymbol{s}_i)$), where $s_0 \sim \mu$. The reward is the scalar value provided by the environment which represents the immediate utility of executing the action a_i in the state $\boldsymbol{s}_i$.

Our MDP. In the considered problem, at each time step i, the system state is described as $\boldsymbol{s}_i = [\phi_i, \dot{\phi}_i, \ddot{\phi}_i]' \in \mathbb{R}^3$, where ϕ_i is the elbow angular position, $\dot{\phi}_i$ is the instantaneous angular velocity and $\ddot{\phi}_i$ is the instantaneous angular acceleration. The electrical currents of the two stimulation channels were the inputs and, in order to reduce the complexity of the control problem, we chose to modulate the stimulation frequency rather than the current or the pulse width. Avoiding also the co-contraction of the biceps and the triceps muscles, we defined a scalar control variable, a, which encoded, at each time sample, the action of sending a single stimulation pulse to biceps muscles ($a_i = 1$), to the triceps ($a_i = 2$) or to none of them ($a_i = 0$).

The reward function expresses the overall goal and provides indications to the agent on how to achieve the task. We designed the reward as follows:

$$r(s_i, a_i, s_{i+1}) = -(\phi^t - \phi_{i+1})^2 - \alpha \cdot \ddot{\phi}_{i+1}, \quad (2)$$

where α is a scaling parameter. The effect of such reward function was to penalize the distance from the target and the instantaneous acceleration with the aim of promoting smoother movements.
In our experiments, we set I equal to 50 time instants, γ equal to 0.99 and α equal to 10.

1) RL Solution with the PPO Algorithm: We decided to face the continuous RL problem by using Proximal Policy Optimization (PPO) algorithm. This method is a policy gradient approach that constrains the policy update to avoid disruptive steps. This surrogate objective represents a lower bound to the policy performance. PPO works by optimizing a surrogate loss based on standard policy performance measure (see Eq. 1) by stochastic gradient ascent. The policy $\pi_{\boldsymbol{\theta}}$ is encoded through a parametric representation (e.g., ANN) with parameters $\boldsymbol{\theta}$. Defining the probability ratio as $\rho_i(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_i|\boldsymbol{s}_i)}{\pi_{\boldsymbol{\theta}_{old}}(a_i|\boldsymbol{s}_i)}$, where $\boldsymbol{\theta}_{old}$ is the vector of policy parameters before the update, a surrogate objective is defined as:

$$L(\boldsymbol{\theta}) = \hat{\mathbb{E}}_i[\rho_i(\boldsymbol{\theta})\hat{A}_i], \quad (3)$$

where the expectation $\hat{\mathbb{E}}_i[\cdot]$ indicates the empirical average over a finite batch of samples and $\hat{A}_i$ is an estimator of the advantage function [12]. Starting from Eq. 3, PPO modifies the objective to prevent steps moving policy $\pi_{\boldsymbol{\theta}}$ far away from $\pi_{\boldsymbol{\theta}_{old}}$. The PPO objective is as follows

$$L^{\text{PPO}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_i \left[\min \left\{ \rho_i(\boldsymbol{\theta}) \hat{A}_i, \eta_i \right\} \right]$$
$$\eta_i := clip(\rho_i(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon) \cdot \hat{A}_i$$

The term η_i prevents the update to move too far away from the previous policy (i.e., $\rho_i(\boldsymbol{\theta}) = 1$) by constraining ρ_i in the closed interval $[1-\epsilon, 1+\epsilon]$.

We used the PPO implementation in rllab [11].[1] We choose a multi-layer perceptron with two hidden layers of 26 hyperbolic tangent (tanh) neurons each and a softmax output layer as function estimator. This structure defines the policy distribution over the three discrete actions ($a \in \{0, 1, 2\}$). We finally set the max path length parameter equal to the episode duration I, the batch size equal to $I \cdot 100$ and the total number of iteration equal to 150. The parameter ϵ was set to 0.2.

2) The ANN Model of the Human Arm: As explained above, the algorithm needs to interact with the system to collect samples of the current policy $\pi_{\boldsymbol{\theta}}$. As mentioned in the introduction, we want to avoid to interact with the real system at each update, and thus we rely on a parametric model of the system. We decided to model the subject's arm response to the electrical stimulation using a feedforward ANN. This allows to represents the nonlinear dynamics that characterizes the system. The data for building the model were collected during a subject-specific acquisition session. The ANN estimated the state transition dynamics of the system given the agents action:

$$\boldsymbol{s}_{n+1} = f_{\boldsymbol{w}}(\underline{\boldsymbol{s}}_n, a_n), \tag{4}$$

where n is the discrete time index, $f_{\boldsymbol{w}}(\cdot)$ is the estimated dynamics of the environment, $\boldsymbol{w}$ is the vector of the neural network weights and $\underline{s}_n = [\phi_n, \dot{\phi}_n, \ddot{\phi}_n, h_n^1, h_n^2, h_n^{'1}, h_n^{'2}]$ is the enlarged state. To increase the amount of information in the input, we extended the state $\boldsymbol{s}_n$ by including four additional signals, h_n^{ch} and $h_n^{'ch}$ (with $ch \subset \{1, 2\}$, indicating the stimulation channel), defined as:

$$h_n^{ch} = \begin{cases} \sum\limits_{c=1}^{n} a_c & \text{if } a_n = ch \\ h_{n-1}^1 & \textit{otherwise} \end{cases}, \quad h_n^{'ch} = \begin{cases} \sum\limits_{c=1}^{n} a_c & \text{if } a_n = ch \\ 0 & \textit{otherwise} \end{cases}$$

Those signals keep memory about the past stimulation sequences and take into account the muscle fatigue.

An acquisition protocol was designed to collect the training data for the ANN (see Section III-A), and the training dataset was defined as follows: the input was a matrix including both the enlarged state and the action at the time sample n, and the output was the vector of the state at the time sample $n+1$. The Keras python library[2] was used to train a single-layer feedforward ANN with 13 tanh hidden neurons, selecting the mean squared error as loss function and the Adam optimization algorithm [13]. Note that the policy does not observe the enlarged state but only a projection of the first three components (see Sec. II-C).

D. PID Control Solution

To define the PID control solution, a SISO plant was considered. The input of the plant was the stimulation pulse width and the output was the elbow angle obtained in response to the given input. The stimulation frequency was fixed at $25Hz$ and the current was set at a subject-specific value as described in Section III-A. The incremental PID control law was considered:

$$u(n) = u(n-1) + k_I(n)e(n) + K_P \Delta e(n) + k_D \Delta^2 e(n), \tag{5}$$

where n is the discrete time instant, $e(n) = \phi^t - y(n)$, $\Delta e(n) = e(n) - e(n-1)$ and $\Delta^2 e(n) = e(n) - 2e(n-1) - e(n-2)$. The proportional, integral and derivative parameters of the PID are k_P, k_I and k_D respectively. The output y was the elbow angle and the control action $u(n)$ was the pulse width (μs) at the time instant n.

1) Step-Response Identification of the Human Arm: Considering the output signal $y(\cdot)$ and the input signal $u(\cdot)$, we used a 2^{nd}-order linear time-invariant system with two poles and one zero to model the system dynamics through the step response method:

$$G(z^{-1}) = \frac{Y(z^{-1})}{U(z^{-1})} = \frac{b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \tag{6}$$

where $Y(z^{-1})$ and $U(z^{-1})$ are the z-transformations of $y(\cdot)$ and $u(\cdot)$, respectively.
We probed the human arm with the step sequence defined in Eq. 7:

$$u(n) = \begin{cases} \overline{u} & \text{if } 0 \leq n \leq N_{tot} \\ 0 & \text{if } n < 0 \end{cases} \tag{7}$$

where $\overline{u}$ is a fixed pulse width value and N_{tot} is the duration of the step signal.
The step sequence was applied 10 times and the average response signal was computed. The system parameters were estimated by applying the instrumental variable (IV) method [14] for discrete time systems.

2) Calibration of the PID parameters: Once identified the plant dynamics $G(z^{-1})$, we considered the PID transfer function in the z-domain:

$$C(z^{-1}) = \frac{U(z^{-1})}{e(z^{-1})} = \frac{k_i + (1 - z^{-1})k_p + (1 - 2z^{-1} - z^{-2})k_d}{1 - z^{-1}}. \tag{8}$$

Then, we considered the nominal closed loop transfer function $H(z^{-1})$:

$$H(z^{-1}) = \frac{C(z^{-1})G(z^{-1})}{1 + C(z^{-1})G(z^{-1})}. \tag{9}$$

[1] The rllab library is available on GitHub (https://github.com/rllab).

[2] Keras library is available at https://keras.io.

The Matlab® automatic PID parameters calibration tool[3] was used to tune the PID gains. We designed the PID controller by balancing the reference tracking and the disturbance rejection performances, while keeping the minimum phase margin. We finally chose the parameters that allowed the closed-loop stability, with the minimum phase margin and no overshoot in the transient phase.

III. Experimental Protocol and Participants

Five healthy subjects (males, average age: 25.25 ± 1.25) were recruited for the study. Each subject was involved in two sessions: a calibration and a testing session.

A. Calibration Session

1) System configuration: The exoskeleton lengths are adjusted on the subject anthropometric measures, the wrist prono-supination and the humeral rotation were fixed on a comfortable position for the subject. The shoulder DoFs were locked by the brakes. The current amplitude was identified separately for the two channels (cur_1 and cur_2) fixing the pulse width at $400 \mu s$ and increasing the stimulation amplitude every second in steps of $1mA$ till reaching a value, tolerated by the subject, able to produce a functional movement without reaching the full range of motion.

2) Acquisition of data for the ANN model of the human arm: During the acquisition session, the subject was asked to be relaxed. The stimulation frequency was randomly modulated by setting a probability to send a pulse to the two channels at each time instant. The stimulation was sent alternately to the two channels in order to cover the whole range of motion. The acquisition procedure lasts 25 minutes and a total number of 35000 samples were collected.

3) Recovery Phase: The subject was allowed to recover for 5 minutes.

4) Acquisition of the data for the PID fine tuning (step response): First, the initial position ϕ_s was chosen. Then, starting always from that position, the pulse width step sequence u (defined in Section II-D1) was sent ten times. The value $\overline{u}$ was set to $400 \mu s$, the total duration of the step signal was set to 50 time instants with at least 10 seconds between two consecutive steps.

B. Testing Session

Each subject tested both controllers in two different tasks (low and high range target) starting always from the same rest position. For each task and each controller 10 repetitions were carried out for a total of 40 repetitions per subject. Each repetition lasted 2 seconds with 10 seconds apart. The setting time, i_{set}, was defined as the first time instant in which the actual elbow angle was closer than 2 degrees from the target position and the instantaneous angular velocity was close to zero. At i_{set} the elbow brake was activated to hold the position and the stimulation was switched off.
In summary, the testing session included the following phases:

- System configuration.
- 10 repetitions of low target controlled by RL.
- Recovery phase of 5 minutes.
- 10 repetitions of low target controlled by PID.
- Recovery phase of 5 minutes.
- 10 repetitions of high target controlled by RL.
- Recovery phase of 5 minutes.
- 10 repetitions of high target controlled by PID.

TABLE I
Setting of the Variables

Subject	ϕ_s $[\circ]$	ϕ_l^t $[\circ]$	ϕ_h^t $[\circ]$	cur_1 $[mA]$	cur_2 $[mA]$
S1	92	120	140	8	9
S2	103	125	150	13	10
S3	100	125	140	5	11
S4	125	70	90	8	10
S5	120	100	80	11	12

Table I shows the values of the parameters for the five subjects.

C. Measures of Performance and Statistical Analysis

The absolute position error (Eq. 10) and the smoothness (Eq. 11) were computed on each repetition. The repeatability of the gesture within the same task was assessed by computing a dissimilarity index, defined in Eq. 12.

$$e^{abs} = |\phi^t - \phi_I| \tag{10}$$

Where ϕ^t is the target position of the current task and ϕ_I is the elbow angular position at the end of the repetition span.

$$sm = \frac{\dot{\phi}_{mean}}{\dot{\phi}_{max}} \tag{11}$$

Where $\dot{\phi}_{mean}$ and $\dot{\phi}_{max}$ are respectively the mean and the maximum instantaneous velocity of the repetition. Smoothness values around 0.5 were previously found in healthy subjects performing similar movements supported by a passive exoskeleton [15].

$$d = \frac{1}{I} \sum_{i}^{I} std(\mathbf{\Phi}(i)) \tag{12}$$

Where $\mathbf{\Phi}$ is the $[10 \times 1]$ vector whose each element is the elbow angle of the j^{th} repetition (with $j = 1, .., 10$) at the time instant i.

Linear mixed model analyses for repeated measures ($p = 0.05$) were made on each of the computed metrics (setting time, smoothness and absolute error). The repeating factor was the task repetition, the task and controller were entered as fixed effects and the metrics as dependent variables. Subjects were included as random effects, since we chose them as representative of the healthy population.

[3] References at https://it.mathworks.com/discovery/pid-tuning.html webpage

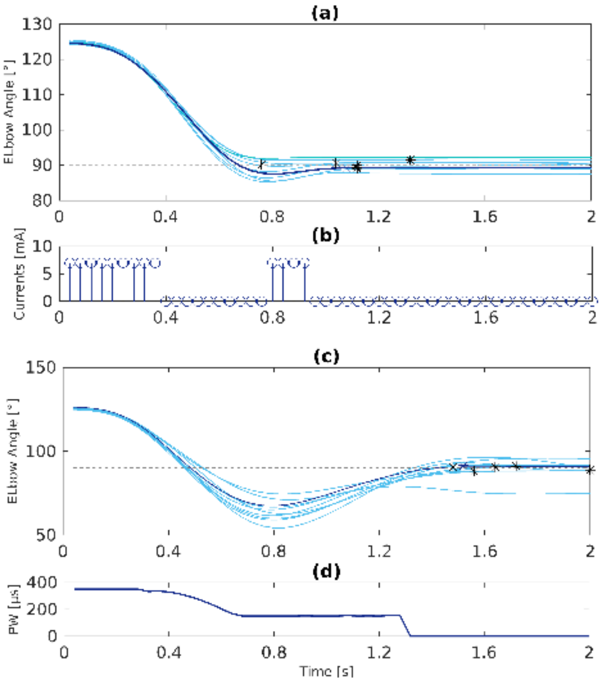

Fig. 2. Repetitions of the low-range target reaching task. Panels (a-b) shows the RL control performances; panels (c-d) those of the PID.

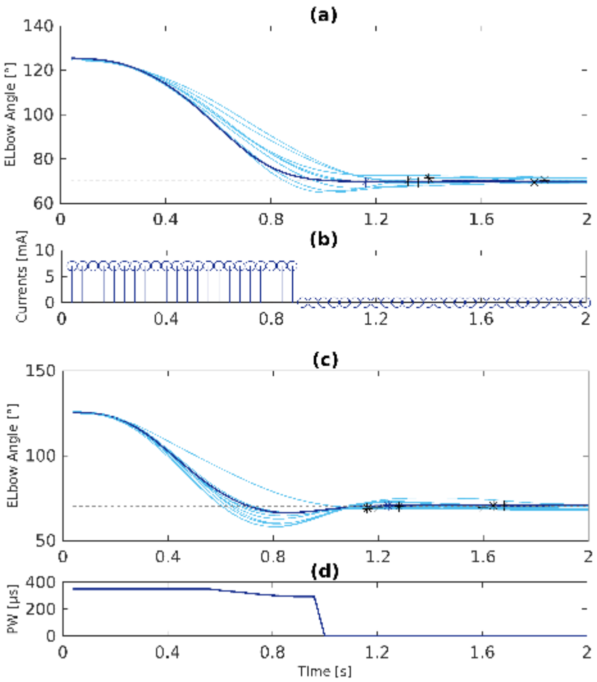

Fig. 3. Repetitions of the high-range target reaching task. Panels (a-b) shows the RL control performances; panels (c-d) those of the PID.

IV. Results

The execution of the reaching tasks by the fourth subject is shown in Figures 2 and 3 for the low and the high range target respectively. Panels (a) and (c) show the elbow angular trajectories of the PPO and the PID respectively. The dashed line indicates the target angle and the asterisks above the trajectories specify their setting time. When the asterisk is missing, the transient phase did not ended within two seconds. For the trajectory highlighted in dark in the panels (a) and (c) the corresponding control action provided by the PPO and PID are shown in panels (b) and (d) respectively.

In both tasks, the PID control was characterized by a higher overshoot in the transient phase. This is particularly noticeable in Figure 2, where the trajectories in panel (c) overtook the target position of over 30°. Also the setting times were shorter in the case of RL control. The control action in the RL case worked as a discrete process in which a single stimulation pulse represented an event. The stimulation patterns were modulated in order to gently drive the arm to the target position, avoiding to over-activate the muscles. In addition, only the agonist muscle was activated during the execution of the task. The control action in the PID case instead, consisted of a continuous modulation of the pulse width until the target was reached. The intrinsic variability of the FES-induced trajectories for the same task is easy to notice by looking at the overlapped repetitions (we obtained dissimilarity values of 1.34°, 3.89°, 2.42° and 2.30° respectively for the trajectories in panels (a) and (c) of Figures 2 and 3).

Table II shows the performance measures achieved by the two control systems. There was a significant difference in terms of controller for all of the three performance measures. The RL control showed better performances by looking at the mean values of position error and smoothness. Moreover, it showed a faster dynamics since it reported shorter setting times.

Transient phase behaviour of the two control systems was compared by counting the times there was an overshoot: a higher number of overshoots was achieved by the PID with a 31% of times (compared to the 16% of the RL control).

In terms of repeatability, we obtained a dissimilarity (median [quartiles]) of about 2.30 [1.46 2.51] and 2.35 [2.31 2.70] for the PID and the RL control respectively.

V. Discussion

Our results revealed that RL can perform better than PID in the control of FES induced flex-extension movements facilitated by a passive exoskeleton. Indeed, a significant effect of the controller in favour of the PPO was found for all the outcome measures. A good accuracy level was obtained for both the RL and the PID controllers since position errors were low. However, the PPO outperformed the PID in terms of speed of response and general behaviour in the transient phase: the movements driven by the RL resulted not only faster but also smoother and more natural, rarely generating overshoots. Being both closed loop control systems, the RL and the PID were able to compensate for disturbances due to the intra-subject variability of the system dynamics. However, the RL solution had some advantages.

TABLE II
PERFORMANCE MEASURES OF THE RL AND PID CONTROLLER.

$Metrics$	RL		PID		$P_{controller}$	P_{task}	$P_{interaction}$
	low	high	low	high			
	$n \approx 10$	$n \approx 10$	$n \approx 10$	$n \approx 10$			
e^{abs} $[\circ]$	0.677 ± 0.119	0.489 ± 0.082	0.685 ± 0.105	1.002 ± 0.115	0.003	0.388	0.003
sm $[0:1]$	0.523 ± 0.035	0.471 ± 0.032	0.469 ± 0.038	0.456 ± 0.042	0.004	0.742	0.031
i^{set} $[s]$	1.055 ± 0.084	1.220 ± 0.092	1.364 ± 0.078	1.366 ± 0.091	< 0.001	0.449	0.508

First, considering the control laws, the action of the RL control is based on the current state only, while for the PID it is determined after computing the actual error. This features makes the RL controller work as a look-up table and compensate for the intrinsic time-lag in the closed loop response. On the other side, the PID controller is a linear-time invariant system and, for stability reasons, the closed-loop response should not be fast. Moreover, the PID was tuned on the linearized dynamics of the human arm response, therefore its transient behaviour was greatly affected by the the target angle.

In addition, the control action of the RL control resulted more efficient in terms of muscular activation. It was able to stop the stimulation when no muscle activation was needed, while the PID controller stimulated the muscle without interruption, thus favouring the onset of muscle fatigue.

It is also worthy to notice that the PPO modulates the frequency while the PID modulates the pulse width. Thus, the two control systems differ both in terms of algorithm (PPO versus PID) and in terms of control action (frequency versus pulse width). This could be a limitation of the study since it makes challenging to separate the effects. However, we decided to implement a frequency-based controller since the frequency modulation has been revealed more efficient in muscle force production [16]. On the other hand, for the PID we chose to modulate the pulse width, since this is the traditional approach found in the literature. Further research on its use is therefore envisaged.

VI. CONCLUSION

In this work, we compared RL and PID control solutions for an upper limb FES system combined with a passive exoskeleton. The performance of the two controllers were tested during elbow flex-extension movements. Our results showed that the RL control outperformed the PID.

RL also presents several advantages that could be further exploited. First, the control problem could be extended considering trajectory tracking tasks, as demonstrated in pole balancing experiments [17], and involving more degrees of freedom. Then, an online learning approach based on RL techniques can be investigated to obtain fine-tuned controllers, able to adapt to changes in the system dynamics (e.g., muscle fatigue or due to muscle recovery) [18].

ACKNOWLEDGMENT

The project was funded by the European project RETRAINER (Horizon 2020, grant agreement no. 644721). We also thanks Axelgaard Manufacturing Ltd. for donating us the stimulation electrodes.

REFERENCES

[1] French, L. H. Thomas, J. Coupe et al. Repetitive task training for improving functional ability after stroke, Cochrane Database Syst. Rev., vol. 2016, no. 11, pp. 102104, 2016.

[2] Pedrocchi, A., Ferrante, S., Ambrosini, E. et al. MUNDUS project: MUltimodal Neuroprosthesis for daily Upper limb Support (2013) Journal of NeuroEngineering and Rehabilitation, 10 (1), art. no. 66.

[3] Klauer, C., Schauer, T., Reichenfelser, W. et al. A.Feedback control of arm movements using Neuro-Muscular Electrical Stimulation (NMES) combined with a lockable, passive exoskeleton for gravity compensation (2014) Frontiers in Neuroscience, 8 (SEP), art. no. 262.

[4] O.A. Howlett, N.A. Lannin, L. Ada, C. McKinstry, Functional electrical stimulation improves activity after stroke: a systematic review with meta-analysis, Arch Phys Med Rehabil, vol. 96, pp. 934-43, 2015.

[5] Bolsterlee B, Veeger DH, Chadwick EK. Clinical applications of musculoskeletal modelling for the shoulder and upper limb. Med Biol Eng Comput. 2013;51(9):953:63.

[6] Lynch C. L., Popovic M. R. Functional Electrical Stimulation. Closed-Loop Control of Induced Muscle Contractions. IEEE Control Systems Magazine, April 2008.

[7] Kirsch N., Alibeji N., Sharma N., Nonlinear model predictive control of functional electrical stimulation. Control Engineering Practice. Vol 58, Jan 2017, pp. 319-331.

[8] Moon. S. H., Choi J. H., Park S. E. The effects of functional electrical stimulation on muscle tone and stiffness of stroke patients. J Phys Ther Sci. 2017 Feb; 29(2): 238-241.

[9] Kathleen M. Jagodnik, Philip S. Thomas, Antonie J. van den Bogert, Michael S. Branicky, and Robert F. Kirsch. Human-Like Rewards to Train a Reinforcement Learning Controller for Planar Arm Movement. IEEE Transactions on Human-Machine Systems, vol. 46, no. 5, oct 2016

[10] Schulman J., Wolski F., Dhariwal P., Radford, Klimov O., Proximal Policy Optimization Algorithms. 2017. Submitted 20 Jul 2017 (v1), last revised 28 Aug 2017. OpenAI.com

[11] Duan Y.,, Chen X.,, Houthooft R.,, Schulman J., Abbeel P.,. Benchmarking Deep Reinforcement Learning for Continuous Control. Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016.

[12] Sutton R. S., McAllester D., Singh S., Mansour Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932.

[13] Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. Submitted 22 Dec 2014 (v1), last revised 30 Jan 2017.

[14] Young P. The Instrumental Variable Method: A Practical Approach to Identification and System Parameter Estimation. IFAC Proceedings Volumes. Vol. 18, July 1985, pp 1-15.

[15] Ambrosini E, Ferrante S, Rossini M et al. Functional and usability assessment of a robotic exoskeleton arm to support activities of daily life. Robotica. 2014; 32(08):1213-1224.

[16] Kesar T, Li-Wei C, Binder-Macleod S A. Effects of stimulation frequency versus pulse duration modulation on muscle fatigue. J Electromyogr Kinesiol 2008. 18(4): 662-671

[17] Bonarini A, Caccia C, Lazaric A , Restelli M. Batch Reinforcement Learning for Controlling a Mobile Wheeled Pendulum Robot. IFIP International Conf. on AI in Theory and Practice. 2008. 151-160.

[18] Manganini G., Pirotta M., Restelli M. et al. Policy Search for the Optimal Control of Markov Decision Processes: A Novel Particle-Based Iterative Scheme. IEEE Trans. on Cybernetics, vol. 46, no. 11, pp. 2643-2655, Nov. 2016.