An upper limb Functional Electrical Stimulation controller based on Reinforcement Learning: A feasibility case study

D. Di Febbo[1], E. Ambrosini[1], M. Pirotta[2], M. Restelli[3], A. Pedrocchi[1] and S. Ferrante[1]
[1] *NEuroengineering And Medical Robotics Laboratory, Politecnico di Milano, Italy*
[2] *SequeL Team, Inria Lille – Nord Europe, France*
[3] *Artificial Intelligence and Robotics Lab, Politecnico di Milano, Italy*

***Abstract*—Controllers for Functional Electrical Stimulation (FES) are still not able to restore natural movements in the paretic arm. In this work, Reinforcement Learning (RL) is used for the first time to control a hybrid upper limb robotic system for stroke rehabilitation in a real environment. The feasibility of the FES controller is tested on one healthy subject during elbow flex-extension in the horizontal plane. Results showed an absolute position error <1.2° for a maximum range of motion of 50°.**

***Keywords*—Functional Electrical Stimulation, Reinforcement Learning, Rehabilitation, Hybrid Robotic Systems.**

I. Introduction

FUNCTIONAL electrical stimulation (FES) is an effective technology used in rehabilitation to restore impaired motor functions in people affected by neurological disorders [1]. Following stroke, spinal cord injuries or multiple sclerosis, patients often have difficulties in performing complete movement with the upper limb as well as in grasping and manipulating objects. Rehabilitative treatments, based on FES, aim at restoring motor functions of the impaired extremity by activating paretic muscles through electrical stimulation [2]. Recently, different robotics solutions have been proposed to be coupled with FES, the so-called "hybrid robotic systems", in order to facilitate the execution of motor exercise in the plane or in the space and increase the rehabilitative outcomes [3]. However, reliable controllers driving accurate and natural movements through FES are still under investigation. The electrically stimulated human muscle is a highly nonlinear system whose physiological properties are difficult to be modelled [4]. Moreover, it is a strongly time-variant system: spasticity and fatigue can significantly influence the performance in the short period, and muscle strengthening and motor relearning can improve performance in the medium/long period.

Classical control solutions [5] rely on the accuracy of the model by which the system is described. Due to the complexity of the electrically stimulated muscles response, linear assumptions are frequently made. More advanced techniques, such as non-linear [6] and adaptive control systems [7], have been tested in real environments. However, the increased complexity of the controllers implicates more onerous calibration procedures [8], not suitable for clinical settings and non-technically trained operators.

Recently, reinforcement learning (RL) [9] has been investigated to solve upper limb FES control problems in simulation [10]. RL is a sub-field of machine learning which studies how agents can learn from experience collected by interacting with the environment. In order to achieve desired performances, RL algorithms can directly learn an optimal non-linear control law without prior knowledge about the system. The control law, or policy, can be adjusted over time in order to face with dynamical changes of the environment.

In this work, we used the Proximal Policy Optimization (PPO) RL algorithm [11] to control a FES-driven upper extremity of a healthy subject performing four planar reaching tasks supported by a passive exoskeleton. In order to deal with the exploration-exploitation RL problem and the repeated interaction with the environment, we trained the PPO algorithm off-line using an artificial neural network (ANN) model of the subject's arm. We collected the training data for the ANN model through an ad-hoc designed acquisition protocol, with the aim of including as much information as possible about the system dynamics. The feasibility of the controller was tested in a real environment, with a healthy subject who was asked to be completely passive, during elbow flex-extension movements.

II. Methods

A. Apparatus

The robotic system for the upper limb (Fig. 1) consisted of a lightweight passive exoskeleton for the right arm characterized by 3 degrees of freedom (DOF): shoulder rotation, shoulder elevation and elbow flex-extension. The humeral rotation and the prono-supination were fixed at comfortable positions for the subject. Each DOF is equipped with an angle sensor (Vert-X 13 E, ConTelec AG) to measure the position and an electromagnetic brake to lock a desired target position.

The gravity compensation modules for upper arm and forearm consisted of a carbon fiber-tube with springs inside whose pre-tension can be adjusted session in order to change the level of compensation.

Trains of biphasic pulses were sent through surface patch electrodes (Pals® electrodes, Axelgaard Manufacturing Co., Ltd.) by means of a four-channel current-controlled stimulator (RehaMovePro, Hasomed GmbH). Only two channels were used, one connected to the biceps brachii and one to the triceps brachii muscle. The exoskeleton and the stimulator were controlled by an embedded processor (BeagleBoneBlack ®^{TM}) in which the I/O communication was set at 25 Hz.

Fig. 1: The exoskeleton for the right arm. The coloured arrows indicate the degrees of freedom.

B. Selection of the Motor Tasks

With the aim of investigating the feasibility of the RL control in a real environment, we selected four single target reaching exercises in the horizontal plane: two elbow-flexion and two elbow-extension tasks. The elbow angle, $\phi(\cdot)$, could range between 50° (maximum flexion) and 180° (complete extension) and two starting angles, ϕ_0^f (for the flexion tasks) and ϕ_0^e (for the extension tasks), were defined as the subject's relaxed arm positions. Then, four target angles were defined: ϕ_{lf}^t and $\phi_{hf}^t \in (50°,\ \phi_0^f)$ for the low-range and the high-range elbow-flexion, and ϕ_{le}^t and $\phi_{\mathrm{he}}^{\mathrm{t}} \in (\phi_{0,}^{\mathrm{e}}\ 180°)$ for the low-range and the high-range elbow-extension, respectively.

C. Control System Architecture

The subject's arm combined with the passive exoskeleton represented the plant. In a discrete time domain, we considered the state vector of the system $\boldsymbol{s}_i = [\phi_i,\ \dot{\phi}_i,\ \ddot{\phi}_i]' \in \mathbb{R}^3$ composed of the elbow angular position, the instantaneous angular velocity and the instantaneous angular acceleration (i is the discrete time instant). The current of the two stimulation channels were the inputs. 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. Therefore, we defined a scalar control variable, $\mathrm{a_i}$, which encoded, in each time step, the action of sending a single stimulation pulse on the first channel, on the second channel or in neither one (with the values 1, 2 and 0 respectively). We defined the state-feedback control law, as reported in Eq. (1):

$$\mathrm{a_i} = \mathrm{f}_{\phi^{\mathrm{t}}}^{\mathrm{c}}(\mathbf{s_i}) \qquad (1)$$

Where $\mathrm{f}_{\phi^{\mathrm{t}}}^{\mathrm{c}}(\cdot): \mathbb{R}^3 \to \mathrm{a} \subset \{0,1,2\}$ is some optimal control function w.r.t. the goal of driving the subject's arm from the starting position to the target angle ϕ^{t}.
We used the formalism of Markov Decision Process to model the control problem and we used RL to solve the optimal control problem (for each of the four motor tasks) without requiring the explicit knowledge of the dynamics.

D. The RL Problem Statement and Solution

In the general RL framework, an agent has to learn an optimal policy by interacting with the environment, according to a predefined objective. Based on our control architecture, the environment represents the plant and the policy the optimal control function.

Considering a discrete time setting, we formally described the episodic RL problem as a Markov decision process (MDP) with the tuple $\langle S, A, P, R, \gamma, \mu \rangle$, where S is a continuous set of states $\boldsymbol{s} = [\phi,\ \dot{\phi},\ \ddot{\phi}]'$, A is the finite set of all possible actions $a \subset \{0, 1, 2\}$, P is the transition probability kernel such that $\mathrm{P}[\boldsymbol{s}_{i+1} \mid \boldsymbol{s}_i, a_i]$ represents the probability of reaching the state $\boldsymbol{s}_{i+1}$ from state $\boldsymbol{s}_i$, by performing action a_i, R is the reward function $r(\boldsymbol{s}, a) = E[r \mid \boldsymbol{s},\ a]$, $\gamma: 0 \leqslant \gamma \leqslant 1$ is the discount rate and μ is the initial state distribution. Our scenario can be formalized as an episodic RL problem. An episode defines a finite time interval $I, 0 \leqslant i \leqslant I$: at every time instant i, the agent is in the state $\boldsymbol{s}_i$, and makes an action a_i and reaches the next state $\boldsymbol{s}_{i+1}$. At the same time, the environment produces a scalar reward r_i. The agent goal is to find a policy $\pi(a \mid \boldsymbol{s})$ that maximizes the sum of the collected discounted rewards, as described in Eq. (2):

$$J(\pi) = E_\pi[\, G_0 \mid \boldsymbol{s}_0 \sim \mu\,], \qquad (2)$$

where G_0 is the return at the time instant $i = 0$:

$$G_0 = \sum_{k=0}^{T} \gamma^k r_k \qquad (3)$$

The reward function defines the learning goal, i.e. it defines what the agent has to learn. Hence, we defined the reward function as in Eq. (4):

$$r_{i+1} = -(\phi^t - \phi_i)^2 - \alpha \cdot \ddot{\phi}_\iota \qquad (4)$$

Where ϕ^{t} is the target position of the task, ϕ_{i} is the actual position at time instant i, $\ddot{\phi}$ is the instantaneous acceleration and α is a scaling parameter. The effect of such reward function was that to penalize the distance from the target in each time instant. Moreover, we added a penalization term for the instantaneous acceleration in order to favour smooth movements.

To solve the RL problem, we used the PPO policy gradient method, which has been shown better performances in continuous control tasks than several previous algorithms from literature. We used the PPO implementation in rllab [12] (available at https://github.com/rll/rllab) choosing a multi-layer perceptron, characterized by two hidden layers with 26 hyperbolic tangent (tanh) neurons and a softmax output layer as function estimator. Finally, we set the maximal path length equal to the episode duration I, the batch size equal to $\mathrm{I} \times 100$ and the total number of iteration equal to 150.

E. ANN-based Model of the Human Arm

We created a model of the plant and we used it as the RL environment in order to efficiently train the controller. We chose a feedforward ANN to map the non-linear dynamics of the human arm and we trained it with real data collected during a 20-minute acquisition session. The ANN estimated the state transition dynamics of the environment given the agent's action, as defined in Eq. (7):

$$\mathbf{s}_{\mathrm{n+1}} = \mathrm{f_w}\left(\underline{\mathbf{s}}_{\mathrm{n}}, \mathrm{a_n}\right) \qquad (7)$$

Where $\mathrm{f_w}(\cdot)$ is the estimated dynamics of the environment, $\mathbf{w}$ is the vector of the neural network weights, $\underline{\mathbf{s}}_{\mathrm{n}} = [\phi_{\mathrm{n}},\ \dot{\phi}_{\mathrm{n}},\ \ddot{\phi}_{\mathrm{n}},\ \mathrm{h_n^1},\ \mathrm{h_n^2},\ \mathrm{h_n'^1},\ \mathrm{h_n'^2}]'$ is the enlarged state, and n is a discrete time index. To increase the amount of input information, we extended the state $\mathrm{s_n}$ by including four additional signals, $\mathrm{h_n^{ch}}$ and $\mathrm{h_n'^{ch}}$, defined as in Eq. (8) and Eq. (9):

$$h_n^{ch} = \begin{cases} \sum_{c=1}^{n} a_c, & \text{if } a_n = ch \\ h_{n-1}^{ch}, & \text{otherwise} \end{cases} \quad (8)$$

$$h'^{ch}_n = \begin{cases} \sum_{c=1}^{n} a_c, & \text{if } a_n = ch \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Where $ch \subset \{1, 2\}$ indicates the stimulation channel. These four additional signals take into account the previous stimulation sequences, thus considering muscle fatigue.

We designed an ad-hoc experimental protocol to collect the real data for training the ANN. During the acquisition, the subject wore the exoskeleton and the two shoulder DOFs were locked in a position which allowed horizontal planar elbow flex-extension movements. The subject was asked to remain passive, while stimulation sequences were sent to both muscles. The current amplitude of the two stimulation channels (cur_1 and cur_2) were initially identified at comfortable values for the subject and the pulse width was set at 350 μs. The stimulation frequency was randomly modulated by changing the probability to send a stimulation pulse at each time instant. The two stimulation channels were activated individually in an alternate manner in order to cover the entire range of motion. The acquisition procedure lasted 20 minutes and a total number (N) of 35000 samples were collected. We used the data sampled at each time instant n to compute the extended state $\underline{\mathbf{s}}_n$. Then, we defined the inputs matrix X $[N \times 8]$, whose n^{th} row is $x_n = [\underline{\mathbf{s}}'_n, a_n]$, and the targets matrix Y $[N \times 3]$, whose n^{th} row is $y_n = \mathbf{s}'_{n+1}$, for training the ANN model.

We used Keras (https:/keras.io) to implement and train a single-layer feedforward ANN with 13 hidden tanh neurons, setting the mean squared error as performance function and the Adam optimization algorithm [13].

F. Experimental Protocol and Measures of Performance

We tested the policies obtained for the four target angles on one healthy subject (male, 26 years old). As in the data collection for training the ANN, the subject was asked to remain passive and wore the exoskeleton; the two shoulder DOFs were locked in order to allow elbow flex-extension movements in the horizontal plane. The stimulation parameters were the same used in section II.E. Each policy was tested 10 times; during each repetition the brake of the elbow joint was activated when the angular position was in the range of ±2° from the target with an instantaneous angular velocity close to zero. The time instant at which the elbow brake was activated was defined as i^{set}.

We evaluated each repetition in terms of the time needed to reach the target, i^{set}, the absolute position error, e^{abs} (see Eq. (10)), and the smoothness, sm (see Eq. (11)).

$$e^{abs} = |\phi^t - \phi_I| \quad (10)$$

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

$$sm = \frac{\dot{\phi}_{mean}}{\dot{\phi}_{max}} \quad (11)$$

Where $\dot{\phi}_{mean}$ and $\dot{\phi}_{max}$ are the mean and the maximum instantaneous velocity within the repetition, from the start to the setting time i^{set}.

Furthermore, we evaluated the repeatability of the gesture within the same task by computing a dissimilarity index, defined in Eq. (12).

$$d = \frac{1}{l}\sum_i^l std(\boldsymbol{\phi}^i) \quad (12)$$

Where $\boldsymbol{\phi}^j$ is the [10 x 1] vector containing the samples of all the repetitions at time instant i. Values of the metric d close to zero indicate a high reproducibility of the movement between the repetitions.

The values of the parameters are reported in Table I.

TABLE I
SETTING OF THE VARIABLES

ϕ_0^f	ϕ_0^e	ϕ_{lf}^t	ϕ_{hf}^t	ϕ_{le}^t	ϕ_{he}^t	γ	I	α	cur_1	cur_2
92°	125°	90°	70°	120°	140°	0.99	50	10	8mA	10mA

III. RESULTS

First, we collected the subject's training data for the ANN model with the acquisition protocol described in section II.E. Then, we trained the model (mse=0.002, 0.019, 0.028, for ϕ, $\dot{\phi}$, $\ddot{\phi}$ respectively) and we used it as a simulated environment for our RL experiments. We obtained a policy for each motor task by running the PPO algorithm and we tested the achieved policies on the same subject, according to the experimental protocol reported in section II.F.

Table II reports the performances measures achieved by the RL control in the four tasks. Each task is represented by the target angle (one per target angle). The low values of the absolute position error showed the capability of RL control to drive accurate movements. However, the amount of dispersion around the mean suggested a consistent variability in the performance of the controller. This is mainly due to the high intra-subject variability of FES-induced movements.

TABLE II
PERFORMANCE MEASURES OF THE RL CONTROL

	e^{abs} [°]	i^{set} [s]	sm	d[°]
$\boldsymbol{\phi}_{lf}^t$	1.14 ± 0.74	1.16 ± 0.33	0.39 ± 0.08	1.14
$\boldsymbol{\phi}_{hf}^t$	0.95 ± 0.78	1.66 ± 0.03	0.42 ± 0.11	2.30
$\boldsymbol{\phi}_{le}^t$	0.56 ± 0.45	1.34 ± 0.34	0.53 ± 0.13	2.55
$\boldsymbol{\phi}_{he}^t$	0.97 ± 0.87	1.50 ± 0.14	0.46 ± 0.06	1.83

Overall, the four tasks achieved values of smoothness of about 0.5 which was previously found in healthy subjects who performed similar voluntary movements only with the help of a passive exoskeleton for weight relief [14].

In Fig. 2 and Fig. 3, we reported two angular trajectories and the stimulation pulses for two exemplary tasks. All repetitions of one task are shown in panel (a) to provide a visual indication of repeatability. One trajectory representative of the mean behaviour and one with a behaviour different from the mean were highlighted. For each highlighted trajectory, we reported the relative control action in panels (b) and (c).

In both tasks, the stimulation train corresponding to the most accurate movements (angular trajectories in blue) was able to smoothly stop the arm in the target position. On the other hand, in the worst trajectory (shown in cyan) the arm was moving too fast, then the stimulation was switched off to slow down

the movement and reactivated after few time instants to reach the target. Such a stimulation train reduced the smoothness of the movement. It is worthy to notice that, even if the controller was allowed to activate both stimulation channels for all of the tasks, it always decided to activate only one channel. The optimal control strategy, indeed, just involved the stimulation of the agonist muscle to reach the target.

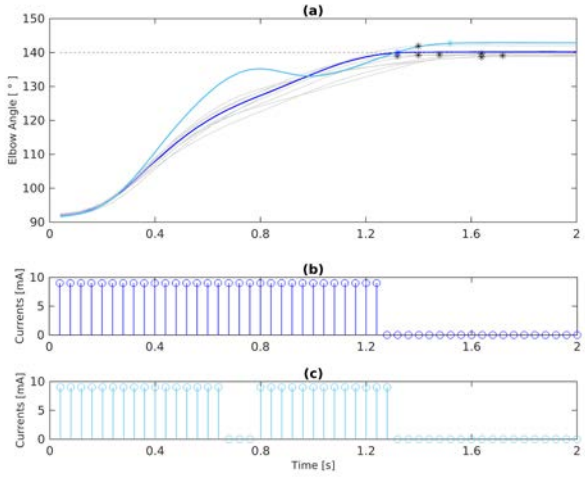

Fig. 2: Performance of the RL control during the execution of the high-range elbow flexion task. Panel (a) shows the angular trajectories of the ten repetitions; the setting times, when present, are indicated by the black stars. Panels (b) and (c) show the stimulation sequences relative to the trajectory highlighted with the same colour in panel (a).

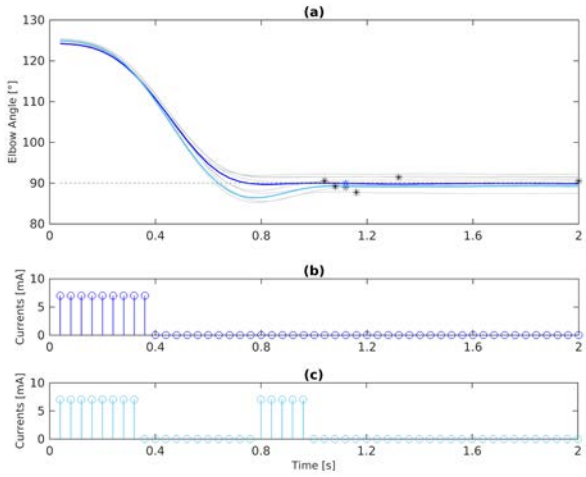

Fig. 3: Performance during the low-range flexion task. Panel (a) shows the angular trajectories (black stars indicates setting times). Panels (b) and (c) show the stimulation trains relative to the trajectory in panel (a).

IV. Conclusion

We developed and tested in a real environment an RL algorithm to control an upper limb hybrid robotic system consisting of FES to the biceps brachii and triceps brachii muscles and a passive exoskeleton for weight relief. We created a simplified control problem by involving a healthy subject, performing planar reaching tasks and modulating the stimulation frequency, and we trained the PPO algorithm off-line using an ANN model of the subject's arm. At last, we tested the obtained policies on one healthy subject.

This work showed the feasibility of the RL control of FES and provided promising results about its capability to overcome the typical problems of the traditional control solutions, mainly due to the complex nature of the response of electrically stimulated muscles. However, our results are very preliminary since the controller was tested only on one subject with no proof about the absence of voluntary contractions.

RL control has further advantages that could be exploited in the future to improve a learning-based control approach. In position control tasks, more complex policies have been demonstrated capable to track a time-variant set point [15], therefore they could be used to implement trajectory tracking controllers for FES. Moreover, on-line learning methods could also be exploited to adapt the RL control system according to the changes of the human arm dynamics in the short (fatigue) and long term (motor recovery). Further evaluations are needed to confirm these promising results.

Acknowledgement

The project was funded by the European project RETRAINER (H2020, GA no. 644721). We thank Axelgaard Manufacturing Ltd. for donating us the stimulation electrodes.

References

[1] J. H. Burridge, M. Ladoucer, Clinical and therapeutic applications of neuromuscular stimulation: a review of current use and speculation into future developments. Neuromodulation, 2001. 4:147-154.

[2] French, L. H. Thomas, J. Coupe, N. E. Mcmahon, L. Connell, J. Harrison, C. J. Sutton, S. Tishkovskaya, and C. L. Watkins, "Repetitive task training for improving functional ability after stroke," Cochrane Database Syst. Rev., vol. 2016, no. 11, pp. 102–104.

[3] Klauer, C., Schauer, T., Reichenfelser et Al. 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] 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.

[5] Vette AH, Masani K, Kim JY, Popovic MR. Closed-loop control of functional electrical stimulation-assisted arm-free standing in individuals with spinal cord injury: a feasibility study. Neuromodulation. 2009;12(1):22-32.

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

[7] Abbas JJ, Triolo RJ. Experimental evaluation of an adaptive feedforward controller for use in functional neuromuscular stimulation systems. IEEE Trans Rehabil Eng. 1997;5(1):12:22.

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

[9] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA. 1998.

[10] Thomas P, Branicky M,1 van den Bogert A, Jagodnik K. Application of the Actor-Critic Architecture to Functional Electrical Stimulation Control of a Human Arm. Proc Innov Appl Artif Intell Conf. 2009;165:172.

[11] 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

[12] 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.

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

[14] Ambrosini E, Ferrante S, Rossini M, Molteni F, Gföhler M, Reichenfelser W, Duschau-Wicke A, Ferrigno G, Pedrocchi A. Functional and usability assessment of a robotic exoskeleton arm to support activities of daily life. *Robotica*. 2014; 32(08):1213-1224.

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