



Using a Collaborative Robot to the Upper Limb Rehabilitation

Lucas de Azevedo Fernandes¹(✉) , José Luis Lima² , Paulo Leitão² ,
and Alberto Yoshiro Nakano¹

¹ Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brazil
lucas_azevedo33@hotmail.com

² CeDRI - Research Centre in Digitalization and Intelligent Robotics,
Polytechnic Institute of Bragança and INESC TEC, Porto, Portugal

Abstract. Rehabilitation is a relevant process for the recovery from dysfunctions and improves the realization of patient's Activities of Daily Living (ADLs). Robotic systems are considered an important field within the development of physical rehabilitation, thus allowing the collection of several data, besides performing exercises with intensity and repeatedly. This paper addresses the use of a collaborative robot applied in the rehabilitation field to help the physiotherapy of upper limb of patients, specifically shoulder. To perform the movements with any patient the system must learn to behave to each of them. In this sense, the Reinforcement Learning (RL) algorithm makes the system robust and independent of the path of motion. To test this approach, it is proposed a simulation with a UR3 robot implemented in V-REP platform. The main control variable is the resistance force that the robot is able to do against the movement performed by the human arm.

Keywords: Robotics rehabilitation · Collaborative robots · Simulation · Reinforcement learning algorithm

1 Introduction

According to World Health Organization (WHO), the absolute number of Years Lived with Disability (YLDs) increased more than 17 million between 2005 and 2015, from this value about 74% of YLDs is linked to health conditions which rehabilitation can be beneficial [1]. The major dysfunction cases affect the elderly population, being considered the risk group of cardiovascular diseases, respiratory diseases, and other disabling conditions. As stated by [2], the world's population aged 60 years and over is set to increase from 841 million in 2013 to more than 2 billion in 2050.

The definition of rehabilitation according to [3] is as it follows: “*The use of all available means to reduce the impact of disabling situations and allows disabled individuals to achieve full social integration*”. Given the growing concerns around

physical therapy, various new technologies have been put to the test with patients to improve their treatment or assist them in their ADLs.

Rehabilitation, therefore, is a relevant process to help a large percentage of the population suffering from some dysfunction previously presented. Thus, the main goal of this project is to demonstrate how a collaborative robot can help the patients with non-paralysing dysfunctions, principally in the joints and muscles of the shoulder. The proposed system is composed of a collaborative robot (UR3) and a computer to perform the calculations referring to a possible insertion of an autonomous control module, called self-control module. The UR3 provides resistance to the patient's arm motion, i.e., when the patient execute some force, in turn, the robot applies an opposite force to this movement.

This paper presents the results measured through a simulation in the V-REP platform, as the force acquired in the movement and the variation of the UR3's resistance. In the virtual platform, an interface was developed to perform the control of some variables in the rehabilitation process, such as the applied torque in UR3 joints and the force detected by the sensor. Also, the obtained information of the contact between human and robot is showed, and represents a tool to therapists to analyse the advance of treatment. The force data about the movement will be acquired by a force sensor attached to the UR3. The data measured by the sensor represents the feedback of the environment and will be used to analyse the system.

The rest of this paper is organised as follows. Section 2 briefly explains some similar approaches. In Sect. 3, it is presented the development of the simulation, the analysis of the main concerns in the implementation of these systems, and the control through the self-control module. Section 4 exposes the obtained results in V-REP platform. Finally, Sect. 5 presents some conclusions regarding this problem and future works.

2 Related Work

Reflected by the evolution of robotics and control systems, there is an increasing number of research and development of products that address robotics for rehabilitation. The main studies in rehabilitation field use the following control methods: Electromyography (EMG) biofeedback, robot-assisted therapy, Virtual Reality (VR) based interventions, Constraint-Induced Movement Therapy (CIMT), and Functional Electric Stimulation (FES) [4]. This work focuses on a robot-assisted therapy method.

An aspect that must be considered is the acceptance of rehabilitation robots by patients and therapists. A study conducted in [5] shows a report of the use of a robot-assisted therapy method with 22 patients and 11 occupational therapists. The exercises with the patients were simple and consisted of the movement of a robotic arm with a linked button. The patient should press the button in some previously established positions. The system measures the time between two button presses and other information. After the exercises, various questionnaires are filled to evaluate the system.

Nowadays, it is still difficult to find works that relate simulation of robotic system with rehabilitation robots, due to the difficulty of modelling human limbs and complex robotic systems. In [6] is presented many types of robot-based systems and their controls. One of these control methods is the impedance-based assistance, in which the robots measure the position in a determined trajectory and are capable of inducing a force to correct patient's movements. The major difficulty is to adapt the system for an individual patient. One of the systems that shows this control is proposed in [7], using a robotic system to improve coordination of upper and lower limbs in elderly people.

Systems as MIT-Manus, one of the most well-known rehabilitation system, are composed by manipulators robots. The use of robotic manipulators is troublesome since many of these systems that are used in the industry have not been adapted and prepared to work with humans and, even less, with the human touch, but some adjustments can be made to become these systems suitable to rehabilitation. Table 1 presents some examples of rehabilitation systems with manipulators robots.

Table 1. Rehabilitation Robots based in Robotic Manipulators.

System	Developer	Treatment of...
Braccio di Ferro [8]	University of Genova, Italy	Shoulder + elbow
Gentle/S [9]	University of Reading, UK	Shoulder + elbow + forearm + wrist
iPAM [10]	University of Leeds, UK	Shoulder + elbow + forearm
MIT-MANUS [11]	Massachusetts Institute of Technologies, USA	Shoulder + elbow
REHAROB [12]	Budapest University of Technology and Economics, Hungary	Shoulder + elbow + forearm

3 Proposed System

One of the concerns in this field is the wasted time dedicated to learn and set it up the rehabilitation robots for each patient. Therefore, to develop a new device or method in the rehabilitation field is relevant to think about the utility, usability and facility of the system on both sides, patient and therapist.

The developed work is designated as an end-effector system. These type of systems have a unique fixation point, the physiotherapy exercises and movements are defined in Cartesian coordinates XYZ and the assistance level is modulated by control of impedance or admittance [13].

3.1 Simulation Model

The proposed simulation will be developed in V-REP platform, and its objective is being a tool to test the system with a model of the human arm and predict

the possible failures and corrections needs. The simulation of human arm takes into account the possible consequences of a dysfunction, i.e., the joints of the shoulder may exhibit a small force on the movement or variations of it.

The V-REP is a simulation software that integrates virtual robots with a development environment and can simulate robotic arms, mobile robots, sensors, tools and many different scenes. The Integrated Development Environment (IDE) is programmed in LUA language, and also, presents tools for communications with remote Application Programming Interface (API) as MATLAB [14].

The simulation has three essential parts: The scene with the robot (UR3) and the manikin (model of the human body), the codes that control the simulation, and the control interface. Basically, the simulated elements are composed of mechanical parts and joints. The relevant dynamic properties of the first part that may be changed in the simulation are, mainly, mass, and inertial moment. The joints can also be configured, and its dynamic features refers to the motion: as maximum torque, target velocity, and position control. All these technical features in V-REP linked to a physics engine results in a simulation closest to the real environment.

The developed scene is shown in Fig. 1. It contains two main objects, the robot and the manikin, and both models are developed by outsources. The simulated robot is the UR3, a manipulator with six Degrees of Freedom (DOF), and its joints can be simulated separately. At the end effector of the UR3, a force sensor show the exact force of the human arm during the entire simulation. The UR3 was chosen because it may execute a sufficient torque for the human contact when the joints are used together. The maximum joint torques of this robot are presented in [15].

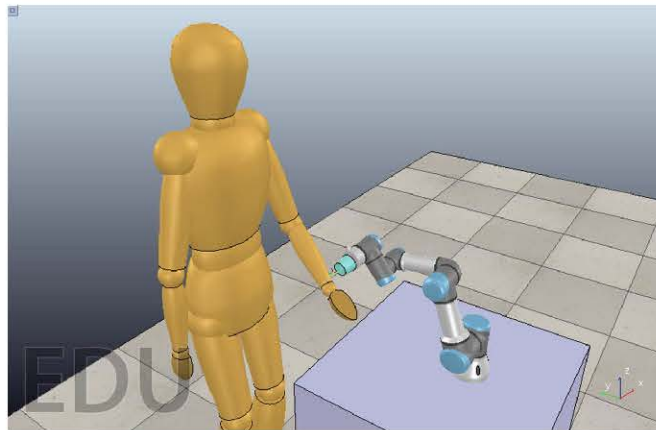


Fig. 1. The Simulated System in V-REP platform.

The goal of the UR3 is to provide a resistance force. In the simulation, the speed of the robot's joint is set as zero, and the torque is increased or decreased, the UR3's expected behaviour is to stand still, so when the human arm exerts its force, the robot applies the opposite force based on the maximum torque set

in the simulation. If the torque provided by the robot is larger than the human arm, there will be no motion; otherwise, if the torque is smaller, the force will not fulfil its function.

The simulated manikin is a special case because only the model of the arm is used. Thus, the main joints for this approach are those that represent the shoulder. The configuration of the manikin takes into account the real values of a human upper limb. A study accomplished in [16] shows a model for estimating the force and the torque during the abduction of the shoulder. The experiment consists in the execution of the movement by the subject and the measure of the performed force through a dynamometer. The subject is male, 28 years old, 1.78 m and 85 kg. In that case study, the maximum torque is 41 Nm. Therefore, the simulation is configured to get closer to the real case.

The codes were developed in the LUA language and were divided into two main scripts. The threaded-child script and the non-threaded child script. The main difference between both scripts is how each one runs during the simulation. The non-threaded child script runs just once after the start and when is called by other function, such as the press of a button, in this script are found the codes referring to the development of the control interface and all functions linked to it. The threaded script runs in loop during the entire simulation, and it contains the codes about the movement, the data acquisition, the messages that must be sent to another script, and the API connection.

The control interface is developed in XML and was used to configure the main parameters in the simulation, as the maximum torque of shoulder, the target position for the shoulder's joint (degree), the resistance torque of all UR3 joints, the possible failures in the movement, and other parameters related to communication with external API. The link with the external API (MATLAB), is built to execute the self-control module. To establish communication, V-REP must be configured in synchronous mode. Thus, MATLAB will execute a part of its code and send a trigger to V-REP, in turn, the simulator executes one step and returns to MATLAB the requested value. The simulation step is 100 ms. The variables returned to MATLAB are the position and the current force measured through the force sensor. The connection persists until the simulation stop.

3.2 Self-control Module Approach

The self-control module is developed to improve the usability and facility of the system; the code of this control was implemented in the MATLAB language. The communication between MATLAB and VREP is exemplified by the Fig. 2 through a block diagram. When the autonomous control is active, the robotic arm is able to recognise the force of the patient and change joint's torque values, making the system responsible, in real-time, by itself and adaptable to any patient. Besides, as the premises of the system work with the patients that are capable of performing some upper limb's movement and the biofeedback is the force of this movement, no path planning is required on the robot's side, but the therapists shall indicate to the patient which are these motions.

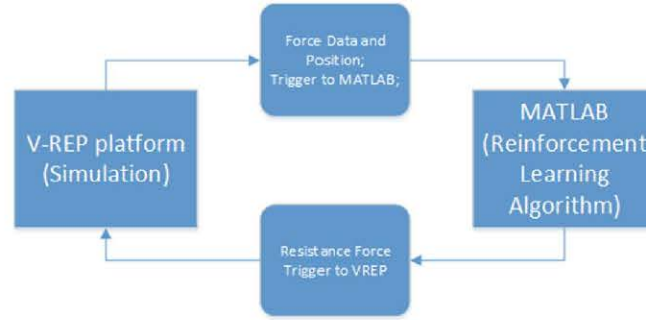


Fig. 2. Block Diagram of the proposed simulated system

The self-control module is developed based on a machine learning technique (RL). The RL is used to make the system learns with the environment, and it is desired that the robot performs a resistance against the human arm force. As presented in [17], basically the algorithm is separated in states and actions. In this case, the states are all forces performed by the human arm, measured through the sensor attached in the UR3's end-effector. The actions are represented by the possible decisions that the system can make, working with the resistance, the decisions are: increase the force of all joints of UR3, decrease the same variable or hold this.

The goal of an RL algorithm is to choose the best decision based on the system's current state. However, this behaviour is not known at first; therefore, the method of trial-and-error is performed. Initially the system is in some state s_x , makes a decision (some action a_x is performed) and goes to another different state s_y . When the system act in the environment some feedback signal is collected and this value is used to evaluate the action a_x in the certain state s_x . So, if the action results in an expected state (considered positive for the system), then the algorithm assigns a positive reward to the set "previous state - action - next state", else the reward is negative to the same set. The positive reward, in this work, will be awarded when the force measured by the sensor is within a range of 70% to 85% of the maximum force the patient can perform, as the simulation is based on the experiments accomplished by [16], the positive reward is assigned each time the force sensor reads a value between 29 and 35 Nm. If the arm presents some dysfunction in the movement and also presents a reduced force, the system did not assign positive rewards.

Thus, the rewards acquired are used to calculate a matrix, named Q-matrix, which relates the states with the actions. Therefore, if an action a_x in the state s_x get many positive rewards, the Q-matrix value corresponding to this pair (state, action) will increase. To make a decision, the algorithm searches in the matrix which action has a larger value in the current state and chooses this. After the occurrence of a few number of episodes, the value for correct actions in Q-matrix will be larger than the wrong actions and, thus, completing the learning of the system. The episodes in this approach are the set of iterations that comprehend an entire movement of the human arm, i.e., the movement of

raise up and bring down the limb represents an episode. As the system learns by itself, it is not necessarily a human control over the joints, completing the objective of providing a resistance force to movement without supervision.

4 Results

This section explains the obtained results by simulation performed with the self-control module activated and deactivated. The presented values refer to the movement of the arm and the robot. The simulation is developed to comprehend some lacks in the rehabilitation field, being a tool to test the system before its practical application and help the therapist learn about the utilities of the robotics applied to this problem.

To help in the usability, an interface is built to control the entire simulation and presents some data related to the movement. Figures 3a and b show the tabs “General Settings” and “MATLAB Communication”, sequentially. The first tab configures the main parameters linked to human arm movement (1 and 2), the resistance torque of the UR3 (3), and the possible variation of the patient force, which can be random (button “random failures”) or given by the user (4 and 5). When the button “Add New Values to Arm” is pressed, the simulation starts and the table is filled with the configured parameters. The second tab, called “Control Panel”, presents some results of the movement as torque of the robot’s joints and torque of the shoulder. The second tab is showed in Fig. 5. The third is responsible by the communication with the external API, basically when the button “A” is pressed the simulator is able to send and receive signals from MATLAB. If the button “B” is activated, the V-REP send a request to the external API to initiate the self-control module.

To test the simulated system through the interface, the values considered in the arm are from [16], thus the maximum torque of shoulder is 41 N.m. The failures in the movement are set to occur in 17% of the path and stop in 42% and are presented as a decrease in force that the shoulder is capable of performing. A value of 6 N.m is configured to be the maximum torque that the robot joints can execute in opposition to movement. Figures 4a and b show the movement of the robotic arm linked to the human, then Figs. 5a and b show some values related to the performed movement.

In the occurrence of a failure in the movement of the patient’s arm, immediately the force sensor records this information. As the robot dynamically change its position, should be considered the force over the three Cartesian axes XYZ , therefore, it is shown in Fig. 6 the resultant force in the movement. The movement has as reference the human shoulder, so the x-axis in the image shows the angulation of it, so the failures occur between 300° (17%) and 315° (42%). The other variations of the value measured by the sensor happen due to the direction of the movement, i.e., the value of force in the end effector will be different according to the angulation of the UR3’s joints and human arm.

When the self-control module is active, the human arm will execute the movement of raise up, using the shoulder as the main joint. The movement of

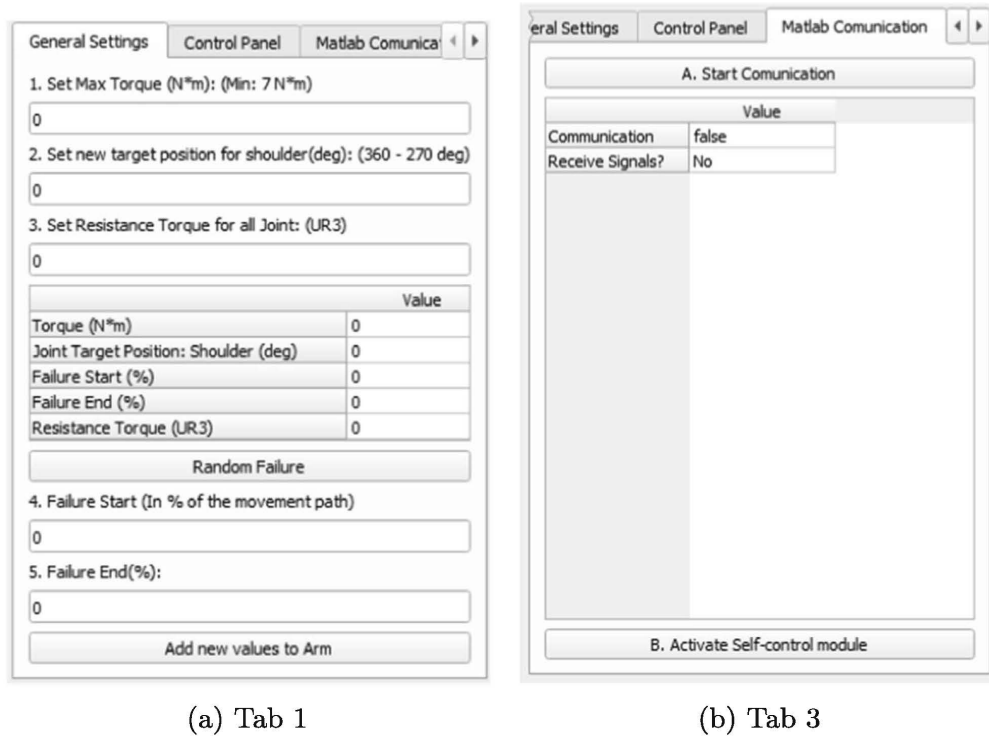


Fig. 3. Human-Machine interface for simulation control

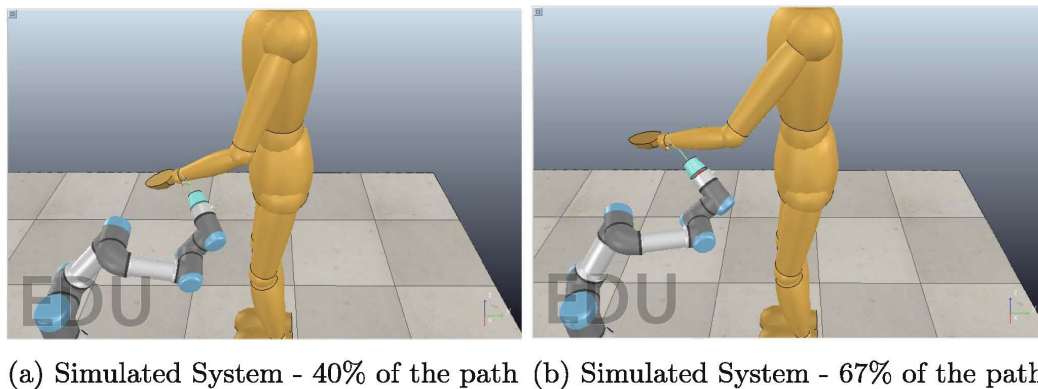


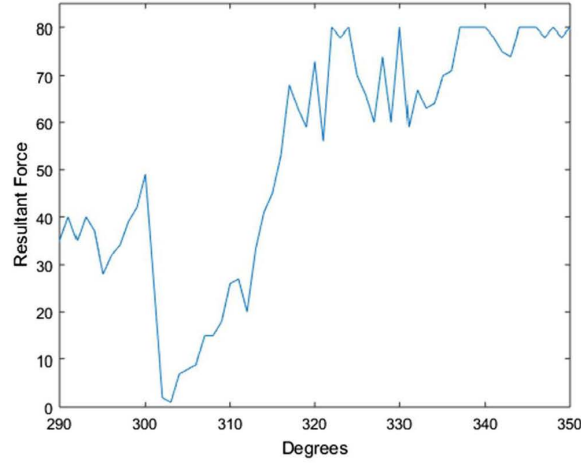
Fig. 4. Simulated Movement of human arm

the human arm pulls the robot attached in the wrist. However, the raising up of arm suffers a resistance force provided by the link with the robotic system, and this value is acquired and sent to MATLAB. The objective of the self-control algorithm is to provide resistance to the movement. If the force of human arm decreases, also the opposite force should decrease, then the total range of movement is reached. The algorithm also should be able to identify the improvement of the patient and provide a larger force in order to correct the variations in the movement of the arm.

General Settings		Control Panel	Matlab Communica
		Value	
Joint Torque: Shoulder(N*m)		41	
Joint Torque: Elbow(N*m)		24.242820739746	
Joint Position: Shoulder (deg)		335	
Joint Position: Elbow (deg)		326	
UR3 Torque: Joint1 (N*m)		2.6670660972595	
UR3 Torque: Joint2 (N*m)		-2.6541259288788	
UR3 Torque: Joint3 (N*m)		-3.7374215126038	
UR3 Torque: Joint4 (N*m)		1.669199347496	
UR3 Torque: Joint5 (N*m)		5.4660406112671	
UR3 Torque: Joint6 (N*m)		10.536356925964	
Force Sensor: X (N*m)		37.997840881348	
Force Sensor: Y (N*m)		-48.321956634521	
Force Sensor: Z (N*m)		40.695529937744	

General Settings		Control Panel	Matlab Communica
		Value	
Joint Torque: Shoulder(N*m)		41	
Joint Torque: Elbow(N*m)		19.747577667236	
Joint Position: Shoulder (deg)		320	
Joint Position: Elbow (deg)		326	
UR3 Torque: Joint1 (N*m)		2.6864399909973	
UR3 Torque: Joint2 (N*m)		-1.6248977184296	
UR3 Torque: Joint3 (N*m)		-1.7394469976425	
UR3 Torque: Joint4 (N*m)		4.8537645339966	
UR3 Torque: Joint5 (N*m)		5.4401502609253	
UR3 Torque: Joint6 (N*m)		11.092317581177	
Force Sensor: X (N*m)		52.030006408691	
Force Sensor: Y (N*m)		-22.969562530518	
Force Sensor: Z (N*m)		58.828010559082	

(a) Data - 40% of the path (b) Data - 67% of the path

Fig. 5. Data related to movement**Fig. 6.** Failures in the force of Movement

With the data acquired to the contact with the patient, the RL algorithm updates its values and find a better resistance force that the UR3 should perform, then this information return to V-REP through the synchronous communication and the UR3's maximum torque is changed. This process occurs in loop until the simulation stop or the maximum number of iterations in the RL algorithm is reached. This value of resistance torque is given as a percentage of maximum force that human arm can execute.

In the first episode, the behaviour of the measured force is not constant and can vary a lot, however after a few number of episodes this force tends to be constant, in Fig. 7 the acquired force is shown, after a few number of episodes.

As mentioned before, the RL insertion of the autonomous mode works with the update of a maximum torque in the collaborative robot in order to correct the resistance force. This update is shown in Fig. 8, and as expected in the first

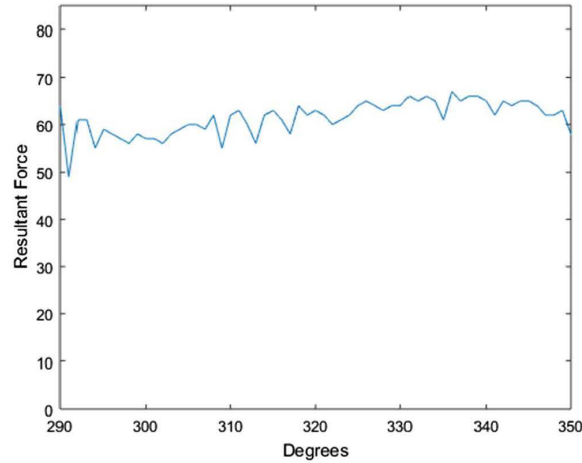


Fig. 7. Resultant Force with the insertion of self control module

episode, the torque increase until a value and stands around it, remaining in this condition in the other episodes. The maximum torque applied in the test was set to 5 Nm because if the value were larger, the arm could not move.

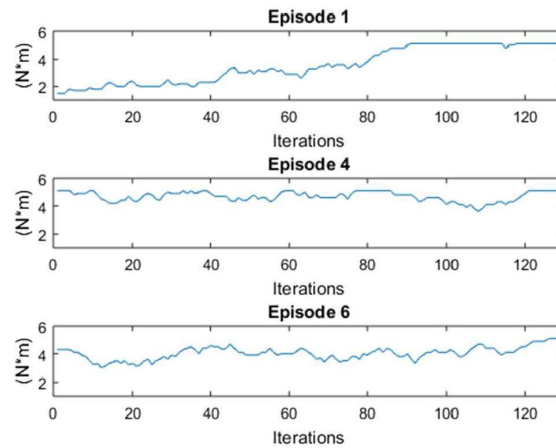


Fig. 8. Maximum Torque Update of UR3 in some episodes

Lastly, it is relevant to show the learning of the self-control module. That behaviour of the system can be presented through the average reward obtained over the episodes. This information represents the evolution of the system after a number of episodes and can be seen in Fig. 9, note that in figure the y-axis represents the average reward to a set of 10 episodes and the x-axis exposes this set, i.e., the first value in the x-axis represents 10 episodes and so on.

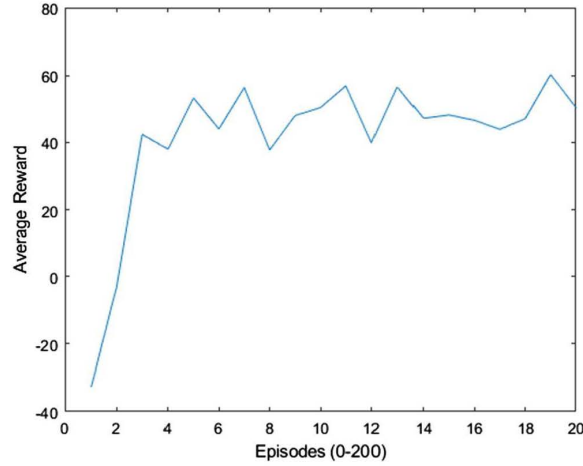


Fig. 9. Average reward of the RL algorithm

5 Conclusion and Future Works

This paper presented the used of a collaborative robot (UR3) as a robotic rehabilitation system, the simulation of it and the insertion of an autonomous mode in a virtual system via an external API. The main contribution of this work is the possibility of test the system with a model of human contact before its application. Besides, the insertion of a self-control module removes the need for the robot's path planning and its configuration to each patient. In this case, the dynamic simulation can provide a great gain for therapists, because it helps in the learning of the proposed system, besides it allows the therapists to test new methods in the simulation environment.

Another important point to consider is that simulation can be performed before the application in a real environment, thus, for new technologies, it is a way to identify possible failures and make adjustments, principally when this technology is applied together to human touch. Following the execution of this work, some relevant aspects emerge for a more detailed approach. In the simulation, the sensor force presents some noise, because of the simulation engine, which is used the Bullet 2.83 engine. Therefore, it is desired to create a model closest to the real component. For the simulation environment, it would be interesting to implement a real model of shoulder that would bring a better understanding of this approach. For a complete study, the algorithm will be implemented in a real robotic system to evaluate its performance and compare the measured data in the real environment with the simulation to verify which variables are influenced.

References

1. Rehabilitation 2030: A call for action plan: Then need to scale up rehabilitation (2017)
2. Chatterji, S., Byles, J., Cutler, D., Seeman, T., Verdes, E.: Health, Functioning, and disability in older adults - presents status and future implications. *Lancet* **385**(9967), 563–575 (2015)
3. Union Européenne des Médecins Spécialistes (UEMS) e Académie Européenne de Médecine e Réadaptation: Livro Branco de Medicina Física e de Reabilitação na Europa. Sociedade Portuguesa de Medicina Fisica e de Reabilitação, Coimbra (2009)
4. Turolla, A.: An overall framework for neurorehabilitation robotics: implications for recovery. In: *Rehabilitation Robotics*, pp. 15–27. Elsevier (2018)
5. Dijkers, M., deBear, P., et al.: Patient and staff acceptance of robotic technology in occupational therapy: a pilot study. *J. Rehabil. Res. Dev.* **28**(2), 33–44 (1991)
6. Novak, D., Riener, R.: Control strategies and artificial intelligence in rehabilitation robotics. *AI Mag.* **36**(4), 23–33 (2015)
7. Yap, R., Kono, D., et al.: Development of wereable gait assist robot using interactive motor rhythmic stimulation to upper and lower limbs. *AI Mag.* **36**(4), 23–33 (2015)
8. Casadio, M., Sanguineti, V., et al.: Braccio di Ferro: a new haptic workstation for neuromotor rehabilitation. *Technol. Health Care* (14), 123–142 (2006)
9. Amirabdollahian, F., Taylor, M., et al.: The Gentle/S project: a new method of delivering neuro-rehabilitation. *Assistive Technology - Added Value to the Quality of Life* (10), 36–41 (2001)
10. Kemna, S., Culmer, P., et al.: Developing a user interface for the iPAM stroke rehabilitation system. In: *IEEE International Conference on Rehabilitation Robotics*, Kyoto, Japan (2009)
11. Hogan, N., Krebs, H.I., Charnnarong, J., Srikrishna, P., Sharon, A.: Mit-manus: a workstation for manual therapy and training. I. In: *Proceedings IEEE International Workshop on Robot and Human Communication*, pp. 161–165. IEEE (1992)
12. Reharob: Reharob (2000). <http://reharob.manuf.bme.hu>. Accessed 06 May 2019
13. Loureiro, R., Harwin, W., et al.: Advances in upper limb stroke rehabilitation. *Med. Biol. Eng. Comput.* **49**(1), 1103–1118 (2011)
14. Coppelia Robotics. <http://www.coppeliarobotics.com>. Accessed 17 June 2019
15. Coppelia Robotics Homepage: Max. joint torques - 17260. <https://www.universal-robots.com/how-tos-and-faqs/faq/ur-faq/max-joint-torques-17260/>. Accessed 02 Sept 2019
16. Ribeiro, D.C., Estivalet, M.G., Loss, J.F.: Modelo para estimativa da força e torque muscular durante a abdução do ombro. *revista portuguesa de ciências do desporto* **8**(3), 321–329 (2008)
17. Sutton, R., Barto, A.: *Reinforcement Learning: An Introduction*, 2nd edn. A Bradford Book, London (2015)