Manipulator for upper limbs rehabilitation- ROBCO®

Nayden Chivarov¹,² and Denis Chikurtev¹,²

1 Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., block 2, Sofia, Bulgaria
2 European Polytechnical University, Pernik, Bulgaria

nshivarov@code.bg, dchikurtev@gmail.com

Abstract: This article presents a manipulator for upper limbs rehabilitation Robco®. The manipulator will be able to allow rehabilitation of the shoulder, elbow and wrist. Manipulator’s hardware, software and web-based user interface have been researched and developed to allow the required functions and properties for rehabilitation procedures. Usability experiments have been performed to investigate the functionalities. The results demonstrate precise positioning and repeatability. Finally, we can conclude that the rehabilitation of upper limbs can be robotized, which will reduce time and costs of the rehabilitation procedures.

1. Introduction.
Technological progress poses a number of challenges to medicine. On the way to making progress, a number of treatments have undergone qualitative changes that necessitate a redefinition of the classic methods of treatment and rehabilitation and their adaptation to the new social order [1].

In recent years, robotics has been entering the medical field in various areas [2, 3]. With the implementation of Robots, a variety of manipulations and operations are performed, helping to reduce recovery time. One of the areas where robots are used is rehabilitation [4, 5].

This article addresses the issues of robotic upper limb rehabilitation, using the advances in modern robotic technology.

It is of utmost importance that the principle of controlled autonomy of robots is respected, according to which the initial planning of the treatment and the final decision for its implementation remain always at the discretion of the physician/rehabilitator.

Medical robots will continue to advance precision surgery and repeat procedures and manipulations, and will have the potential to improve rehabilitation outcomes and provide highly efficient logistical support in hospitals [6]. There is great progress and additional potential in the field of exoskeletons that can be worn on the human body and serve both for the rehabilitation and support of the elderly and the disabled [7].

Medical robots have the potential to reduce health care costs by enabling healthcare professionals to focus their attention from treatment to prevention and by providing more budgetary resources to better adapt to the diversity of patients’ needs for ongoing care training of medical specialists and research [8].

2. Manipulator for upper limbs Rehabilitation- ROBCO®
The Robco upper limb rehabilitation manipulator (Fig. 1) has 2 degrees of freedom, with a stroke volume from -30 to 180°. It is controlled by PC and microprocessor controller, programming the parameters of the device (extension, flexion, pause, power and speed). It is intended for rehabilitation of the shoulder joint, elbow joints and wrist. The manipulator is versatile and is designed for both left and right hands. Used in a seat position.

The main parameters of the manipulator are the workspace, speed, time, control mode and power.
The angular velocity of joint rotation can be varied on a scale of 1 to 9 as follows:
- at first speed the movement is carried out by 0,5°/s, i.e. from -30 to 180 °, run for 7 minutes;
- second: 0,9°/s, for 3 minutes and 8 seconds;
- third: 1,45°/s, for 2 minutes and 41 seconds;
- forth: 1,75°/s, for 2 minutes;
- fifth: 2,15°/s, for 1 minute and 37 seconds
- sixth: 2,5°/s, for 1 minute and 24 seconds;
- seventh: 2,9°/s, for 1 minute and 12 seconds
- eight: 3,3°/s, for 1 minute and 3 seconds
- ninth: 3,75°/s, for 56 seconds

The device has a torque function that indicates the power of operation:
- large - actuators run at 100% power. The system only monitors the trajectory of the movement and seeks to overcome any resistance encountered during movement. In this setting, if a spasm occurs during the therapy process, the system will not respond and will strive to overcome the change in resistance. This can lead to trauma, and therefore is monitored under medical supervision.
- medium - actuators operate at 65% of their capacity and are able to stop and reverse the direction of movement when a resistance greater than engine power occurs.
- small - the smallest power represents 35% of the actuator capacity. When a resistance greater than the power of the device occurs, it will stop movement. This type of adjustment is the most sensitive and guarantees the safety of therapy in patients with contractures, increased muscle tone. For the convenience and safety of the patient, the manipulator has a remote control, which can be stopped and started at any time.

The manipulator has 2 degrees of freedom of rotation type. The manipulator type is "SCARA" [13]. The purpose of the developed model is to allow rehabilitation of the upper limbs and more precisely the shoulder, elbow and wrist.

Figure 1 shows the construction of the robotic manipulator. It consists of 3 units, base, which is static, and two active joints. A specialized pad is mounted on the first joint, on which the user places the forearm. At the end of the second joint, a handle is mounted which user hold with his palm.

![Figure 1. Manipulator for upper limbs Rehabilitation- ROBCO ®.](image)

The maximum rotation angles, which are set as end positions, are: for joints 1 from -120 to 60 and for joints 2 from -60 to 30. The following restrictions are specified to guarantee user safety:
If $\varphi > 45$, then $\theta < 10$;
If $\varphi < -45$, then $\theta > -10$;

Where $\varphi$ is the angle at the first joint of the manipulator and $\theta$ is the angle at the second joint (see figure 2). These restrictions are introduced to limit the movements of the manipulator so that movements that endanger the patient are not performed.
3. **Manipulator Hardware System.**

The manipulator is powered by two stepper motors. Depending on the drive gear for first joint, the ratio of the angle of rotation to the number of steps is 1: 3, and for second joint it is 1: 2. In order to operate stepper motors, proper drivers for them have been selected. The motor drivers connect to an Arduino controller that is connected to a computer [9]. Drivers must be provided with additional power according to the controllers used, while the Arduino controller is powered by the computer. The schematic diagram of the connection is shown in Figure 3.

![Schematic diagram of Hardware connections.](image)

Since the manipulator is driven by stepper motors, we rely on A4988 drivers to control them [10]. These drivers allow to limit the current to the motor. This is of great importance when it comes to operating a device that will work with humans. In addition, these drivers allow the motor steps to be divided into 1/2, 1/4, 1/8, and 1/16. Thus, accurate positioning can be achieved. The choice of driver is intended to demonstrate the drive mode and capabilities of the robotic manipulator. In fact, most stepper

![Figure 2. Limitations of rotation angles – maximum end angles.](image)

**Figure 2.** Limitations of rotation angles – maximum end angles.
motor drivers are controlled by the same principle as the one chosen. The main parameters are on / off, step resolution, number of steps and direction of rotation. From a hardware standpoint, these drivers require a minimum of three controller signals: on / off, number of steps, and direction. Additional signals are used for setting the resolution of the steps. An enable input expects a logical 0 or 1, thus activating or deactivating the driver itself. Impulses are used to determine the number of steps as input. Each impulse responds in one step. And the direction input expects a logic 0 or 1 signal. To divide the steps into micro steps, the other 3 inputs (MS1, MS2 and MS3) must be connected to the controller. The steps are set according to the manufacturer's description. These inputs also expect to receive logical 0 or 1. If we have activated the half-step mode, then the number of pulses transmitted corresponds to the number of micro-dull ones.

Arduino controllers are among the most common for controlling robots and other mechatronic systems. They are easy to program and are affordable. In addition, they provide many options for working with different sensors and devices as they have analogue and digital input / output pins [9].

For our purposes, to manage two drivers completely, i.e. to use all their functionality, we need 12 digital input / output pins. This means that we can use an Arduino Uno controller that has 13 such pins. Each pin of the controller must be connected to the corresponding driver pin as shown in Figure 2, and the controller will be connected via USB port to a computer.

The role of the computer is to enable the control parameters of the robotic manipulator and the user interface to be set up and actuated. The scheme thus presented distributes the computing resources between the controller and the computer. For maximum speed, the controller executes trajectories and accelerations calculated from the computer, and in fact takes care of motor control only. And the computer is engaged in complex calculations such as calculating the speed of motion, calculating a trajectory, setting the number of steps to execute, and more. In addition, a user interface will be implemented on the computer, through which the rehabilitator will be able to set parameters for each procedure.

4. **Manipulator Software Architecture.** Software control is divided into 3 levels. At the low level is the motor control by the controller. The intermediate level is provided by ROS [11]. And the high level is a web-based user interface. This type of structure allows remote control of the manipulator via the web. In this way, the parameters of the individual procedures can be set remotely.

4.1. **Arduino software.** The low-level software is tasked with executing the received commands from the top level. It receives procedure data such as rotation angle and rotation speed for each joint, then transforms this data into motor-driven data. The rotation angle for each joint is determined by calculating the required number of steps and / or micro steps.

\[
\text{Steps} = \frac{\deg \times \text{motor}\_\text{steps} \times \text{microsteps}}{360}
\]

Where steps are the necessary steps to reach the set angle, deg is the set angle, motor_steps are the number of steps for one complete rotation of the motor and microsteps is the coefficient according to the microstep mode.

The speed of rotation is determined by calculating the pulse rate in milliseconds at a given rpm.

\[
\text{Step\_pulse} = \frac{60 \times 1000000}{\text{motor\_steps} \times \text{microsteps} \times \text{rpm}}
\]

Where motor_steps are the number of steps for one complete rotation of the motor, microsteps is the coefficient according to the microstep mode and rpm is the desired rpm.

4.2. **ROS.** This level makes the connection between the low level and the user interface. The parameters specified in the interface are processed and sent for execution to the controller [12]. Here the correct execution of the tasks is monitored, the number of repetitions is counted and new parameters are dynamically set. In addition, a stop command can be sent to the controller at any time during operation and to terminate the current procedure.
In figure 4 is illustrated the general software communication system. The hardware interface between the computer and the controller is USB, using a dedicated library for communication and work with Arduino. Web interface is running on the robot computer and uses a specialized ros-bridge-suite library for communication through web ports.

5. **Web-based User Interface.**
In order to adequately use the manipulator, a sufficient set of options should be provided for adjusting and monitoring the rehabilitation procedures. We have developed a user interface that will allow physicians and patients to get as much opportunity as possible for easy operation of the manipulator. In figure 5 are shown the features and options provided by the web-based user interface.

![Web-based User Interface](image)

**Figure 5.** Web-based user interface.
We can divide the main interface menu into five sub-menus. Starting from the top down, these are: connection and emergency stop, settings and timing of the procedure, adjusting the maximum joint angles and managing the procedure.

This arrangement is intended to minimize the likelihood of errors occurring when a new procedure is initiated. The first submenu is represented by two buttons and an empty box where a checkmark appears when the interface has contacted to the manipulator. The connection button is blue and provides a function for connecting and interrupting communication with the robot controller. The emergency stop button instantly interrupts the robot's electric motors. This ensures that the robot does not injure or endanger the health of the patient.

The procedure setting submenu contains buttons for setting the number of cycles and rotation speed. As to set the selected values from the fields, the set buttons must be pressed. In addition, there is a drop-down menu that shows a calendar with a schedule. Here you can trace the history of procedures and upcoming ones. The last menu is for timing and time display.

Entering and storing schedules in the calendar and timer is done using a database where the entered information is stored. These databases are part of the user interface and are not related to the direct operation of the manipulator.

The next submenu is to adjust and set the maximum end angles of rotation. These are the restrictions we set in section 2. Even if the doctor tries to set values that do not meet the set limits, the software will not allow and will set the maximum values according to the restrictions.

There is submenu for power setting. The power can be set to small, medium and large using corresponding button. The active button is colored green after pushing “Set Power” button.

When you press the buttons to set the selected values in the two submenus, these values are sent to the ROS nodes. They, in turn, generate the corresponding parameters for submission to the controller. Changing these parameters is dynamic, but again it is forbidden to change these parameters while the current procedure is running. To change the values, the procedure must be terminated or paused.

Going through all the settings for one procedure follows the submenu for managing the procedure. There are buttons to start, pause and stop the current procedure. In addition, there is a bar that displays the progress of the procedure and the time remaining until its completion.

6. Experiments and Results.
A number of experiments have been conducted to test how the manipulator and control systems perform their work. Initial tests were conducted without a hand on the manipulator. The purpose was to check the correct execution of the tasks without risk to the user. Then, the same experiments were repeated, but this time with the motors loaded, a hand placed by the user. The experiments involved performing three different configurations to reach maximum angles, each test being tested at different speeds from 1 to 9. All experiments were performed for 10 replicates.

During the experiments, it is checked that each joint reaches the specified rotation angle and what is the deviation from the assignment. In addition, the total run time of 10 repetitions is measured and compared to the theoretically calculated times according to the speed selected. The other checks are for the functions of the buttons and whether the system responds adequately when the corresponding button is pressed from the user interface.

The results of the precision positioning experiments show a deviation at idle of 1-2 degrees and with the arm set 4-7 degrees. The results of the speedometer experiments show a minimum delay or acceleration of 1-2 seconds when idling. When working with a raised hand, the difference in speed is within 3-4 seconds. The results of the button functionality tests show that the system reacts instantly when the emergency stop button is activated and the other buttons function normally.

7. Conclusion.
Rehabilitation manipulators can offer significant improvements in the quality and results of rehabilitation procedures. Research in this field can contribute to the creation of modern and smart rehabilitation robots. The researched manipulator and the developed systems for its management mark the successful start of research on manipulators and systems for automated rehabilitation.

Studies show that the model developed can offer good and reliable rehabilitation procedures. In addition, remote control and monitoring capabilities extend its applications and functionality and create great convenience.
The foreseen future work on improving and developing such a rehabilitation manipulator is as follows. Development of a system and algorithm for automatic "STOP" if there is a risk of injury to the patient. Buttons for switching to the previous or next procedure. Preview the set procedure on a special display. Adding a submenu for pre-saving and selecting procedures. Adding a degree of freedom to the wrist rotation. Add translation of elbow pad to adjust to patient need.

Acknowledgment.
This research was carried out as part of the project “Telecontrolled Service Robots for Increasing the Quality of Life of Elderly and Disabled, No. DN 07/23 – 15.12.2016”, financed by the Bulgarian National Science Fund. The work was partially supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Young scientists and postdoctoral students” approved by DCM # 577 / 17.08.2018.

References.
[1] Oujamaa, L., I. Relave, J. Froger, D. Mottet, and J-Y. Pelissier. "Rehabilitation of arm function after stroke. Literature review." Annals of physical and rehabilitation medicine 52, no. 3 (2009): 269-293.
[2] Taylor, Russell H., Peter Kazanzides, Gregory S. Fischer, and Nabil Simaan. "Medical robotics and computer-integrated interventional medicine." In Biomedical Information Technology, pp. 617-672. Academic Press, 2020.
[3] Petrescu, Relly Victoria, Raffaella Aversa, Antonio Apicella, and Florian Ion Petrescu. "Future medicine services robotics." American Journal of Engineering and Applied Sciences 9, no. 4 (2016): 1062-1087.
[4] Krebs, Hermano I., Mark Ferraro, Stephen P. Buerger, Miranda J. Newbery, Antonio Makiyama, Michael Sandmann, Daniel Lynch, Bruce T. Volpe, and Neville Hogan. "Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus." Journal of neuroengineering and rehabilitation 1, no. 1 (2004): 5.
[5] Van der Loos, HF Machiel, David J. Reinkensmeyer, and Eugenio Guglielmelli. "Rehabilitation and health care robotics." In Springer handbook of robotics, pp. 1685-1728. Springer, Cham, 2016.
[6] Peter S. Lum, Charles G. Burgar, Peggy C. Shor, Matra Majmundar, Machiel Van der Loos, Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke, Archives of Physical Medicine and Rehabilitation, Volume 83, Issue 7, 2002, Pages 952-959, ISSN 0003-9993, doi.org/10.1053/apmr.2001.33101.
[7] Lu, Renquan, Zhijun Li, Chun-Yi Su, and Anke Xue. "Development and learning control of a human limb with a rehabilitation exoskeleton." IEEE Transactions on Industrial Electronics 61, no. 7 (2013): 3776-3785.
[8] Ryan A. Beasley. “Medical Robots: Current Systems and Research Directions”, Journal of Robotics, Volume 2012, doi:10.1155/2012/401613.
[9] Warren, John-David, Josh Adams, and Harald Molle. "Arduino for robotics." In Arduino robotics, pp. 51-82. Apress, Berkeley, CA, 2011.
[10] Robotic, Pololu. "Electronic. A4988 Stepper Motor Driver Carrier." Pololu Robotics & Electronics.
[11] Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In ICRA workshop on open source software, vol. 3, no. 3.2, p. 5, 2009.
[12] Zubrycki, Igor, and Grzegorz Granosik. "Introducing modern robotics with ros and arduino, including case studies." Journal of Automation Mobile Robotics and Intelligent Systems 8, no. 1 (2014): 69-75.
[13] Cox, Damon Keith, Marvin L. Freeman, Jason M. Schaller, Jeffrey C. Hudgens, and Jeffrey A. Brodine. "Methods and apparatus for extending the reach of a dual scara robot linkage." U.S. Patent 8,016,542, issued September 13, 2011.