Internship Experience for Learning the Operation of a Cable-Driven Robot for Rehabilitation Tasks

G. Carbone[1(✉)], D. Cafolla[1], M. Ceccarelli[1], O. Aydinoglu[1,2], and M. Demirel[1,2]

[1] University of Cassino and South Latium, via G. Di Biasio 43, 03043 Cassino, Italy
{carbone, cafolla, ceccarelli}@unicas.it,
[2] Izmir Institute of Technology, 35430 Urla, İzmir, Turkey
{ondercanaydinoglu, muratdemirel}@std.iyte.edu.tr

Abstract. This paper describes an example of internship experience at LARM, Laboratory of Robotics and Mechatronics, at University of Cassino and South Latium. In particular, the main focus of this paper is the learning process as basis of a proper internship. Namely, a specific experiential learning approach is proposed as referring to a low-cost servo motor operation task. The learning task is further defined and developed as referring to a real application with a cable-driven robot for rehabilitation tasks, which has been designed and built at LARM in Cassino and under further investigation within Agewell project.

Keywords: Education · Lab. training · Experimental robotics
Service robotics

1 Introduction

Academic internships are part of the field of experiential education aiming to apply classroom learning, theories, and experiences to professional settings. They are usually linked to an undergraduate curriculum and include reading reference sources, writing a report, critical thinking, and, especially, a real world problem solving while applying concepts from the classroom, [1].

LARM, Laboratory of Robotics and Mechatronics is an internationally recognized research center having international collaborations and exchange programs with several countries worldwide, [2]. Several academic internships have been successfully carried out throughout the last twenty years. This paper aims to describe the learning process as basis of a proper academic internship by referring to the specific experience of the first author in his internship period at LARM during the second semester of the academic year 2016–2017. The internship has been following the attendance of classrooms delivered by Prof. Ceccarelli and Prof. Carbone, which provided the necessary background in mechanisms theory and robotics, respectively.

J. C. García-Prada and C. Castejón (Eds.): *New Trends in Educational Activity in the Field of Mechanism and Machine Theory*, MMS 64, pp. 195–207, 2019.
https://doi.org/10.1007/978-3-030-00108-7_22

The learning process is further detailed in the following sections to show how a real world complex task for the operation of a cable driven robot for rehabilitation tasks has been achieved within a short eight weeks internship period.

2 Internship Arrangements

The internship has been developed in several phases as described in the scheme of Fig. 1. The first phase has been the definition of specific problem to be addressed. Namely, it has been decided the main task as being the control and operation of a set of low-cost servomotors. Then, as second phase some research has been carried out to identify the know-how and theoretical aspects of the proposed task. As third phase, specific low-cost hardware has been identified. Namely, an Arduino board has been selected, since it is very versatile and requires basic level programming and electronics skills, [3]. Similarly, specific servo-motors and electronics have been identified. Fourth phase has been learning by practicing and interactions with staff at LARM on how to integrate the selected hardware components to achieve a proper operation and control of the servo-motors. Finally, the experience learned with the above-mentioned components has been applied to CALOWI 2 (Cassino Low-Cost Wire Driven Robot version 2). This is a cable-driven robot for rehabilitation tasks, which has been designed and built at LARM in Cassino and it is under further development within the Agewell project in collaboration with Technical University of Cluji-Napoca, Romania. The specific used version of CALOWI 2 has three active degrees of freedom to be carefully synchronized to achieve a proper robot operation, [4].

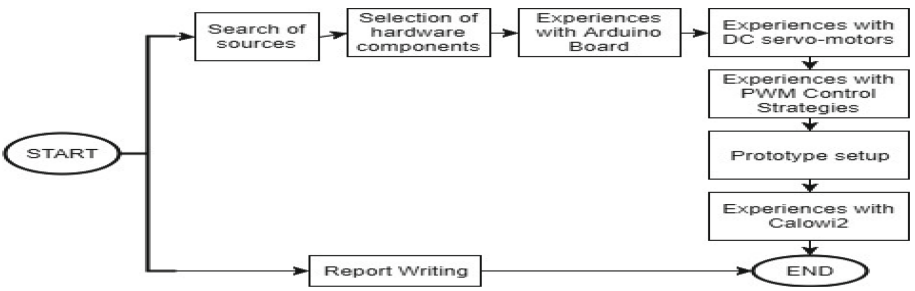

Fig. 1. A flow-chart of the learning process.

3 Arduino Control Board

3.1 Main Features

One of the main focus of the internship has been getting experienced with Arduino board, which has been chosen as the core control unit of the used prototype for this educational purpose. Arduino is a microcontroller which can be used standalone or in combination with other hardware to achieve the control of some device acquiring

inputs, elaborate them, take decisions due to condition in the code and, if necessary give some output as feedback for the control loop. In this paper, an Arduino Uno has been selected and used. Arduino Uno is based on the ATmega328, [5], it has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and other features are shown in Fig. 2, [6].

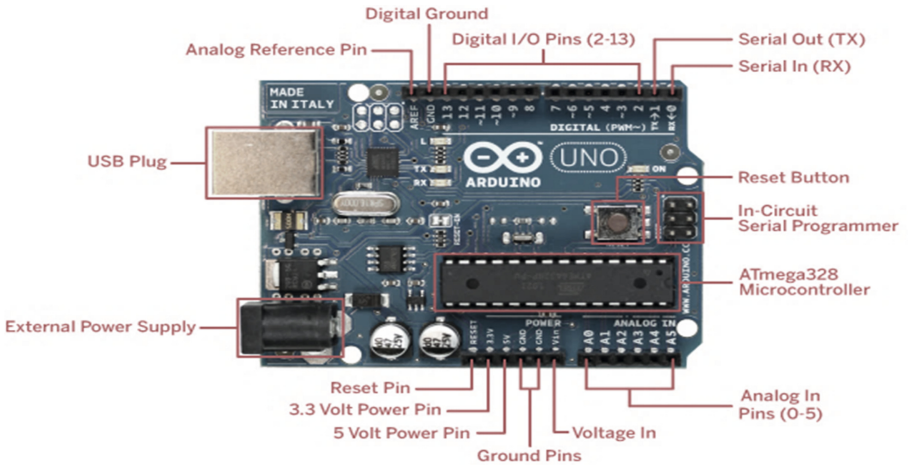

Fig. 2. A scheme of Arduino Uno, [5].

3.2 Serial Communications

Arduino can be interfaced with a PC through an Usb port. Arduino can either send commands or receive data via serial port. According to this, Arduino can receive sensors outputs and send values of variables, control robots movements and debugging program codes. Serial communication works on 1 s and 0 s (bits). These highs (1) and lows (0) bits form together and turn into bytes, after that they can be converted to ASCII encoded symbols and letters, [7].

Figure 3 shows a flow-chart for serial communication in Arduino environment.

At first Arduino's baud rate is set using the Serial.begin() function. Serial.available () checks if there is any data in the Serial port buffer. If so, Serial.read() function is used to read the data in the serial buffer. Finally, Serial.print() and Serial.println() functions visualize in the Serial monitor interface whatever has been stored in the buffer. These functions can help the debug procedure to check visually inputs and outputs, and communication between the control unit and the user, [8].

3.3 Data Parsing

Data parsing is useful when sending several information at the same time on a buffer is needed, for example a vector of position that an actuator has to reach such as in Fig. 4. The parsing procedure will analyze this flux of data and assign it to the needed

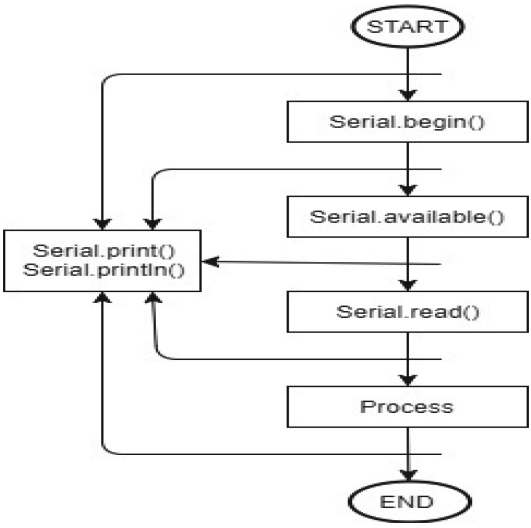

Fig. 3. An example flow-chart for serial communication.

variables. For this project, the parsing method has been chosen synchronizing the motion of different servo-motors. The positions are sent in a data flux, divided and converted to integers using .toInt() command, [9]. An example of data parsing vector position is shown in Fig. 4 where 30,60,415 instruction stands for a target position of 30°, 60°, 415° for motor 1, motor 2 and motor 3, respectively.

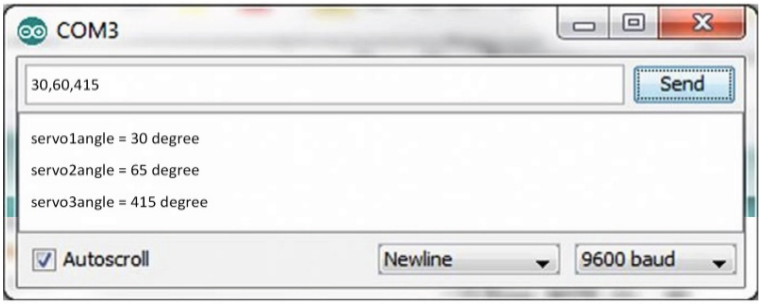

Fig. 4. An example of data parsing.

4 Using Servo-Motors

A further step during the internship has been the implementation of a control strategy to move the chosen servo-motors to achieve the experiments target.

A servo-motor is a DC motor which is controlled for specific angular rotation with the help of additional servomechanism by means of a closed loop feedback control

system. Servo motor applications are seen in remote controlled servomechanisms for controlling the direction of motion. Furthermore, they can provide angular precision, [10]. A servo-motor presents three cables namely, brown, red and yellow. The brown cable must be connected with the common ground (GND), the red one must be connected with the power supply (5 V) and the yellow one must be connected with the control signal, which is provided through a pulse width modulation (PWM), [5].

4.1 Pulse Width Modulation (PWM) Strategy

Servo control is achieved by sending to a servo a pulse-width modulation (PWM) signal. Namely, a series of repeating pulses of variable width where either the width of the pulse or the duty cycle of a pulse train determines the position to be achieved.

When a square wave is sent, 5 V is applied in the "on" position; 0 V is applied in the "off" position. The width at which the "on" part is active is called "Pulse Width", [11]. The parameters to set the pulses are the minimal pulse width, the maximal pulse width, and the repetition rate or duty cycle. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential mechanical rotation in the clockwise direction as it does in the counter-clockwise direction. Usually, the neutral position is related to 1.5 ms pulse width. A graphical example of PWM is shown in Fig. 5.

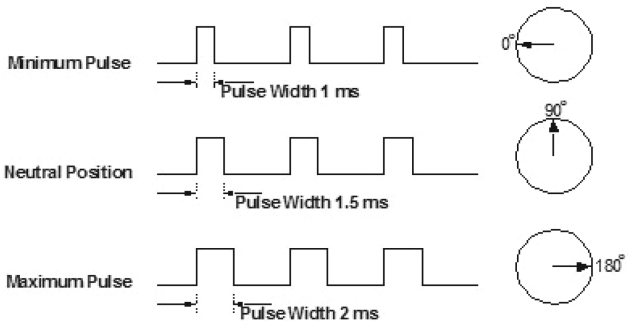

Fig. 5. An example of PWM signals with different duty cycles.

4.2 Testing Servo-Motor Rotation

Experimental activity has been carried out to operate three HS-785HB servo motors by using an Arduino Uno board. As a first step both motors data sheets and real PWM scales are checked with the help of an Arduino code. This Arduino code has been set up by using the commands in Sect. 3 (Sending data, PWM, Serial Monitor). It is to note that servo-motors usually have an angular motion range of $\pm 180°$. HS-785HB servo motors have an angular motion range of $\pm 1332°$. Max PWM Signal Range, Max Travel Angle and No-Load Speed at 4.8 V are taken from datasheet [12] and compared with real values as reported in Table 1.

Table 1. Comparing Datasheet and Real values.

	Datasheet	Real
Max PWM Signal Range	600–2400 µs	615–2415 µs
No-Load Speed (4.8 V)	1.68 s/360°	1.57 s/360°
Max Travel Angle	2826°	2664°

4.3 Adafruit 16-Channel Servo Driver

Microcontrollers such as Arduino board usually have a limited number of PWM outputs and a limited computational power. If a prototype presents too many actuators, signal problems can occur. To overcome these problems, an Adafruit 16-Channel 12-Bit PWM /Servo Driver has been selected and used in this project in cooperation with an Arduino Uno board. With this hardware it is possible to obtain up to 16 PWM outputs using only 2 pins on the Arduino controller. This hardware solution keeps other pins available for scaling up the hardware, for example, to increase the number of controlled servo-motors of for the operation of other devices. Furthermore, Adafruit 16-Channel 12-Bit PWM /Servo Driver uses an I2C protocol, which allows a common clock and synchronous operation of all the connected servos. Accordingly, Adafruit_PWMServoDriver.h and Wire.h libraries have been used instead of Servo.h Arduino library, for an optimal communication between the servo driver. Moreover, the function pwm.setPWM(pin number, ON, PWM) has been used to send the position to the actuators, [13].

Figure 6 shows an example of how Arduino, a servo-motor, the Adafruit servo driver and power supply are connected. It is important to note that Adafruit 16-Channel 12-Bit PWM /Servo Driver also simplifies the cabling of servo-motors, since a single power supply unit is needed for all the connected servos.

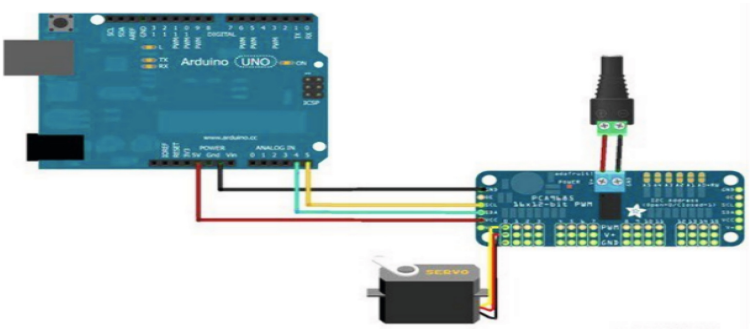

Fig. 6. Adafruit Servo Driver servo-motor connection example, [13].

5 Synchronized Control of the Servo-Motors

Robot manipulators are composed by links and joints. Links are the rigid members connecting joints or axes. The axes are the movable components of the robotic manipulator that cause relative motion between interconnected links, [14]. Manipulators must have synchronized motion of each joint to achieve a proper target operation task.

Experimental tests have been carried out aiming to achieve a synchronized control of three servo-motors with different angular configuration targets. At first a calibration procedure has been carried out to map the servo-motors in Arduino environment. Namely, servo-motors max and min values have been set as 158 and 660 PWM, respectively. Moreover, 0.18 ms PWM pulse width has been set as equals to 1° of rotation. Starting position has been set in the middle of the servo-motor motion range. 0° corresponds to a PWM of 400 and the range is between +1332° and −1332°.

The developed program code includes three different angles that are defined as servo1angle, servo2angle and servo3angle.These angles are converted to PWM values, which are defined as servo1pwm, servo2pwm and servo3pwm by using Eqs. (1–3).

$$\text{servo1pwm} = (\text{servo1angle} * 18)/100 + 400 \tag{1}$$

$$\text{servo2pwm} = (\text{servo2angle} * 18)/100 + 400 \tag{2}$$

$$\text{servo3pwm} = (\text{servo3angle} * 18)/100 + 400 \tag{3}$$

There are two possible cases for servo1pwm. It can be servo1pwm > 400 and servo1pwm < 400. This means that user command to servo to rotate clockwise or counterclockwise. For identifying these two cases, the developed program code includes two if loops. In each if loop there is a for loop for step counting. Each for loop determines how many steps are needed to complete the desired motion with constant PWM. This constant is 1° = 0.18 PWM to achieve a smooth motion. In the same for loop number of steps for the second and third servo-motors will be set as the same of servomotor1 but angular step rotation will be different according to factor 2 and factor 3, respectively. Equations (4–5) can be used for finding the step rotation of second and third motors.

$$\text{factor2} = (\text{servo2pwm} - 400)/((\text{servo1pwm} - 400)/0.18) \tag{4}$$

$$\text{factor3} = (\text{servo3pwm} - 400)/((\text{servo1pwm} - 400)/0.18) \tag{5}$$

The programming process is also described in the flow-chart shown in Fig. 7 where three servo-motors can reach different target angular positions at same time with same step number. In addition, in this code data parsing method has been used to receive a vector of servo rotation angles from the serial monitor interface.

In one experiment angles were defined as servo1angle = 90°, servo2angle = 180° and servo3angle = −90°. In their direction servo 1 rotates about 90° clockwise, servo 2 rotates about 180° clockwise, servo 3 rotates about 90° counter clockwise. Figure 8 shows the synchronous motion experiment target positions.

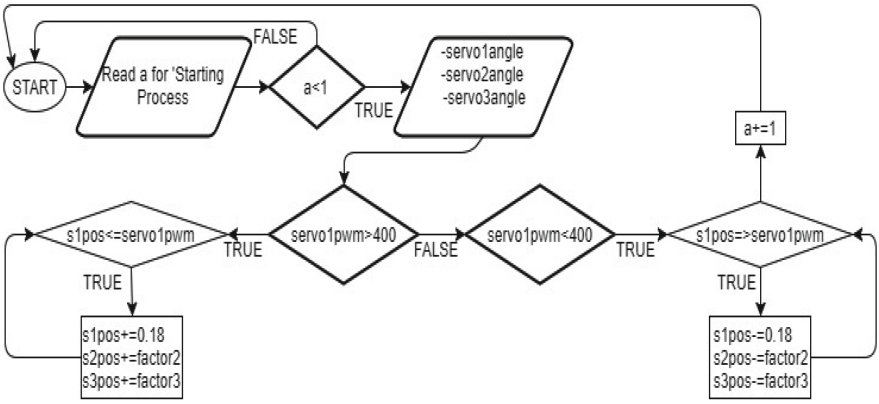

Fig. 7. A flow-chart of the operation of three servo motors.

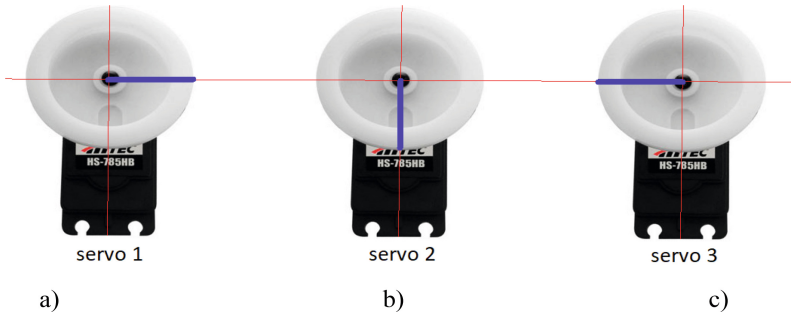

Fig. 8. Three servo-motors with position marks: (a) servo 1 at 90° clockwise position; (b) servo 2 at 180 clockwise position; (c) servo 3 at −90° counter clockwise position.

6 A Case of Study with CALOWI 2

CALOWI 2 robot is a cable-based parallel robot for rehabilitation, which has been designed and built at LARM in Cassino, [4]. The structure of CALOWI 2 is a non-conventional open architecture, which allows an easy accessibility by patients under treatment. Aluminum profiles are used so that the robot is light and stiff while it can be assembled and disassembled easily for storage and transportation. Cables are connected to the end-effector which covers the arm to be trained by using a wristband, [4]. During the rehabilitation, an arm needs elbow support. On CALOWI 2 robot has three servo motors and three cables, which are connected to manipulators and the end-effector through a wristband as shown in Fig. 9. Cable tensioning is guaranteed by gravity. The success of a rehabilitation process requires several key factors. Among them, it is necessary to identify and perform a reliable and safe motion training protocol. For the purpose, several experiments have been carried out to set up proper motion parameters including but not limited to the elbow support position and arm motion ranges. Some tested setting conditions are shown in Fig. 10.

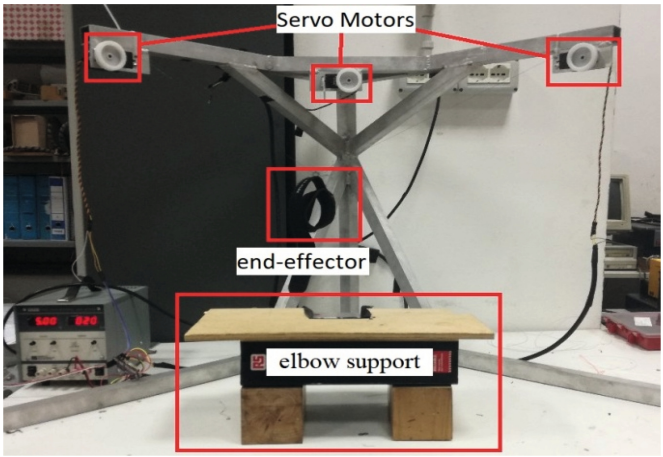

Fig. 9. A picture of CALOWI 2 cable-driven parallel robot at LARM.

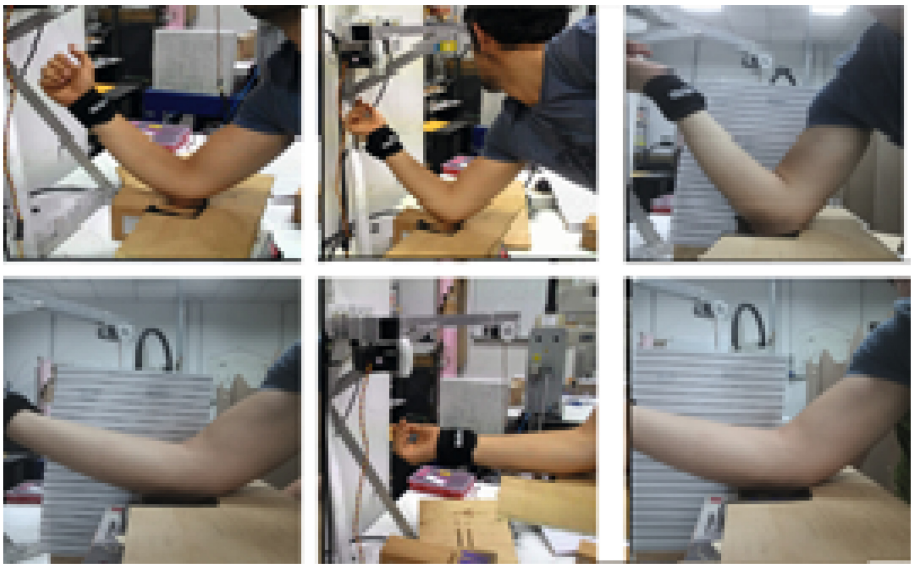

Fig. 10. Several position testing for identifying most favorable arm supporting configuration.

6.1 Programming CALOWI 2 Operation

Sections 3 and 4 have defined the main hardware and software features for the operation of the three DC servomotors, which are used to operate CALOWI 2. As main operation tasks it has been decided to achieve two operation tasks consisting in assisting the motion of a human arm during up-down and left-right motions. Motion distances are constant for each motion.

A specific user interface has been designed on a laptop screen and keyboard interface. At beginning the screen shows a message 'Which motion?' followed by 'How many cycle?' and then a 'Starting process' button becomes available. The first command is achieved through a for loop. There is an integer which is defined as 'a' variable. If this integer equals to 1, the motion will be left-right, else if equals to 2 the motion will be up-down. There is also an integer variable called as 'c'. If c equals 1 the process will start. Another integer variable 'b' is defined to determine the cycle number and this value depends on user input with three possible cases (c = 1; c <= b; c += 1). The three servo-motors start to move in middle angle position (s1pos = s2pos = s3-pos = 400 ms PWM signal). After that, s1pos increases or decreases with constant value (0.18 ms PWM signal), meanwhile s2pos and s3pos increases or decreases with their factors. These factors (factor2, factor3) are calculated at beginning of the algorithm. A detailed programs flow-chart is shown in Fig. 11.

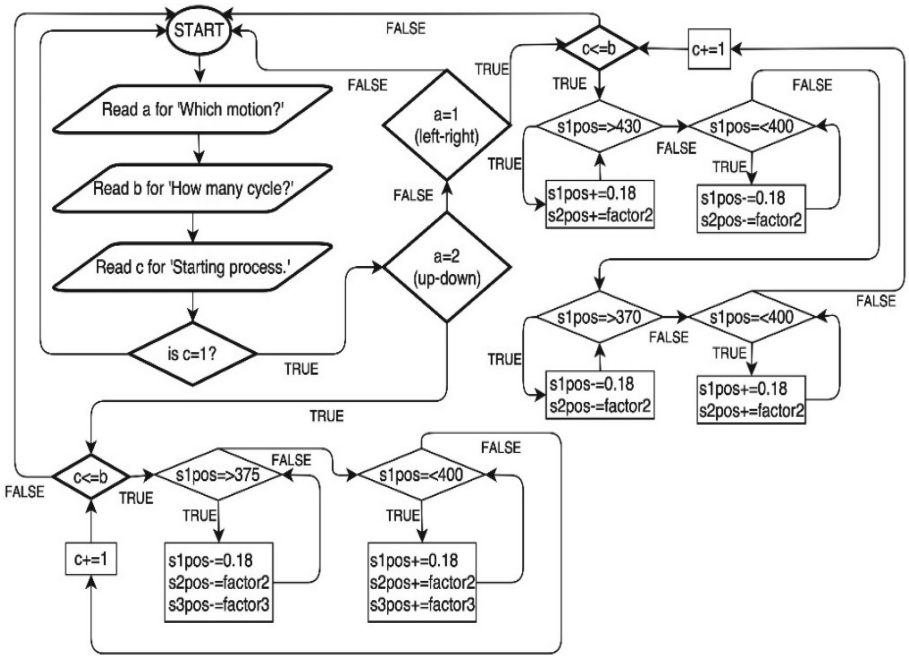

Fig. 11. A CALOWI 2 operations flow-chart.

6.1.1 Left and Right Motion

One cycle of left and right motion takes four second. The motions start position is at middle configuration. End-effector goes to the left position (10 cm) in one second and turns back to start position finally end-effector goes to the right position (10 cm) and turn back to start position. Experimental tests have been first carried out without connecting CALOWI 2 to a human in order to verify a suitable safe operation is successfully shown in Fig. 12. Then, experiments have been taken by assisting the motion of a human arm as successfully shown in Fig. 13.

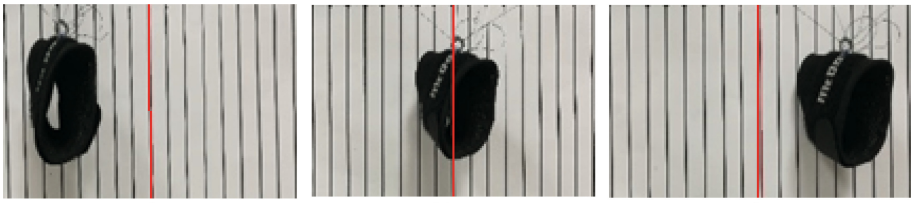

Fig. 12. Experiment of left and right motion without hand: (a) t = 1 s left side; (b) t = 0 s initial position; (c) t = 3 s right side.

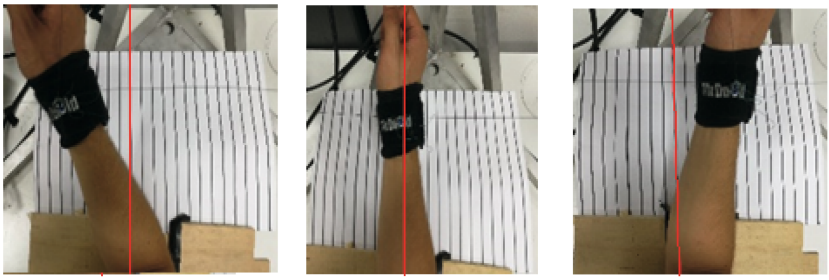

Fig. 13. Experiment of left and right motion with hand: (a) t = 1 s left side; (b) t = 0 s initial position; (c) t = 3 s right side.

6.1.2 Up and Down Motion

One cycle of down and up motion takes 90 ms. The motions start position is on middle. End-effector goes to the up position (15 cm) in 75 ms and turns back to the start position by taking the same amount of time. Experimental tests have been first carried out without connecting CALOWI 2 to a human in order to verify a suitable safe operation is successfully shown in Fig. 14. Then, experiments have been taken by assisting the motion of a human arm as successfully shown in Fig. 15.

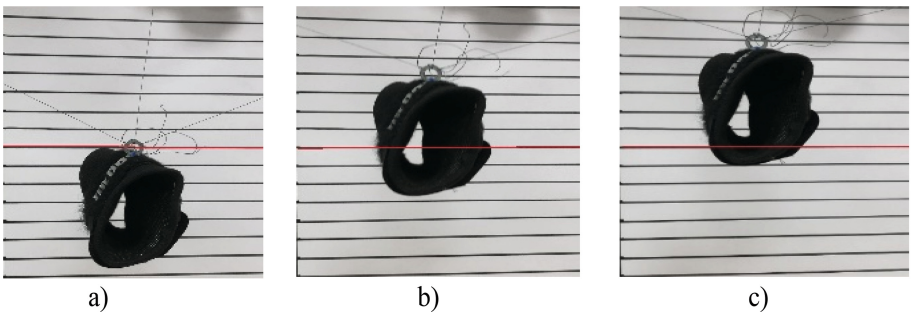

Fig. 14. Experiment of up and down motion: (a) t = 0 s initial position: (b) t = 0.5 s third point; (c) t = 0.75 s fourth point.

a) b) c)

Fig. 15. Experiment of up and down motion: (a) t = 0 s initial position; (b) t = 0.5 s third point; (c) t = 0.75 s fourth point.

7 Conclusions

This paper describes an academic internship which has been successfully carried out at LARM. Main aim of this paper is to show the learning by experience process, which is achieved during an internship. The reported case of study shows how it has been possible to achieve the complex task of operating CALOWI 2, a cable driven robot for rehabilitation tasks, within a short eight weeks internship period.

Acknowledgements. The paper presents results from the research activities of the project ID 37_215, MySMIS code 103415 "Innovative approaches regarding the rehabilitation and assistive robotics for healthy ageing" co-financed by the European Regional Development Fund through the Competitiveness Operational Programme 2014–2020, Priority Axis 1, Action 1.1.4, through the financing contract 20/01.09.2016, between the Technical University of Cluj-Napoca and ANCSI as Intermediary Organism in the name and for the Ministry of European Funds.

References

1. Merritt, R.D.: Student internships. EBSCO Research Starters: Academic Topic Overviews, pp. 1–8 (2008)
2. Ceccarelli, M.: An Illustrated history of LARM in Cassino. In: International Workshop on Robotics in Ale-Adria-Danube Region RAAD 2012, Napoli, pp. 35–42 (2012)
3. Arduino Homepage: Arduino guide. https://www.arduino.cc. Accessed 8 Sept 2017
4. Carbone, G., Gherman, B., Ulinici, I., Vaida, C., Pisla, D.: Design issues for an inherently safe robotic rehabilitation device. In: 26th International Conference on Robotics in Alpe-Adria-Danube Region RAAD 2017, pp. 967–974. Springer, Dordrecht (2017)
5. Banzi, M., Shiloh, M.: Getting Started with Arduino: The Open Source Electronics Prototyping Platform. Maker Media Inc., Sebastopol (2014)
6. Balogh, R.: Educational robotic platform based on Arduino. In: 1st International Conference on Robotics in Education RiE2010, FEI STU, pp. 119–122 (2010)
7. Hisham, A.A.B., Ishak, M.H.I.: Bluetooth based home automation system using an Android phone. Jurnal Teknologi (Sci. Eng.) **70**(3), 57–61 (2014)
8. Margolis, M.: Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects. O'Reilly Media Inc., Beijing (2011)

9. Thu, S., Tin, H.H.K.: Construction of Android and Arduino for home appliances switching system. Int. J. Eng. Appl. Sci. Technol. **1**(11), 66–71 (2016)
10. Sawicz, D.: Hobby servo fundamentals. Princeton ebook. https://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf. Accessed 8 Sept 2017
11. Sparkfun Homepage. https://learn.sparkfun/PWM. Accessed 8 Sept 2017
12. Servocity Homepage. https://www.servocity.com/hs-765hb-servo. Accessed 8 Sept 2017
13. Adafruit Homepage. https://cdn-learn.adafruit.com. Accessed 8 Sept 2017
14. Ceccarelli, M.: Fundamentals of Mechanics of Robotic Manipulation. Springer, Dordrecht (2004)