CHAPTER 3

Exoskeleton for passive rehabilitation: ExoFlex

3.1 Overview

An exoskeleton for elbow and shoulder passive rehabilitation was developed. This exoskeleton was called ExoFlex. In this section, the exoskeleton designed for passive rehabilitation is presented. It is important to remark that the exoskeleton to be presented is active, although it is intended for passive rehabilitation therapy.

Passive rehabilitation is applied to patients at early stages of their recovery process. In this kind of therapy, subjects leave their arms completely relaxed and the physical therapist (PT) mobilizes their joints. Exoskeletons can be used to perform these passive mobilization routines according to predefined trajectories defined by the PT.

The main purpose of this rehabilitation therapy is to start mobilizing the articulations as soon as possible after the injury so that joint spasticity and muscular atrophy are prevented [22]. Some pathologies of traumatic origin such as frozen shoulder or rotator cuff tendinopathy, require that the recovery of functional capacity begins early. However, in other pathologies such as dislocations, fractures and arthroplasties, conducting an initial period of immobilization is crucial, since it helps recovery, reduces edema and pain, prevents radiological deformities and helps consolidation [116].

The design of the exoskeleton is divided in four modules, being three of them tightly coupled: sensing, electronics, software and mechanics. The intersection of sensing, electronics and software modules generates the control of the system. The sensing module obtains data from the mechanics module and the control system (hardware- and software-implemented) generates motion. A diagram of the modules is presented in Fig. 3.1.

ExoFlex was designed to assist flexion-extension of the shoulder, abduction-adduction of the shoulder and flexion-extension of the elbow. The designed soft exoskeleton is cable-driven, i.e., articular motion is generated by the movement of several Bowden cables which connect the actuators to the garment. DC motors with gearboxes were selected as actuators for the system. The cables are attached to the motors via aluminum pulleys coupled to the shafts of the motors. By moving the actuators in different directions, the cables are winded or unwinded over the pulleys (Fig. 3.2). It is important to note that movements in the opposite sense to gravity are directly assisted, as cables only pull the arm of the subject, but they can't push it. Movements in the sense of gravity are partially assisted since, by releasing the cables at a specified speed, these movements can be performed slower than without the device, but not faster than what gravity by itself would generate over the relaxed limb. In other words, if a subject does not leave their arm relaxed, the exoskeleton can move it upwards, but it can't push it downwards.

Although the versions that were designed along the project have noticeable differences, they all are constituted by two basic differentiable elements from a mechanical perspective: the wearable garment, worn by the user, and a fixed frame. The number of actuators and Bowden cables, the electronics system and the nature of the wearable part were redesigned along the different versions. As rehabilitation therapies are performed in a controlled environment, where the user does not have to walk while performing these activities, it was reasonable to get advantage of this situation and embed all the actuation and electronics system in a fixed frame. Not only does this fact ease the design of the exoskeleton, but it also avoids the subject

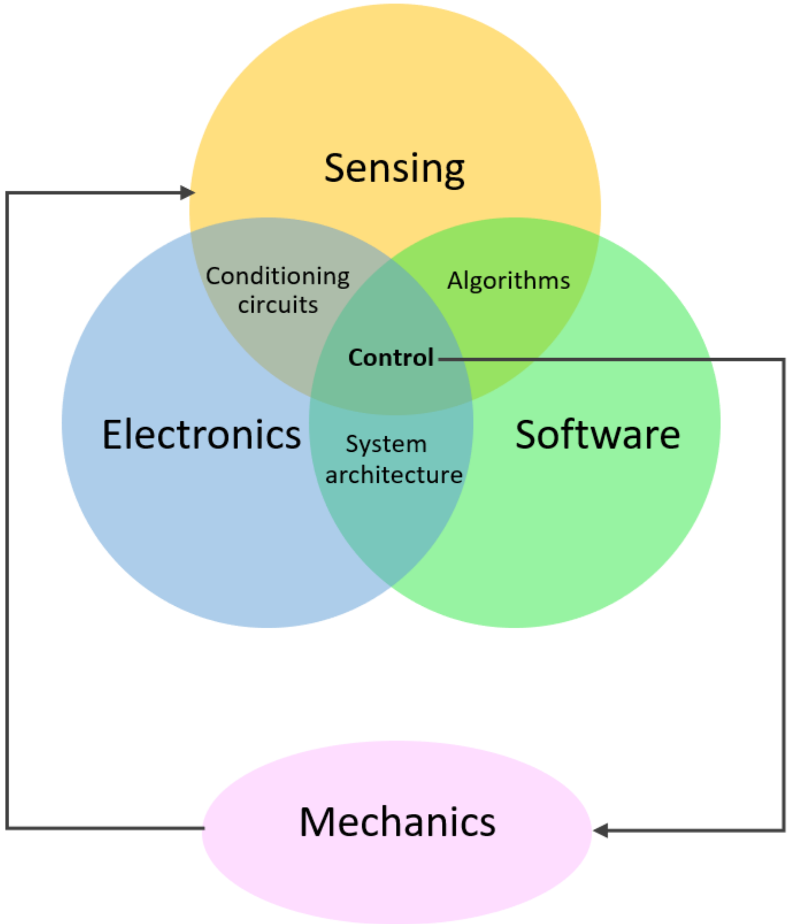


Figure 3.1: Diagram of the modules that constitute the exoskeleton. Sensing, electronics and software are tightly coupled. Sensing module obtains data from the mechanical one. The control system generates actuator motion.

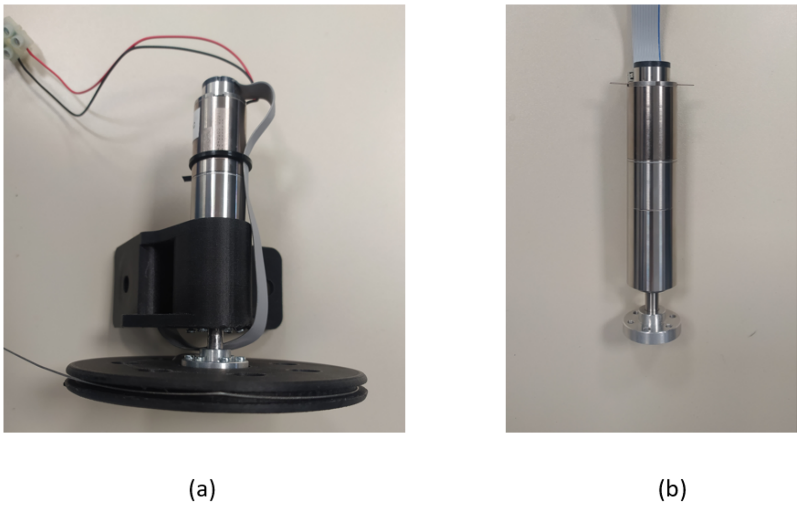


Figure 3.2: DC motors with two coupled pulleys of different radii.

to carry unnecessary weight during rehabilitation, the subject can be comfortably sat during the therapy and, additionally, it reduces the donning and doffing times of the wearable interface. Although there is a rigid structure constituting the exoskeleton, from the user's point of view, ExoFlex is soft and flexible since it does not restrict unassisted movements, it is lightweight and easily wearable. For this reason, the device can be classified as a hybrid exoskeleton, since it is fully wearable in a rehabilitation scenario but, due to the inclusion of the fixed frame, it is not portable by the user.

In regard to the wearable part of the exoskeleton, different textiles, routing strategies for the cables and materials were used during the design process. All these changes were applied according to the observed behavior in experimental tests and the feedback provided by the users.

The design of the textile components of the presented exoskeleton are out of the scope of this thesis. For deeper details in that work, please refer to [117]. The contributions of this work to the mechanical structure of the fixed frame of the exoskeleton were mainly conceptual and qualitative, while the analytical work in that field, was mainly conducted by the research group [118]. The control systems of the exoskeleton (electronics, software, controller and signal processing for sensor data) were developed in this thesis.

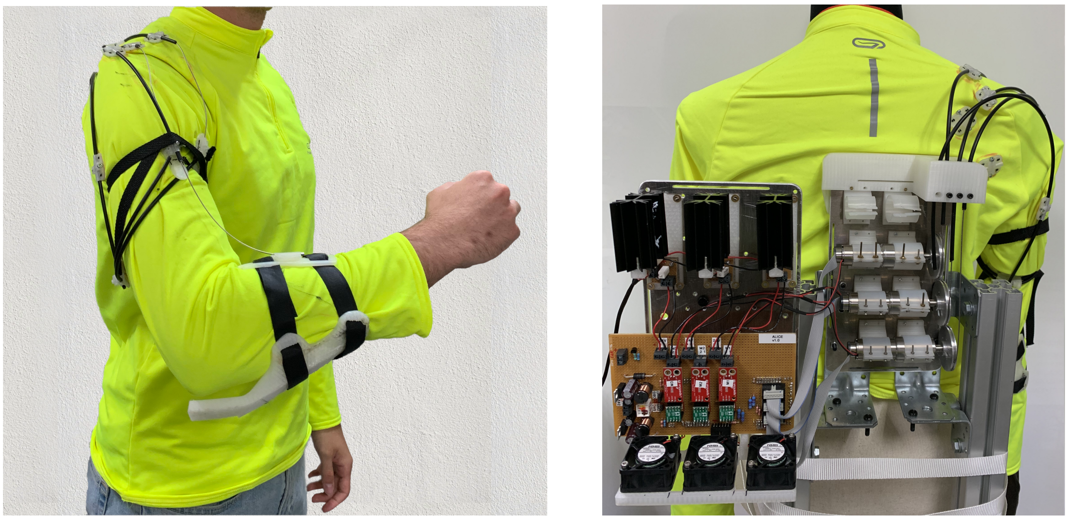
The development process of ExoFlex was iterative and several versions of the exoskeleton were developed until the final one was obtained. During this process, mechanical and electronic engineers collaborated with clinicians to accomplish the requirements that the latter considered of importance. The following subsections provide a general overview of the mechanical, sensing, electronics and software modules.

3.2 Mechanical module

3.2.1 ExoFlex v0

ExoFlex v0 [119] (Fig. 3.3) was the prove-of-concept version. The wearable part of this version was made of a commercial sports shirt, over which some small nylon 3D-printed rigid pieces were attached in strategical points to, in combination with metallic sheaths, route the transmission Bowden cables from the actuators to the garment. Some of those pieces were fixed using Velcro fastenings to adapt to the anatomy of different wearers and others were directly sewn to the fabric. The 3D-printed parts were initially placed to emulate geometry of the tendons of the arms. Starting from that configuration, obtaining the positions and shapes of those parts was the result of an iterative process based on the wearers' experience. In this version, three Maxon 118746 DC motors with planetary gearboxes were used to control three cables, assisting one of each motors shoulder flexion-extension (104 reduction ration), shoulder abduction-adduction (104 reduction ratio) and elbow flexion-extension (53 reduction ratio).

The main problems observed in this version were that very high forces were applied over the trapezius when trying to elevate the arm (this caused pain to the subjects), the edges of some of the 3D-printed parts exerted too high pressures over some of the muscles when shoulder motion was assisted (this generated discomfort) and the elevation angle of the arm was not more than 60 degrees due to the pain and discomfort for the subjects beyond that angle was not tolerable. The flexion movement of the elbow did not generate major discomfort over the users. As seen in Fig. 3.3b, the transmission cables were directly relying over the trapezius. This



(a) (b)

Figure 3.3: ExoFlex v0: (a) exoskeleton worn by a subject, (b) back view, showing the fixed frame embedding the actuation system and electronics. Source: Author's publication ©David Pont Esteban et al. 2019; Published by Springer (https://doi.org/10.1007/978-3-030-36150-1_34).

generated that, when the arm was elevated, great forces were transferred pushing downwards this muscle, which caused pain. Additionally, the lever arm and the incision angle of the cables used to apply torque over the shoulder were minimal. This caused that high tensions had to be applied to the cables so that the arm could be mobilized. These two issues were addressed in ExoFlex v1.

3.2.2 ExoFlex v1

In ExoFlex v1 (Fig. 3.4), both the fixed frame and the wearable part suffered major changes. The textile base was changed for a more compressive commercial cycling jacket. Over the garment, several layers of different textile materials were stacked using force-compliant sewing techniques with the objective of improving fixation, adaption and force distribution. Further details about this textile design are presented in [120]. With this new design, all the 3D-printed parts from ExoFlex v0 were removed, therefore, improving the comfort of the wearer since no rigid parts were pressed onto the wearer's skin. Apart from because of the improvements in the textile couplings, the removal of the rigid parts present in the garment designed for ExoFlex v0 were possible due to the fact that transmission cables in ExoFlex v1 were routed externally to the user's body. In this way, the cables made contact with the users mainly at the anchoring points to the garment.

As observed in Fig. 3.4, a supplementary structure was added to the fixed frame. This new structure accomplishes two objectives. First, it supports the transmission cables for the shoulder so that they do not rest on the trapezius. The elbow cable slightly rests on the trapezius depending on the articular configuration and the subject's complexion, but this does not suppose a burden for comfort since the forces involved to elevate the elbow are much lower than the shoulder ones. In second place, a larger lever arm is provided for arm elevation in flexion and abduction, reducing the necessary forces to elevate the limb. These two aspects eliminate the pain suffered

(a)

(b)

Figure 3.4: ExoFlex v1: (a) exoskeleton worn by a subject, (b) back view, showing the fixed frame embedding the actuation system and electronics. Source: Author's publication ©David Pont Esteban et al. 2022; Published by IEEE (https://doi.org/10.1109/ACCESS.2022.3217528).

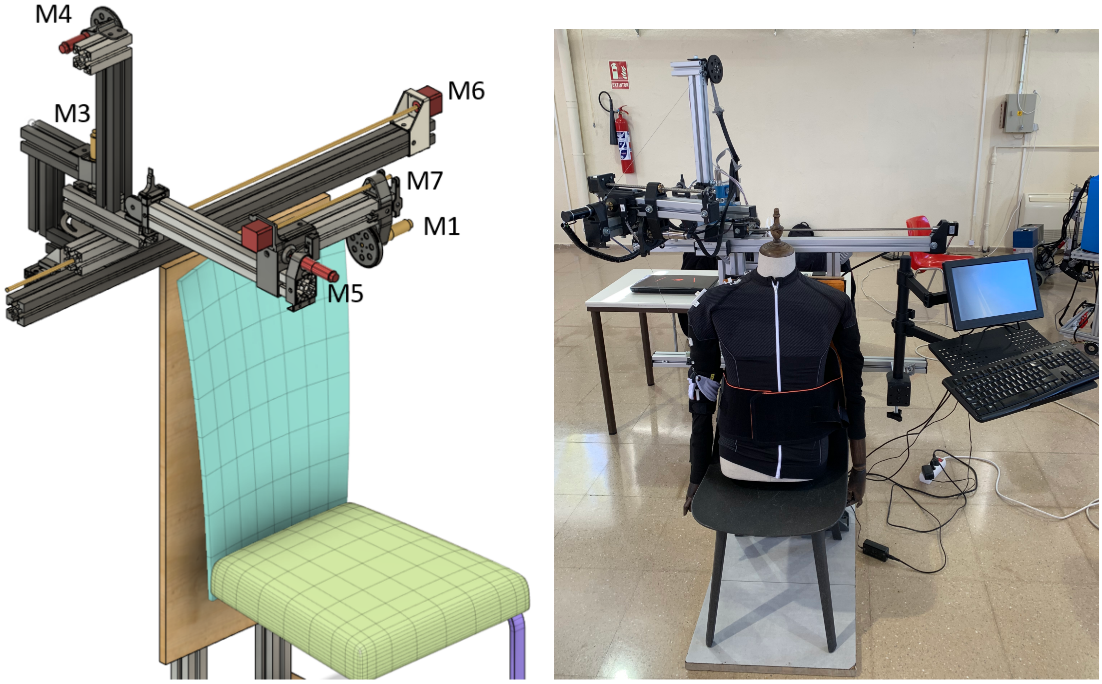
by the subject in ExoFlex v0.

Another relevant aspect introduced in this version is the possibility to easily regulate the height of the structure using the orange cranks in Fig. 3.4b, which allow to better adapt the exoskeleton to the subjects height. The pulleys in Fig. 3.4a can also be manually adjusted to better adapt the user complexion. Two emergency buttons accessible for both the user of the exoskeleton and the supervisor of the therapy were added so that the robot can be immediately stopped whenever it is necessary. The electronics and actuation system were respectively mounted in the left and right boxes in Fig. 3.4b.

This version was qualitatively tested with real patients who suffered rotator cuff tendinitis under the supervision of clinicians. The device proved to work satisfactorily under their criteria, but two main improvements of the exoskeleton were proposed. On the one hand, the maximum elevation angle of the arm (around 110°) was not enough according to the clinicians in order to provide rehabilitation to subjects with a large range of motion (ROM). On the other hand, although the pure shoulder flexion-extension and abduction-adduction movements properly and safely mobilized the arm of the subject, the possibility of elevating the arm with intermediate rotation angles between the two pure movements was suggested to obtain a more versatile rehabilitation tool. These two issues were addressed in ExoFlex v2.

3.2.3 ExoFlex v2

ExoFlex v2 (Fig. 3.5) was mainly focused on providing a wider range of motion for shoulder rehabilitation and to avoid the need to manually adjust the rigid frame to properly adapt to each subject's complexion. The previous passive fixed frame was replaced by a fully automated structure. This new structure constitutes a 5-DoF robot, which enables a whole range of new possibilities. This design is fully



(a) **(b)**

Figure 3.5: ExoFlex v2: (a) CAD design of the exoskeleton, (b) ExoFlex v2 presented in a showroom.

focused on shoulder rehabilitation, leaving the elbow aside. Although the elbow actuation mechanism can be easily added to this version, the research group focused on improving the performance in shoulder assistance, as suggested by the clinicians.

In the previous versions, the actuators were placed in the back of the exoskeleton (Fig. 3.3b and right box in Fig. 3.4b). Cable routing could be manually modified prior to the beginning of a rehabilitation session to better adapt to subject's complexion. In contrast, for ExoFlex v2, the actuator for shoulder motion (motor M1) can be moved along the workspace so that the exoskeleton pulls the cable from different positions and orientations. This enables a much wider variety of movements which the exoskeleton can perform, while increasing the comfort for the subject. However, with this approach, not only the length of the cable has to be controlled, but also the pose of M1. With this exoskeleton, only one cable is used to generate shoulder motion. The capability of the robot to place M1 in different planes allows to just use one cable, since the actuator is able to perform arm elevations in different planes by properly positioning the other actuators.

This version is constituted by two DC motors (M1 and M2) for articular mobilization (one for the shoulder and one for the elbow respectively, although the elbow one was not installed as aforementioned) and three DC motors (M3, M4 and M5) and two stepper motors (M6 and M7) for positioning the shoulder actuator M1. The actuator parameters are shown in Table 3.1. Motor M3 determines the shoulder aperture angle, which is the angle between the sagittal plane and the rod over which M5 is mounted (this rod was called actuation rod). By controlling M3 and the displacement of motors M6 and M7, the elevation plane for the arm is determined. Optical limit switches were added to the frame to perform positioning calibration and as a security measure.

Table 3.1: ExoFlex v2 actuator parameters.

Actuator	Model	Type	Ratio	Sensor
Motor 1	DCX22S GB KL 48V	DC	794:1	Encoder
Motor 2	DCX22S GB KL 48V	DC	326:1	Encoder
Motor 3	DCX22S GB KL 48V	DC	326:1	Encoder
Motor 4	DCX22S GB KL 48V	DC	794:1	Encoder
Motor 5	118746	DC	14:1	Encoder
Motor 6	Nema-17 LD08	Stepper	2 mm/rev	Sensorless
Motor 7	Nema-17 LD08	Stepper	2 mm/rev	Sensorless

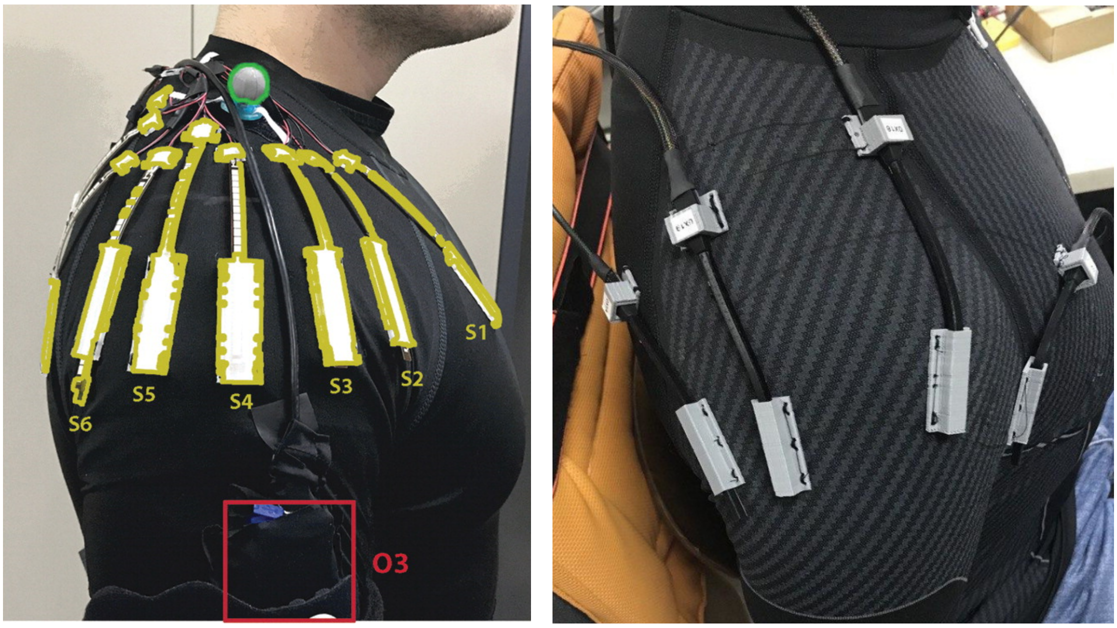
In order to increase the maximum elevation angle, the actuation rod can move up so that space is provided to the arm. Different mechanisms were discussed to elevate the actuation rod and it was decided to implement a cable driven mechanism using motor M4 and a pulley, similar to the mechanisms used to move the human articulations. In contrast to directly installing a motor in the rotation axis of the actuation rod, this solution is bulkier, but the torque requirements for the actuator are reduced, allowing to obtain higher speeds.

One of the drawbacks of the previous version was the short lever arm used to start elevating the shoulder from angles near to 0°. The rod was replaced by a longer one, at the end of which motor M5 was placed with a lead screw mechanism to allow the movement of motor M1 over the rod. This allows to have a longer lever arm than in ExoFlex v1 and a considerably larger initial incision angle between the cable and the user's arm at the start of motion. Depending on the elevation angle of the arm, M5 displaces M1 over the actuation rod so that the angle between the shoulder actuation cable and the arm can be varied. For low elevation angles, M1 is positioned closer to M5, and for high elevation angles, in which the actuation rod is elevated by M4, M1 is positioned further from M5. The kinematics of the system are deeply analyzed in Section 4.3.

In regard to the wearable part of the device, it was simplified to just a commercial orthopedic elbow-pad over which a textile coupling was sewn. The coupling is the same one used for the elbow connection in ExoFlex v1. This design substantially reduces the necessary time to don the exoskeleton and increases hygiene and comfort, since the only contact surface with the user is the elbow. The elbow pad is adjusted to the user via two Velcro straps.

3.3 Motion sensing

A key point to consider in the control architecture of a system is the selection of an appropriate sensory system to monitor all the required variables. Three different aspects should be considered when selecting or designing the sensors for an exoskeleton. First, the sensing frequency should match the controller's needs. Secondly, the noise coupled to the sensor signal has to be evaluated, since the use of a too noisy signal as feedback can negatively impact in the system stability, as well as increase the error. This control sensitivity to noise can be specially critical in SMC [121]. In third place, the location and manner in which the sensors are placed in a soft wearable system has special relevance. These robots are highly deformable,



(a) (b)

Figure 3.6: Soft wearable sensor networks for shoulder motion identification: (a) resistive network. Source: author's publication ©José Luis Samper Escudero et al. 2020; Published by Mary Liebert, Inc. (http://doi.org /10.1089/soro.2019.0040), (b) capacitive network. Source: ©Aldo-Francisco Contreras-González et al. 2020; Published by MDPI (https: //doi.org/10.3390/s20226452).

which hinders the placement of the sensor at a specific desired location, as well as the placement repeatability. Moreover, it might not be possible to rigidly attach a sensor to the exoskeleton due to its soft nature, allowing relative motion between the robot and the sensor, what unavoidably will introduce misleading data in the control loop. Articular position in rigid exoskeletons is normally measured using encoders ([64], [81], [82], [122]). However, other solutions are commonly used for soft exoskeletons, as IMUs [123], flexible resistive sensors [124], silicone stretch sensors [125] or elastomer-based curvature sensors [126].

The research group developed relevant work in upper-limb motion sensing and gesture identification through the use of both resistive ([127], Fig. 3.6a) and capacitive ([128], Fig. 3.6b) flexible sensor networks. However, these networks were not integrated along with the controllers designed in this thesis, as they required long user-specific calibration processes and did not offer a high precision in angular position sensing, although the gesture identification provided good results.

Research in design of motion sensors is out of the scope of this thesis. However, an overview of the motion sensing methods and sensors which were used along with the ExoFlex rehabilitation exoskeleton are presented in this subsection to provide a full view of the exoskeleton. Additionally, a geometrical method which was developed to estimate shoulder orientation is presented in Section 4.3.

3.3.1 Inertial Measurement Unit (IMU)

IMUs represent a widely extended option for angular position and speed measurement in exoskeleton applications [129]–[131]. They have been applied to rehabilitation robots with good and consistent results, however their main drawback is drift [132], although there exist algorithms to minimize this problem [133].

The IMUs used with ExoFlex are LPMS-URS2. These sensors include a gyroscope, an accelerometer and a magnetometer. They allow USB communication and stream data at 400 Hz. They offer a resolution of 0.01° and a static drift of 9 deg/h. IMU calibration, on-body placing and data processing are described in Section 4.2.2.

3.3.2 Angular Displacement Sensor (ADS)

The 2-axis Bend Labs Angular Displacement Sensor (ADS) [134] offers an angular measurement of its bending based on a differential capacitance measurement. The employed model provides information of the sensor bending in the XY and XZ planes. It includes an I^2C interface which streams data at 200 Hz.

Two filters were applied to the sensor raw data. First, the signal was fed to a 50 Hz low-pass second order filter. The processed signal was then fed to a deadband filter (deadband of 0.15°). The second filter removed low-amplitude noise generated when the sensor is not moving. The sensor was attached to the textile interface via Velcro straps and custom-designed 3D-printed parts. This sensor was used to implement an elbow position control using a super-twisting SMC [135].

3.3.3 Motion capture system, OptiTrack

OptiTrack [136] is a marker-based pose-tracker system. It is formed by a group of cameras which can track position markers with a submilimetre precision in the 3D space. OptiTrack was used as position ground truth in the performance of some experiments during the execution of the project, primarily, to evaluate the accuracy of the aforementioned sensor networks and to study human motion.

3.4 Electronics module

An electronics system to control upper-limb exoskeletons was designed. An overview of the system is provided in this subsection. The motivation behind the design of electronics platform is that, in the field of research, when developing a new electromechanical system, actuators often change from one version to another, both in number and type, as well as sensors or communication protocols. Having a flexible electronics platform that can interface several kinds and number of external devices considerably reduces the time it takes to test new variations over the mechanical design, the use of new sensors and the implementation of control techniques.

The electronics system diagram is presented in Fig. 3.7. An architecture constituted by a low-level processing unit and a high-level one was proposed for controlling the exoskeleton. The low-level unit is a Texas Instruments LAUNCHXL-F28379D development board. This board contains the 32-bit dual-core TMS320F2-8379D microcontroller, which runs at 200 MHz. The low-level control of the actuators and interfaces with most of the sensors are implemented over this board. The high-level processing unit is a NVIDIA Jetson Nano. This computer was used to implement a GUI to control the rehabilitation sessions and to monitor the application in real-time. The Jetson Nano sends commands to the TI microcontroller and the microcontroller streams sensors and control data to the Jetson Nano, which are then written to a

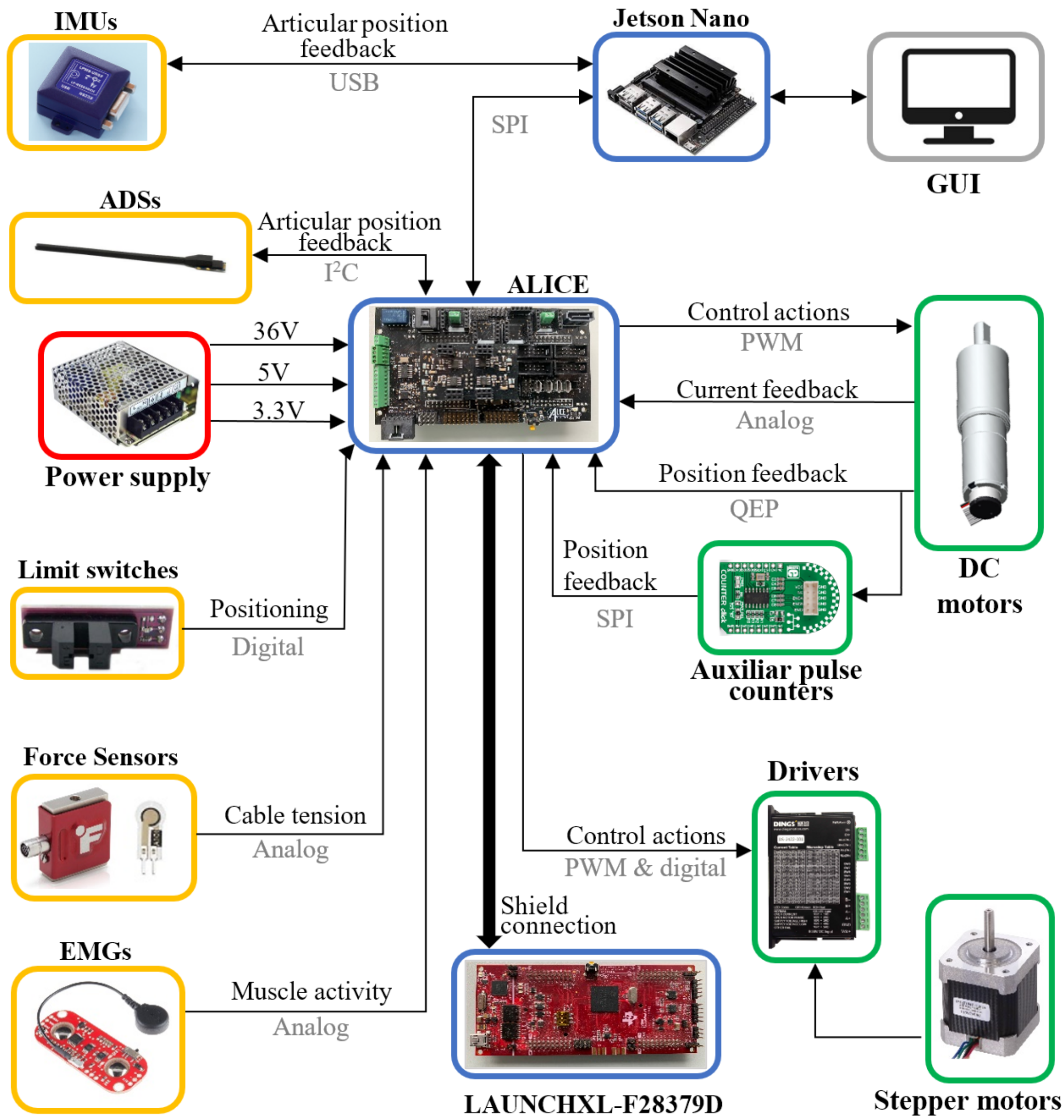
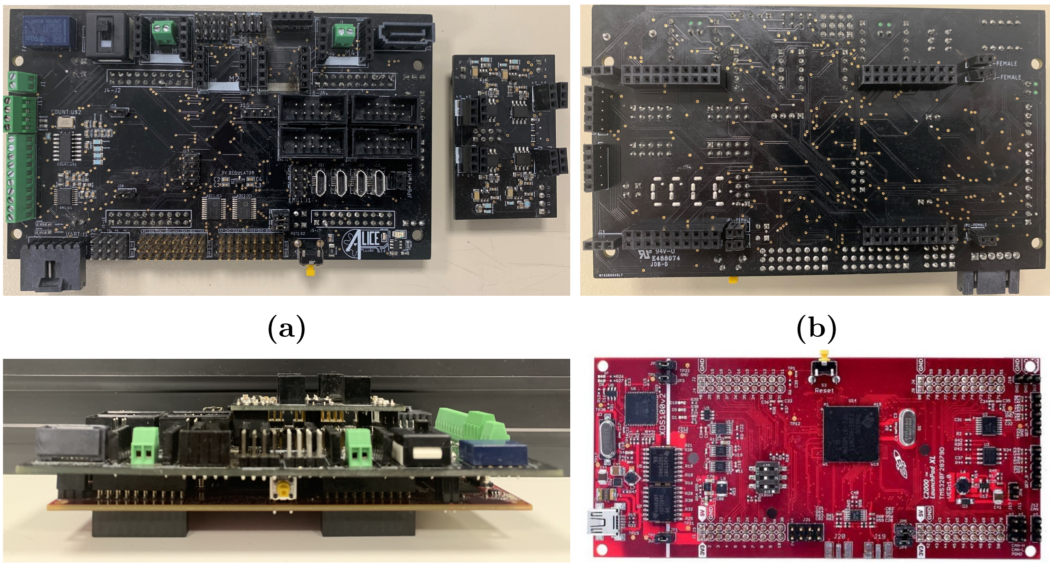


Figure 3.7: Electronics system diagram. Sensors are framed in yellow, power system in red, electronic boards in blue, actuation systems in green and the GUI in gray. The number of each of the devices that the system can interface is not represented in the diagram, but it is presented in Table 3.3.

text file for posterior analysis. Both boards communicate through SPI protocol. IMU sensors are connected to the Jetson Nano via USB.

A shield PCB was designed to interface actuators and sensors of ExoFlex v1 (see Section 3.2.2). This board was called ALICE (Assistive LImb Control Electronics), and is shown in Fig. 3.8. The board mainly outputs in a user-friendly configuration the pins of the LAUNCHXL-F28379D. This provides a fast connection method for the external peripherals. ALICE was designed to control four DC motors with encoder position feedback. The microcontroller only contains three quadrature encoder pulse (QEP) counters so, to control the fourth motor, a quadrature pulse counter chip was added to ALICE. This chip outputs via SPI to the microcontroller the number of counted pulses. Four Pololu 2961 breakout boards, based on the

MAX14870 single brushed DC motor driver carrier, are stacked over ALICE. The fact that the motor drivers are removable allows to replace them if any of them are damaged, what would not be possible if they were soldered directly over ALICE. For current sensing of these four DC motors, an auxiliary PCB which is stacked over ALICE was designed. This supplementary board includes four Allegro ACS723 Hall effect sensors with the appropriate conditioning circuit for the application (see Appendix B).



(a) **(b)** **(c)** **(d)**

Figure 3.8: ALICE board. (a) Top view of the main board (left) along with the current sensing board (right). The current sensing board is connected to the pins in the empty central area of the main board. (b) Bottom view of ALICE. Through the female connectors located around the board, ALICE is connected to the LAUNCHXL-F28379D. (c) ALICE and microcontroller board connected. (d) Texas Instruments LAUNCHXL-F28379D board.

The LAUNCHXL-F28379D has 15 analog input pins. These ports can be used to read analog sensors, such as current sensors, force sensors or EMG sensors. This number of ADC inputs can be insufficient depending on the experimental setup, so two 8-channel 12-Bit I^2C ADCs were incorporated to ALICE to extend the number of inputs for analog sensors to 31. This was done as an expectation of system scalability, what would require adding more degrees of freedom and, therefore, increase the number of sensors.

The development board offers plenty of communication modules. However, not all of them can be used at once since pins are shared among several peripherals of the microcontroller. To increase the flexibility of the system, once all the pins were assigned to their corresponding function, the unused pins with communication capabilities were presented in ALICE as generic communication connectors. In this way, the generic communication ports of ALICE are four UARTs and two SPIs. Note that via software, the pins can be reconfigured to be easily used as GPIOs if necessary. After the assignation of all the communication pins, remaining unused GPIOs were exposed as a GPIO block in ALICE.

The increase of the number of actuators in ExoFlex v2 (Section 3.2.3) required to include more circuitry to the exoskeleton electronics system apart from ALICE. An external board with a driver and a current sensor was attached to ALICE to drive the fifth DC motor. The used driver was the same as for the other four DC motors, and for the current sensing, a Sparkfun SEN-14544 breakout board based on the ACS-723 Hall effect sensor was used. Additionally, two external quadrature encoder pulse counters were added to the system in an external board. In this case, Mikroe's Counterclick breakout boards were used. Encoder readings are transferred via SPI to the microcontroller. To control the two stepper motors, four GPIOs are used (one configured as PWM and another as a digital output for each motor driver).

The hardware resources of ALICE, together with the aforementioned supplementary boards, are summarized in Table 3.2. With those resources, the system can simultaneously control five DC motors and two stepper motors while reading four IMUs, four ADSs and up to 31 analog sensors, including force sensors, current sensors and EMG sensors (Table 3.3).

Table 3.2: ALICE hardware resources.

Module	**Use**	**Channels**
PWM	Motors	7
QEP	Encoders	3
I^2C	ADSs	1 (4*)
	ADCs	1 (2*)
SPI	Encoders	1 (3*)
	Jetson Nano	1
UART	Generic	2
Analog Inputs	Current sensors	5
	Force sensors	4
	EMGs	6
	Generic	16

* The number in parenthesis is the number of elements controllable with one channel.

Table 3.3: Actuators and sensors interfaceable with the electronics system.

Device	**Number**
DC motors	5
Stepper motors	2
Encoders	6
IMUs	4
ADSs	4
Analog sensors	31

3.5 Software module

Specific software was designed for both the microcontroller and the Jetson Nano. The microcontroller software was designed in the context of this thesis, while the Jetson Nano one was developed by the research group to control the rehabilitation session (sending commands to the microcontroller according to the user's inputs) and to store in a text file all the received data from the microcontroller.

A real-time multi-thread software was developed to control the upper-limb rehabilitation exoskeleton and it was implemented over the TI microcontroller. The architecture is briefly described in this section.

As more actuators and sensors are added to the system, the computational load for the control unit increases. In this sense, the software must be modular and scalable, so that if the number of sensors or actuators increases, this can be reflected in the software in a fast and simple way. Additionally, software must run in real-time so that the execution of the application is deterministic, accomplishing the predefined deadlines for control execution and sensor readings collecting. Moreover, it is crucial to have a system that is able to log the data collected in experimental tests for posterior analysis.

The software was implemented over the Texas Instruments Real-Time Operating System (TI-RTOS) for microcontrollers. This operating system provides a set

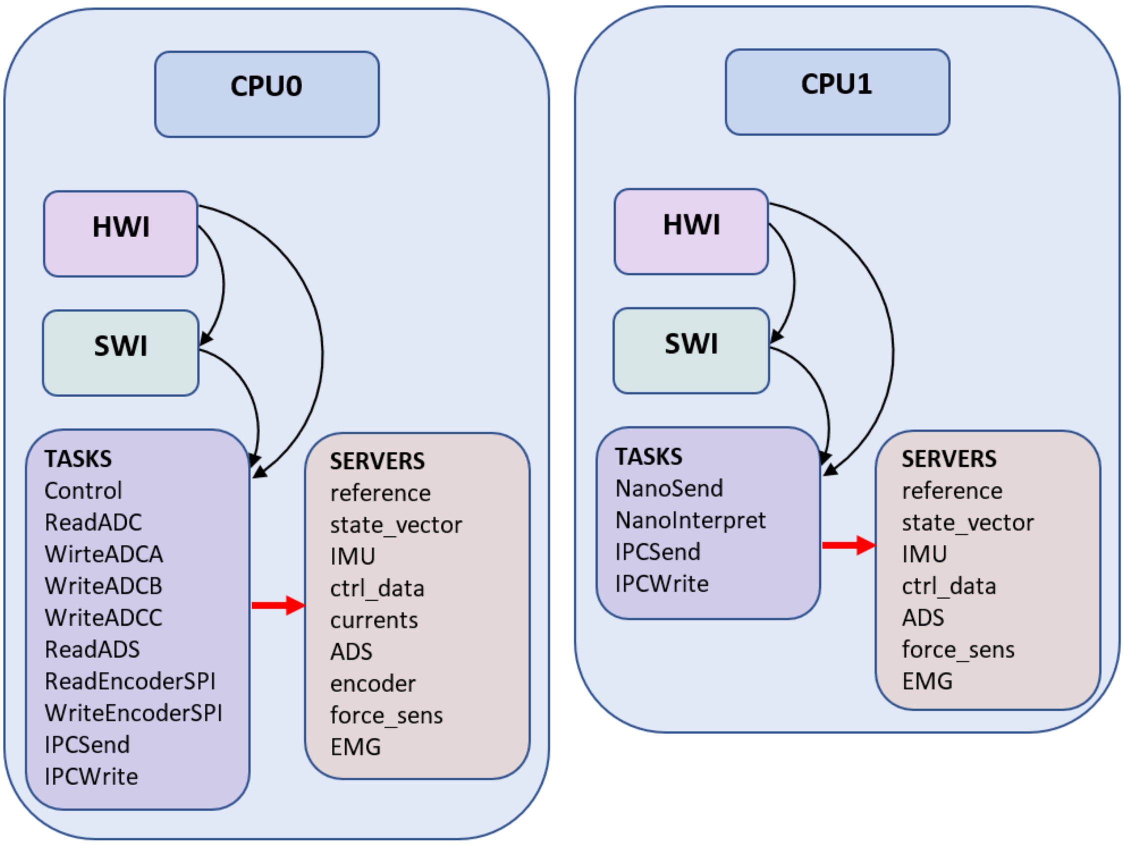


Figure 3.9: Overview of tasks and servers organization for both CPUs. Hardware interrupts can trigger software interrupts or task execution. Software interrupts can trigger task execution. Tasks read and write data to servers to share information in an organized and protected manner.

of tools such as task, mutex, semaphore, clock and timer, among others, which allow writing real-time software. The code was written in C language using Code Composer Studio v11.2.0 (CCS), developed by Texas Instruments.

The code was structured in tasks (can also be called threads). The execution period of the tasks is controlled by a scheduler provided by TI-RTOS, which uses timers which periodically unlock semaphores for code execution of each task. Data are shared among tasks via servers, which are mutex-protected variables. Every execution period of a task, necessary data are read from servers, specific task code is executed, and at the end of the task, new processed data are written to the servers. Figure 3.9 presents a basic overview of how tasks are organized in both CPUs. Hardware interrupts (HWIs) trigger the execution of some software interrupts (SWIs) and tasks, while software interrupts also trigger the execution of some tasks. CPU0 is in charge of the exoskeleton control and the readings of analog sensors, encoders (the integrated and the external ones) and ADSs. CPU1 is focused on the SPI communication with the Jetson Nano. Data are shared between the two CPUs via a read/write protected block of shared memory. Table 3.4 and Table 3.5 present a high-level description of the processing performed in each task.

Hardware Interrupts were used whenever possible to reduce both the work load for both CPUs (avoiding the need to poll communication modules) and the latency

Table 3.4: Tasks executed in CPU0.

Name	**Function**
Control	Reads sensor data from servers, executes controller, writes control data to servers.
ReadADC	Triggers an ADC conversion of all ADC modules (A, B and C).
WriteADCA	When ADCA data are obtained, it writes them to the server.
WriteADCB	When ADCB data are obtained, it writes them to the server.
WriteADCC	When ADCC data are obtained, it writes them to the server.
ReadADS	Reads the ADS sensors via SPI, filters data and writes them to the server.
ReadEncoderSPI	Request data from encoder pulse counters.
WriteEncoderSPI	When encoder data are received, they are written to the server.
IPCSend	Reads data from servers and sends them to CPU1.
IPCWrite	Receives data from CPU1 and writes them to servers.

Table 3.5: Tasks executed in CPU1.

Name	**Function**
NanoSend	Reads data from servers and sends them to the Jetson Nano.
NanoInterpret	Reads the data received from the Jetson Nano and saves them to the servers.
IPCSend	Reads data from servers and sends them to CPU0.
IPCWrite	Receives data from CPU0 and writes them to servers.

in execution of critical code sections. All the communication modules (SPI, I^2C and UART) make use of the internal FIFO registers so that software intervention in the communication processes is minimized. Hardware interrupts are generated when the FIFO registers for both sending and receiving data are empty or full, respectively, and software is executed to attend those situations.

The deadlines of all the tasks were set equal to their execution periods. A rate monotonic scheduling was used. This type of scheduling assigns higher priorities to more frequent tasks (the ones with shorter periods) in a static way (priorities do not change during code execution), and it is optimal [137]. It is understood by optimal that, if with this scheduling it is not possible to obtain an admissible plan that allows the execution of each task within its deadline, it is not possible to obtain a valid scheduling with any other method based on static priorities.