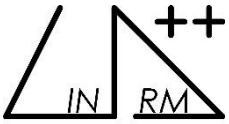


LINarm++
Affordable and Advanced LINear device for ARM rehabilitation

Deliverable D1.3
LINarm++ prototype

Contractual delivery date	31.10.2016 (month 18)
Actual delivery date	16.11.2016 (month 18 + 16d)
Version	1.0
Dissemination level	PU
Authors	Matteo Malosio (CNR-ITIA) Alessio Prini (CNR-ITIA) Giulio Spagnuolo (CNR-ITIA) Tito Dinon (CNR-ITIA) Matjaz Mihelj (UL) Janez Podobnik (UL) Andrea Crema (EPFL)

Table of Contents

1 Executive summary

This deliverable deals with the final prototype of the LINarm++ experiment. The overall system and its subcomponents are depicted with significant pictures and graphic representations. Detailed descriptions of the developed subcomponents are available in Deliverable D2.2, D3.2, D4.2, D5.2, D5.3. They are here presented in their final integrated versions, reached, in some cases, through a series of steps to solve technical issues arose during the project. The document is structured as follows. After an introduction to the platform and its foreseen functionalities in Chapter 2, its main subcomponents are depicted in Chapters from 3 to 8. The complete demonstrator, arranged in different configurations and with different sets of submodules, is described in Chapter 9.

2 Introduction

LINarm++ is a multisensory and multimodal device for neuromuscular rehabilitation of the upper limb, designed to enable enriched rehabilitation treatments [1]. Originating from an existing low-cost variable-stiffness rehabilitation device, it expands its functionalities by integrating additional modules in order to augment application scenarios and applicable clinical techniques. The newly developed system focuses on the integration of the following set of optional modules interacting with the patient (Figure 2.1):

- A redesigned version of a mechatronic linear variable-stiffness device for rehabilitation;
- A low-cost unobtrusive sensory system for measuring the patient's physical activity and his physiological state, in order to obtain and constantly update a comprehensive state of the patient.
- An easily wearable FES system, allowing selective and effective stimulations of upper-limb muscles.
- Engaging on-line adaptable rehabilitation scenarios and virtual environments, which adapt during the training to the level of difficulty that is most appropriate for each individual subject, to ensure the best level of subject's activity in terms of motor and cognitive engagement.

All the platform is managed by a central control system in charge of synchronizing and updating rehabilitation parameters in accordance with a patient model. The patient model is in charge of determining training task parameters in relation to the user's performance and the physical/physiological state, in order to influence the user's engagement and performance, with the aim of fulfilling the actual needs of the patient during the therapy.

The result is a modular, integrated and affordable rehabilitation device, enabling a biomechanical, neurological and physiological-based training of patients, including innovative features currently unavailable within off-the-shelf rehabilitation devices. Its modularity enables it to be configured according to the actual needs and budget capabilities of the actual usage environments.

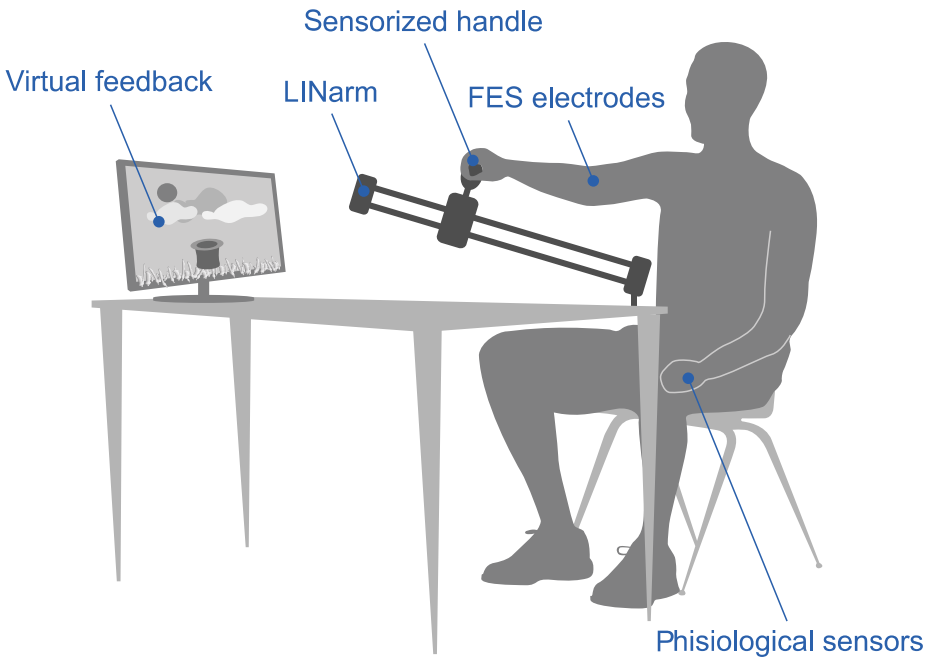


Figure 2.1 – Representation of the LINarm++ rehabilitation platform.

3 Mechatronic device

Note: *The mechatronic device is described in details in deliverables D3.1 and D3.2.*

The linear mechatronic device (Figure 3.1) embeds a newly developed Variable Stiffness Actuator (Figure 3.2) specifically designed to be embedded in devices performing linear motions [2]. The cams to realize antagonist non-linear springs have been designed within proper disc-shaped mechanical constraints to guarantee a correct wire wrapping/unwrapping (Figure 3.3) and assembled coaxially to limit the overall dimensions of the carriage (Figure 3.4). Motors and sensors are embedded within the so-called "motorized unit" of the device (Figure 3.5). A proper covering of the device has been designed and realized to guarantee a safe interaction and protection of the user, besides a proper protection of mechanical and electrical components (Figure 3.6). The complete device have been assembled on a tripod through a spherical joint to allow a proper orientation of the device to assist differently-oriented rehabilitation tasks (Figure 3.7).

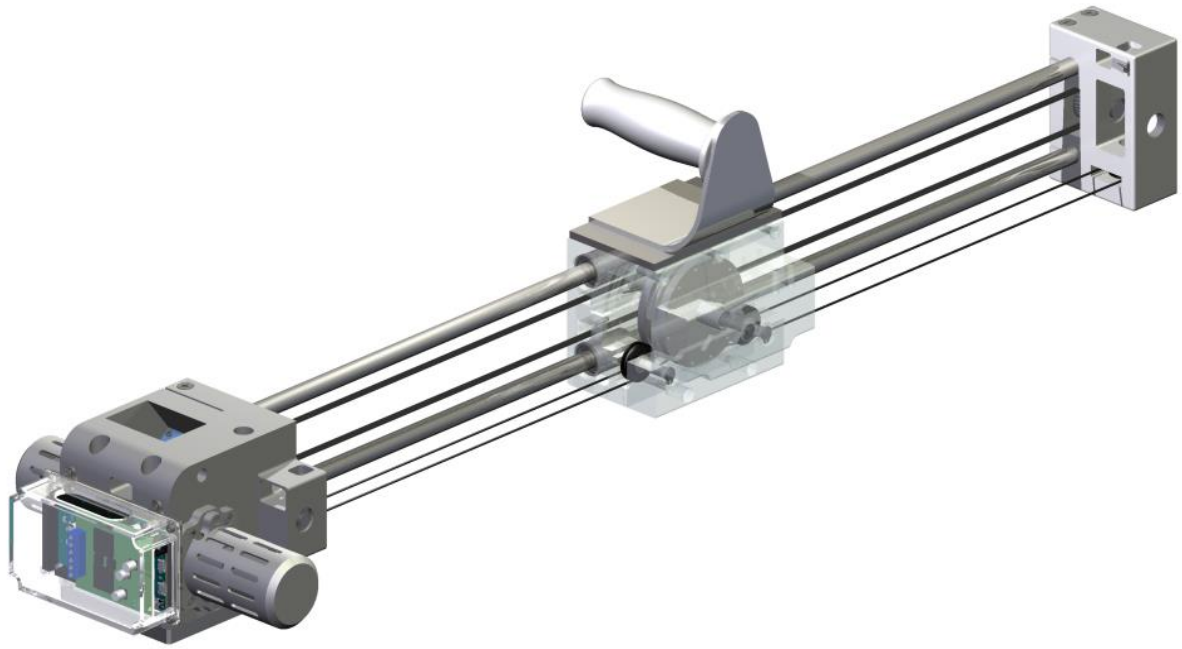


Figure 3.1 – LINarm2: a variable stiffness device for upper-limb rehabilitation embedding the LinWWC-VSA architecture with the coaxial configuration.

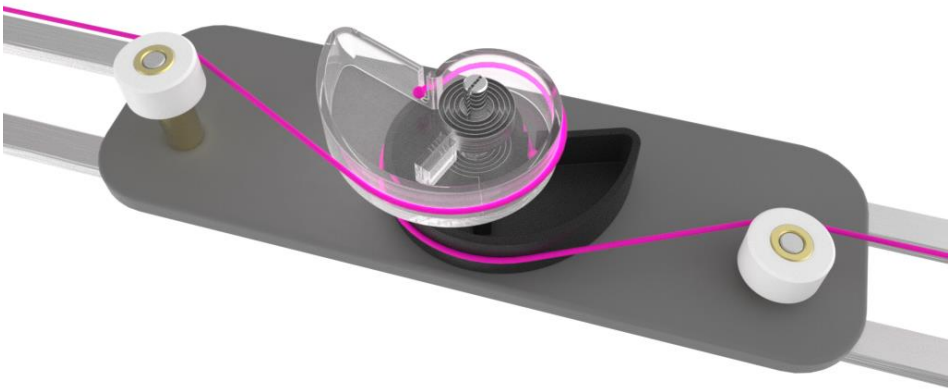


Figure 3.2 – LinWWC-VSA: the variable stiffness actuator embedded within the LINarm++ mechatronic device.

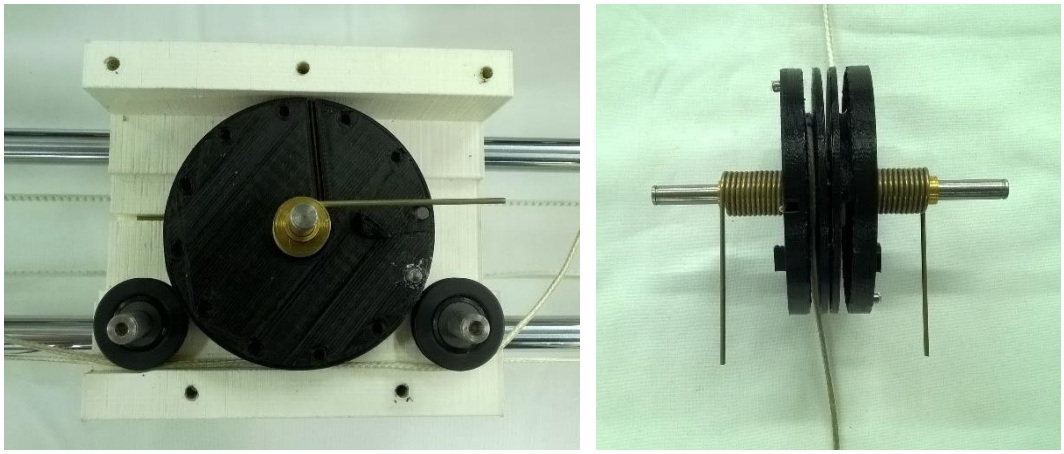


Figure 3.3 – Detailed view of the cam-based VSA mechanism embedded within the linear carriage of the mechatronic device.

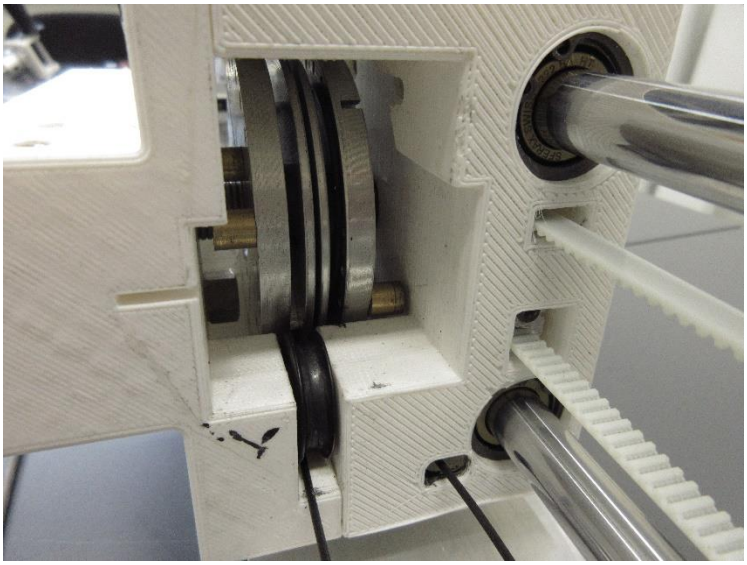


Figure 3.4 – The cam-based VSA mechanism assembled within the carriage.

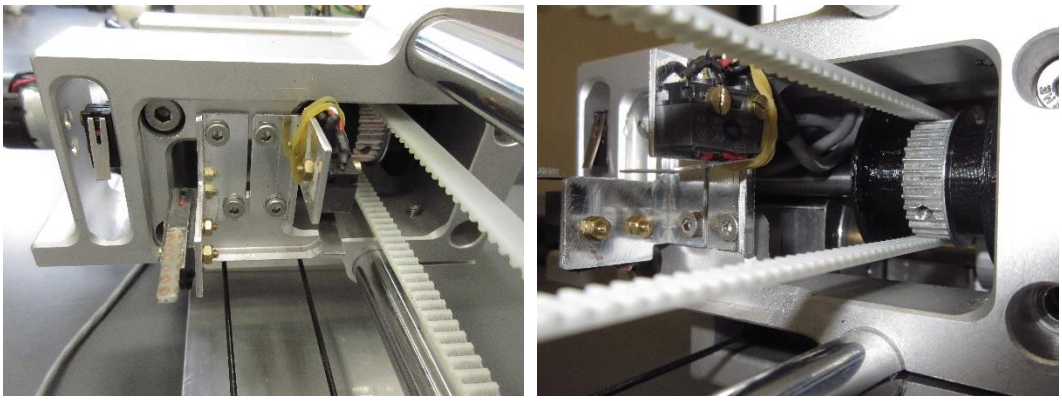


Figure 3.5 – Detailed view of the motorized unit of the mechatronic device. It embeds the motors, a wire-based trasmission system and the required electronics as proper end-strokes to reset the device.

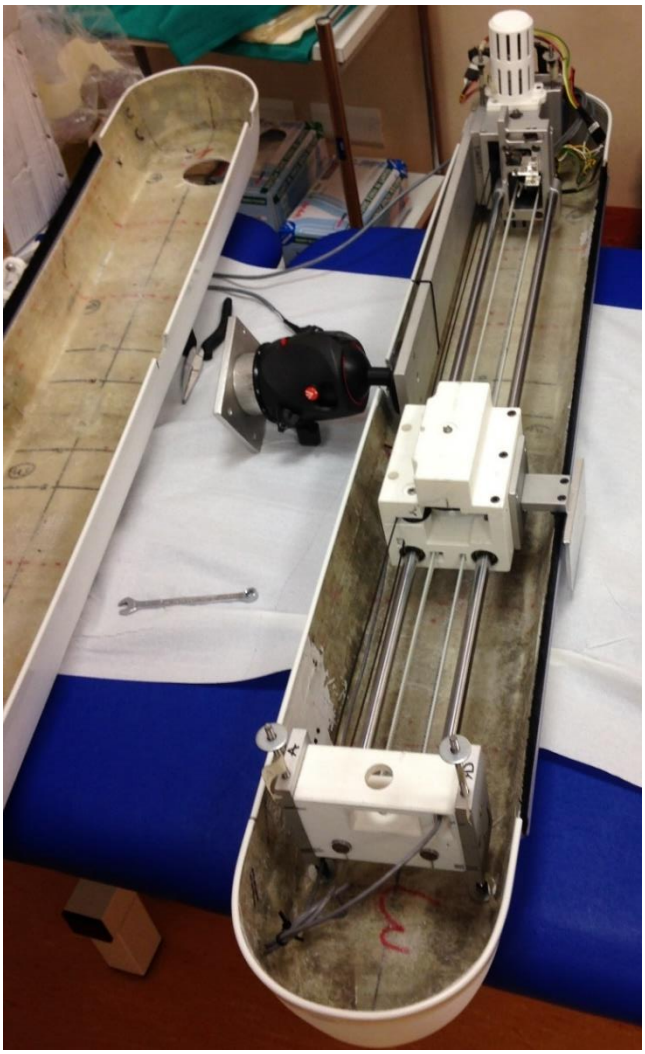


Figure 3.6 – LINarm2 mechatronic enclosed inside its coverage.

Figure 3.7 – LINarm mechatronic device with the covering.

4 Embedded-and-wearable sensory system

Note: *Embedded-and-wearable sensory system is described in details in deliverables D4.1 and D4.2.*

Embedded-and-wearable sensory system were implanted in two types of handles as shown in Figures 4.1 and 4.2. The handle-set includes a base with electronics on which both types of handles can be mounted (see Fig. 4.3), a spherical handle (see Fig. 4.1) and a cylindrical handle (see Fig. 4.2). The base is attached to the end-point of the robot. One base is used so that one PCB with electronics can be used for two shapes of handles. Each handle includes sensors for measuring physiological parameters (heart rate sensors, skin conductance sensors and temperature sensors), while cylindrical handle also includes the force cell for measuring the grasp force.

Figure 4.1 Spherical handle with physiological sensors.

Figure 4.2 Cylindrical handle with physiological sensors.

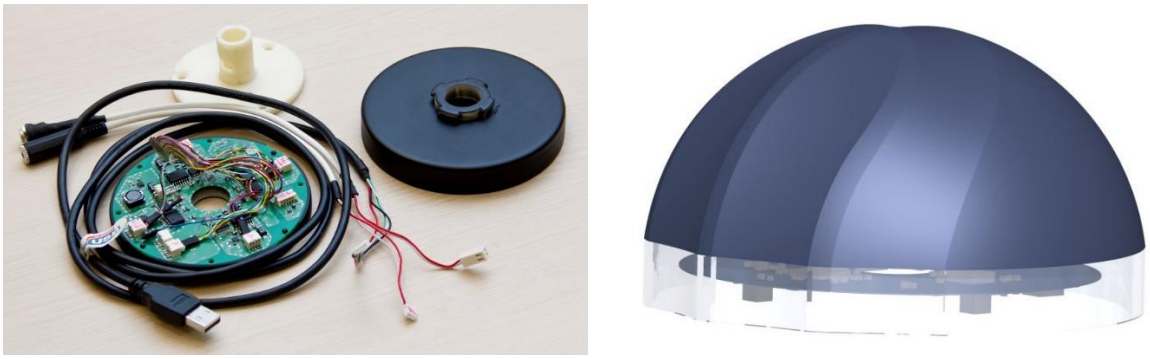


Figure 4.3 Base with electronics for handles and a CAD model of the integrated system.

Sensors in the two types of handles are distributed according to the measurement schema presented in Fig. 4.4.

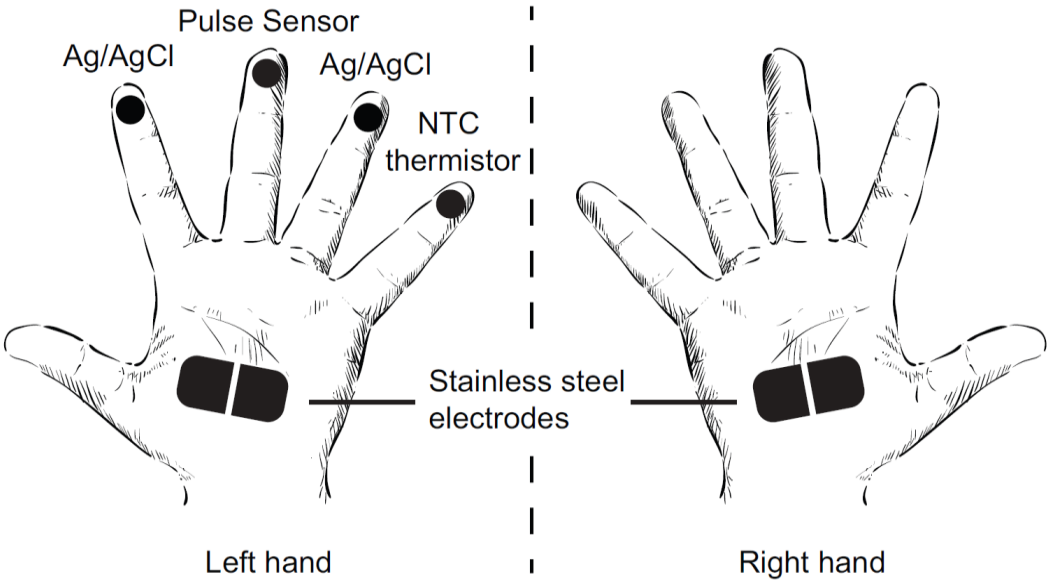


Figure 4.4 Predicted measurement locations for both left and right palms at time of interaction with the robot. Left (active) hand is interacting with the robot and is simultaneously being measured for physiological response at the handle. Right (passive) hand is resting on the static handle, populated only by the ECG electrodes.

The electronics is designed in accordance with medical standards. All electrical components are properly isolated for guaranteeing safety of the patient. Block diagrams of the measurement electronics are shown in Fig. 4.5 for the first prototype and Fig. 4.6 for the final implementation (used in the handle base shown in Fig. 4.3).

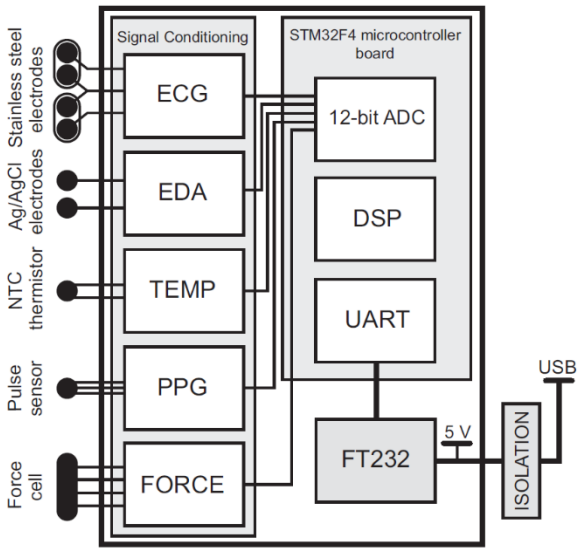


Figure 4.5 Scheme of the sensory system electronics. Separate analog frontends are locally regulated to ensure stable operating voltage. Isolation for this prototype is provided externally, and power is supplied by the USB port.

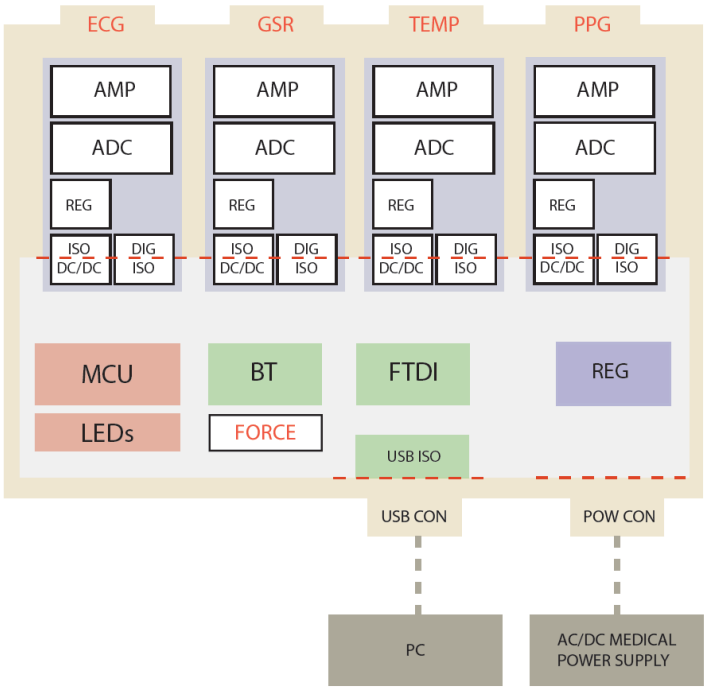


Figure 4.6 Block diagram of the final version of low-cost measurement system for measurement of physiological signals

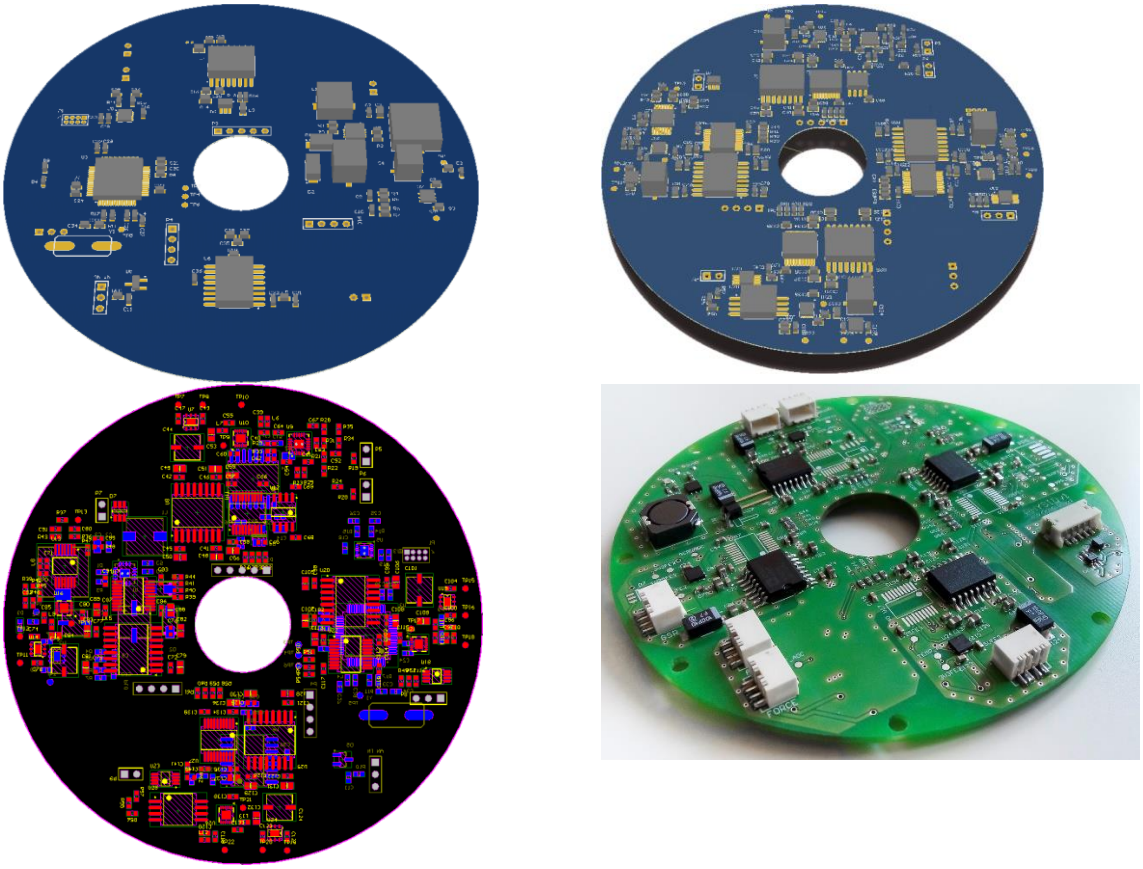


Figure 4.7 Printed circuit board of the final electronics.

The base of the handle is firmly attached to the robot end-effector. The two handles are designed in such a way to be easily interchangeable based on the requirements for specific patient. Both options are shown in Fig. 4.8.

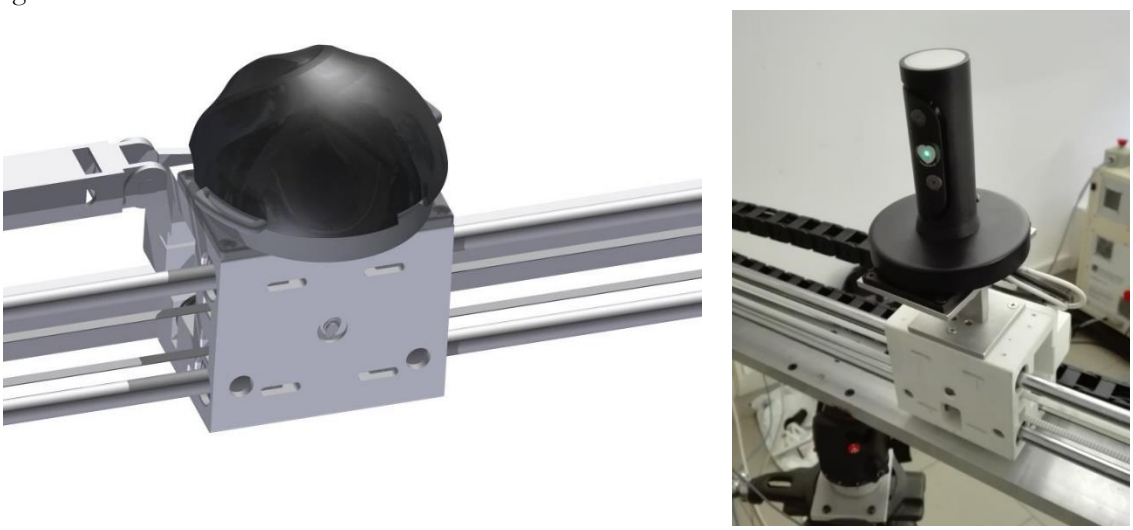


Figure 4.8 Render of the spherical handle on the robot mechanism and the cylindrical handle on the robot.

Sensory systems embedded into the two handles were validated in operational conditions while mounted on two different robots, namely the HapticMaster robot and the LINarm++ device. Figures 4.9 and 4.10

show examples of the acquired and processed data. As noted at the beginning of this chapter, embedded-and-wearable sensory system is described in full details in deliverables D4.1 and D4.2.

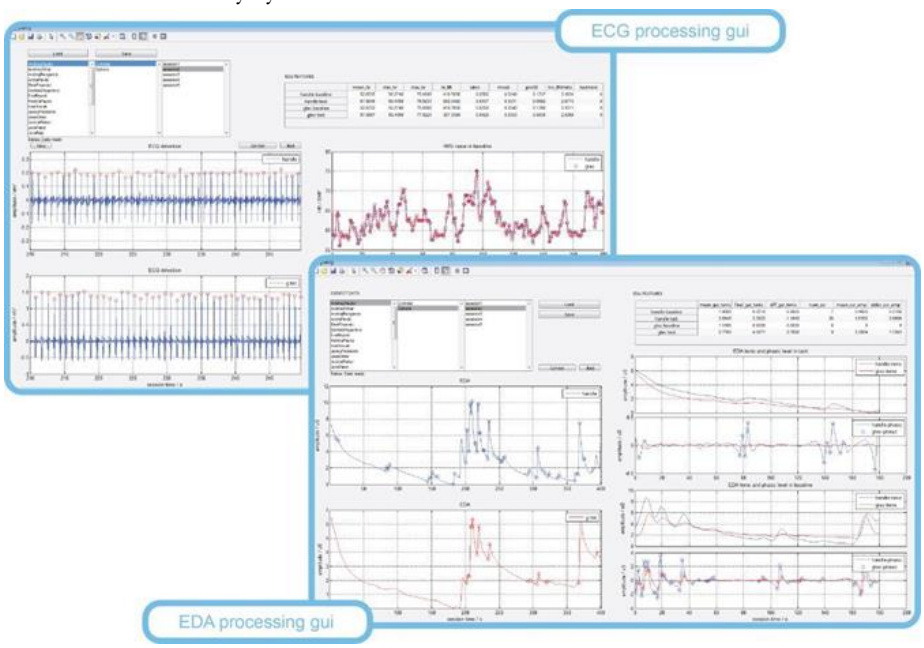


Figure 4.9 Graphical user interface with acquired signals.

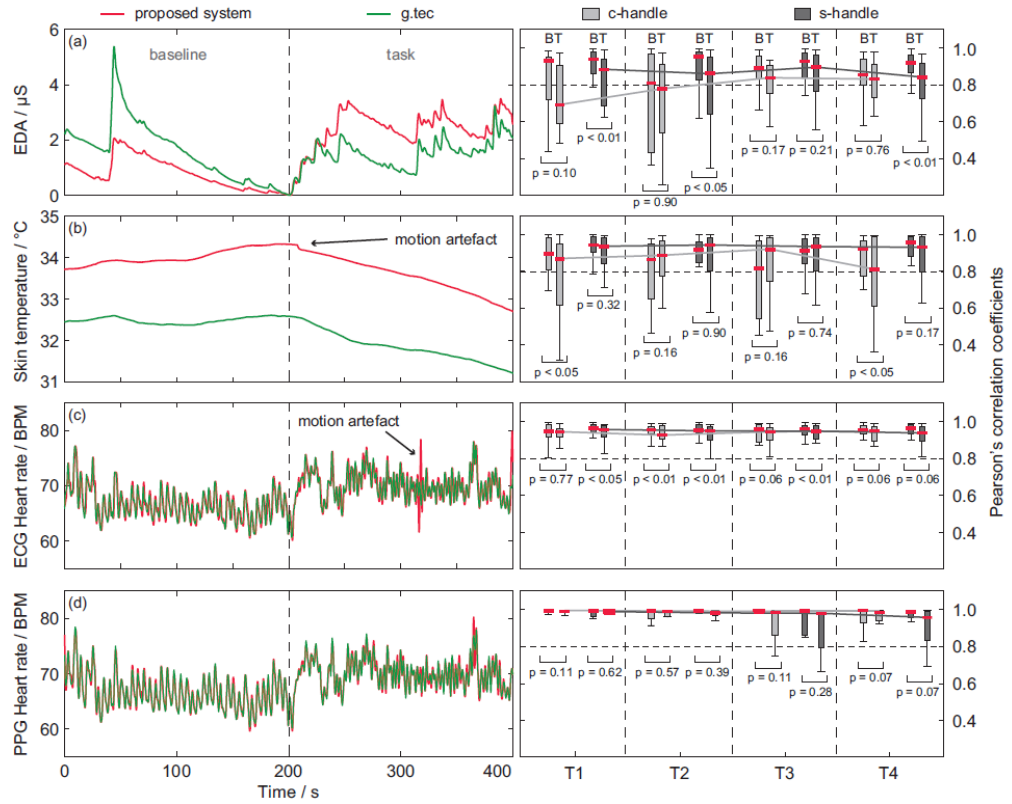


Figure 4.10 Raw physiological waveforms (left) and Pearson's correlation coefficients for all tasks are represented as box plots (right): Electrodermal activity (a), Peripheral skin temperature (b), ECG HR signal (c), and PPG HR signal (d). Gray lines are connecting the median values of Pearson's correlation coefficients in tasks to better illustrate the correlation trend between different tasks. B and T are marking baseline and task period for each of the four tasks respectively.

5 Neuromuscular Electrical Stimulation system

Note: *The Neuromuscular Electrical Stimulation system is described in details in deliverables D5.1 and D5.2.*

An NMES system has been developed in this perspective to offer both multimodal and simple to use motor support and sensory feedback. It has been developed fitting with Linarm++ budget and Technological Readiness requirements.
For simplicity of wearing and parsimony of the overall number of channels, minimizing the requirements of the electrostimulator, only two muscle groups are elicited by NMES in each exercise and are empirically chosen as the proximally dominant and distally dominant muscle for each action.
Hand-to-Mouth assistance relies on the stimulation of the Biceps Brachii and of the Brachioradialis whereas Reaching assistance uses the Deltoid Anterior and the Triceps. On the basis of this rationale and to fulfill the ease-of-installation requirement, the NMES system comprises an electrical stimulator (Rehastim One, Hasomed GmbH, Magdeburg, Germany), standard transcutaneous electrodes, customizable sleeves which aims to simplify the positioning of the electrodes on the subject (Figure 5.1) and a switch box to select different sets of electrodes (Figure 5.2). The active component of each exercise is split in two independent tasks.
NMES has to elicit, for each task, a response able to support motion in the main expected direction. The optimal location of stimulation and intensity is obtained through a ranking process. Each targeted muscle uses a multi-electrode wearable containing four independent active electrodes and a common reference electrode.
The selection of the current intensity i and of the location of stimulation in the multi-electrode is performed during a calibration phase, compensating for sub-optimal positioning or avoiding to stimulate areas that could elicit adverse sensations. Since each muscle is deemed responsible for a specific task t, the calibration procedure aims at finding which is the best responsive electrode e, able to elicit the force Φ_t expected to be necessary for the task.
The scan proceeds sequentially for each task, for each electrode, with the current ramping in intensity up to $I = i_{max}$, or interrupted with the pain button.
During the identification procedure, the stimulation frequency is set to $F=30Hz$ and the pulsewidth is set to half of the dynamic range of the stimulator. The stimulation parameters for each task t are identified as the combination of location e_t and minimum current $I_{t,e}$ required to elicit the target force. If more than one electrode per matrix is suitable to induce motion without discomfort, the one with minimal current is chosen.
Referring to Figure 5.3, the stimulation is interrupted when the electrical current induces pain to the patient, known by the Overcurrent Button, or when the limb rotates by a predefined target angle, measured through an IMU. The best electrode is the one able to obtain the angle rotation target of the limb with the minimum current.
The stimulation able to induce motor contraction is used in a non-patterned fashion and the stimulation profile is continuous with the movement. Once location and current max intensity are defined, the stimulation intensity is obtained by means of pulsewidth and/or current modulation. The NMES assistance can be modulated in accordance with the percentage of the LINarm movement cycle and according to preset activation profiles, chosen among a set of approximated biomimetic responses.

The FES system is integrated within the ROS framework and the LINarm++ Manager through a proper library. All the parameters of the electrostimulator can be controlled within the LINarm++ manager (Figure 5.4) and stimulation profiles can be properly defined (Figure 5.5).

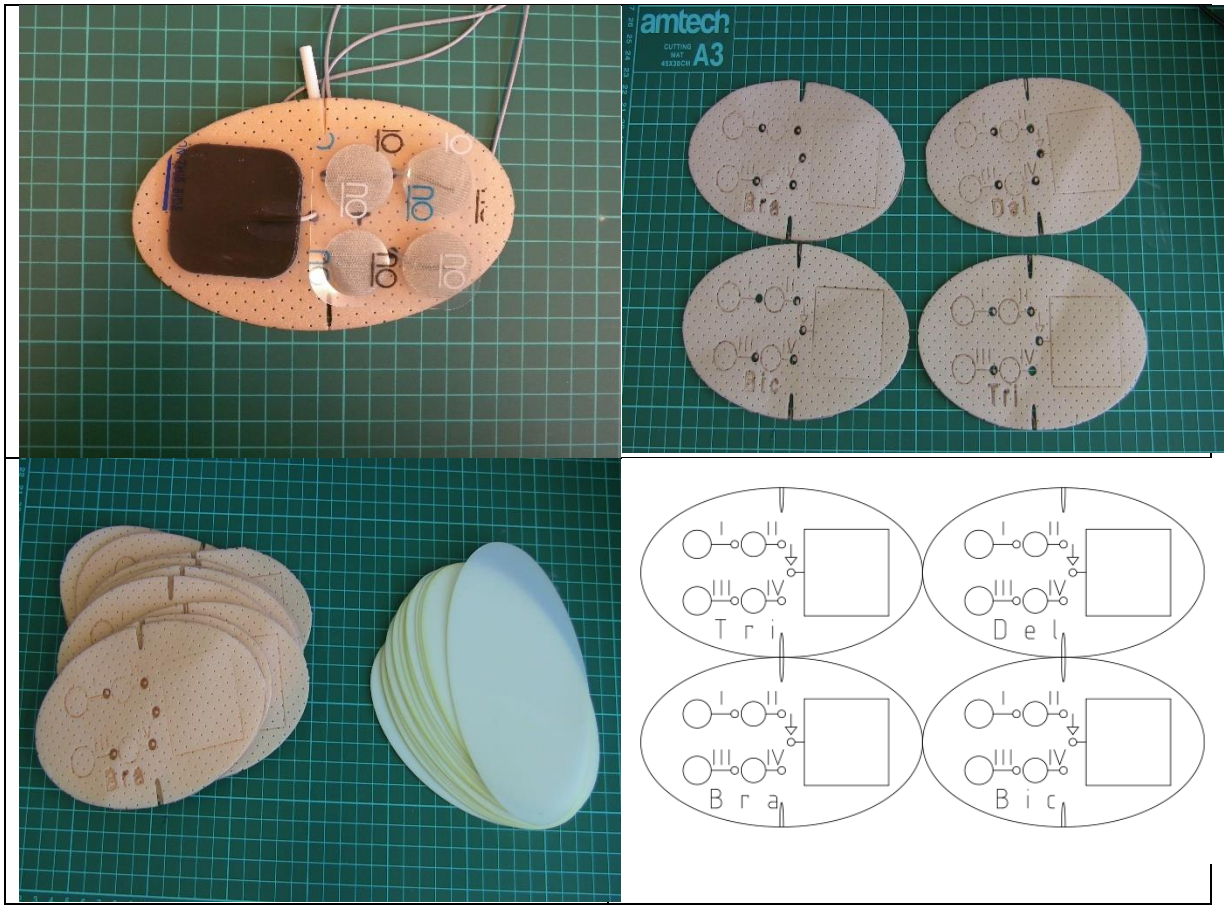


Figure 5.1 – Examples of custom pads for stimulation. Top left, the skin side of the patch. Bottom left, precut patches and covers for electrode protection. Right: example of custom cut and engraving for labelling purposes.

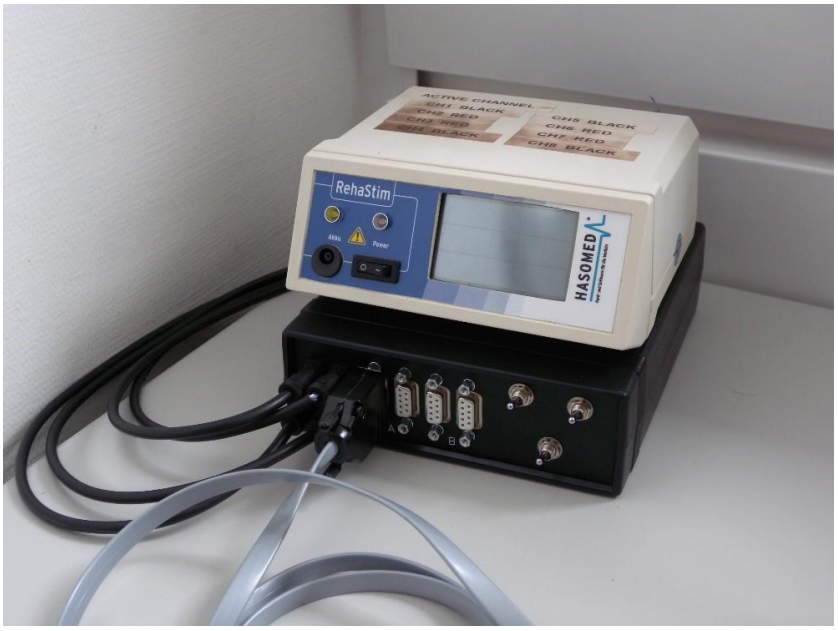


Figure 5.2 – The HASOMED electrostimulator and the switch box realized to allow easily reconfigure the stimulation system to control different stimulation patches.

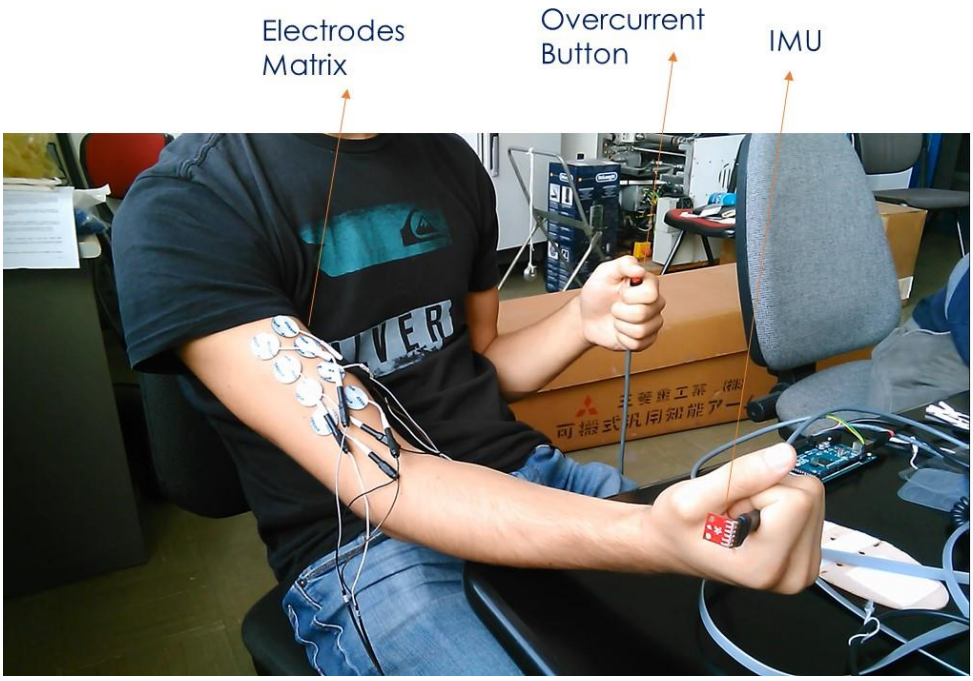


Figure 5.3 – FES autocalibration setup. An auto-calibration procedure for the identification of the best electrode to be used within each matrix of electrodes have been implemented. A step-shaped current signal stimulation is generated by the stimulator for each electrode. The stimulation is interrupted when the electrical current induces pain to the patient, known by the Overcurrent Button, or when the limb rotates by a predefined target angle, measured through an IMU. The best electrode is the one able to obtain the angle rotation target of the limb with the minimum current.

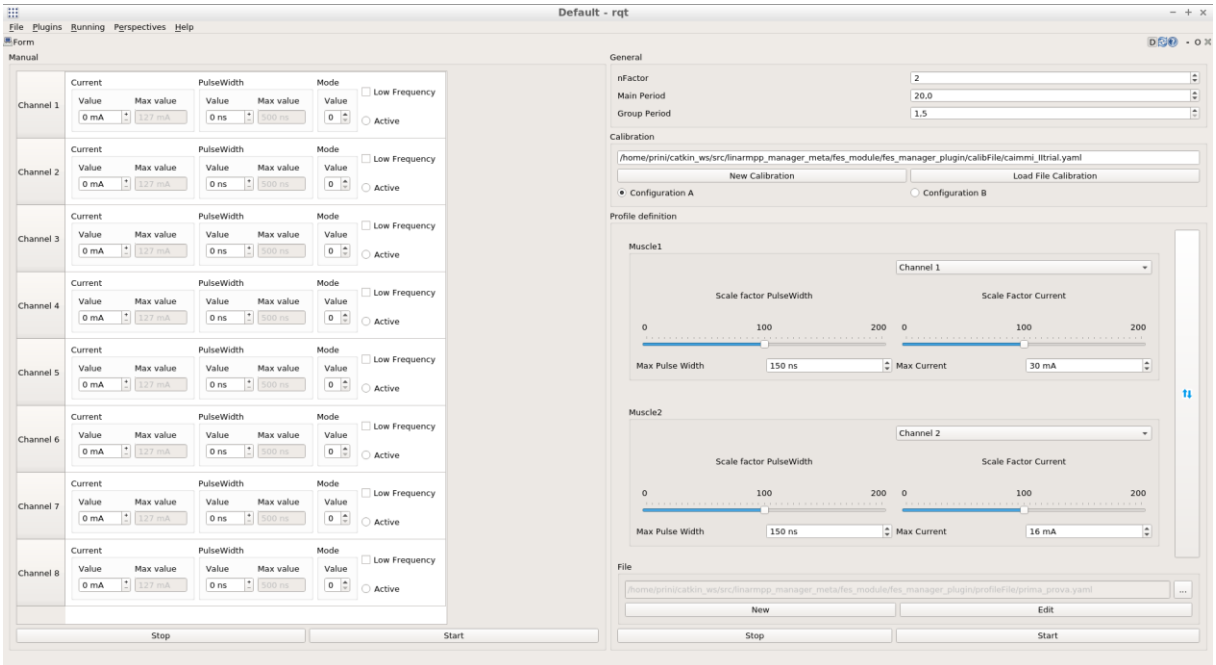


Figure 5.4 – Graphic user interface to control the FES module within the ROS framework. All the stimulation parameters can be controlled independently changing independently both current and pulsewidth values(left). The stimulation profiles for the two muscles can be loaded from the proper file and can be manually scaled by proper widgets during the exercise(Profile definition panel on the right).

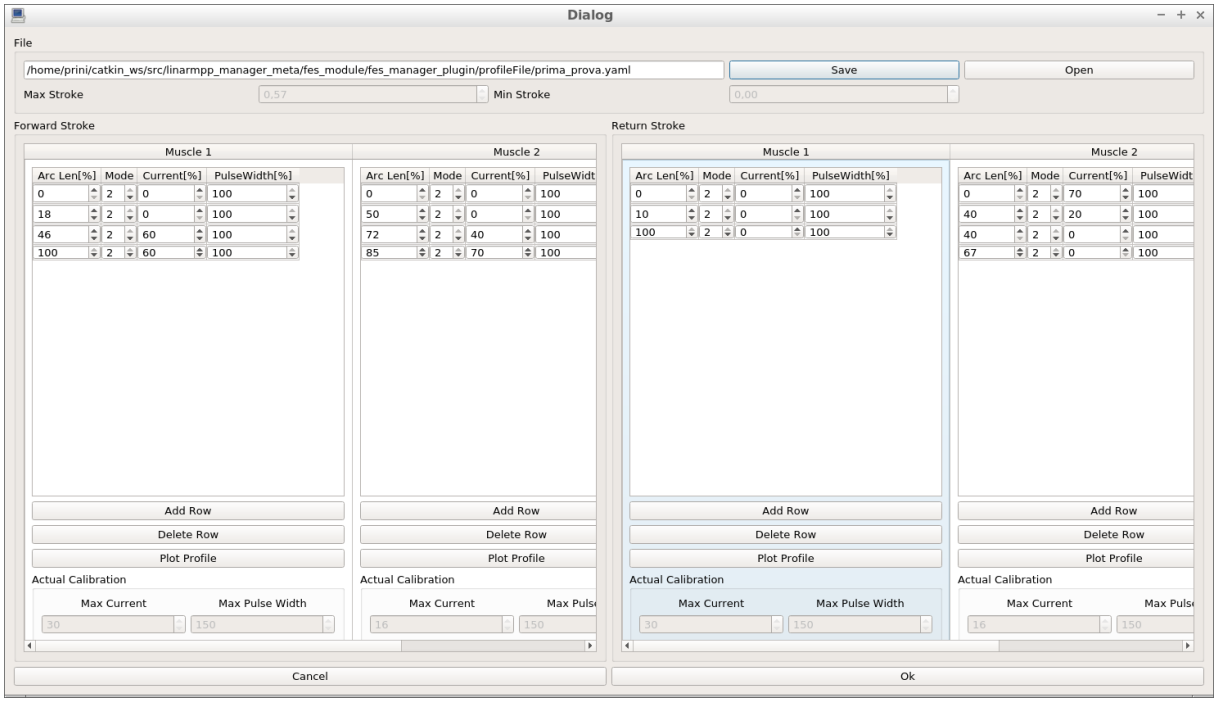


Figure 5.5 – FES profiles definition interface. The stimulation profile of each muscle in charge of determining a desired arm movement can be defined specifying, for each percentage of the arm stroke, the current and pulsewidth values. Values are interpolated linearly along the trajectory.

6 Virtual environment

Note: *Virtual environment is described in more details in deliverable D5.3.*

A set of various games was developed using Unity development kit. Games differ in level of engagement required from the subject. For motor task the difficulty is changed in two ways:

- by increasing or decreasing the speed of the objects which need to be caught or followed,
- by increasing or decreasing the damping of the robot.

For cognitive task the difficulty is changed by presenting harder or easier cognitive task.

Aim of task in virtual environment is to ensure that the patient's attention is properly gained and maintained throughout the provided rehabilitation task. Also important is the possibility of adaptive stimulation of subject's activity in terms of motor and cognitive engagement.

Figures 6.1 to 6.7 present different training scenarios that were implemented for the LINarm++ project specifically.

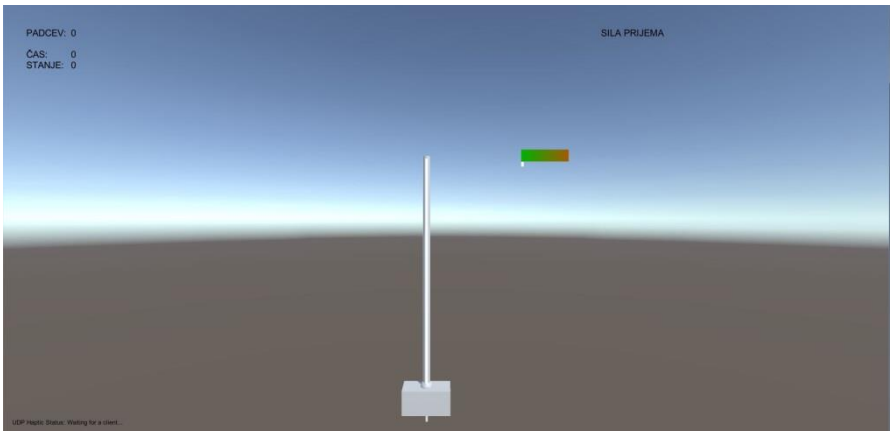


Figure 6.1 Pendulum test game. Pendulum game was developed to test the sensory system for measuring the physiological responses. The task is to balance the virtual inverted pendulum by moving the robot end-effector.

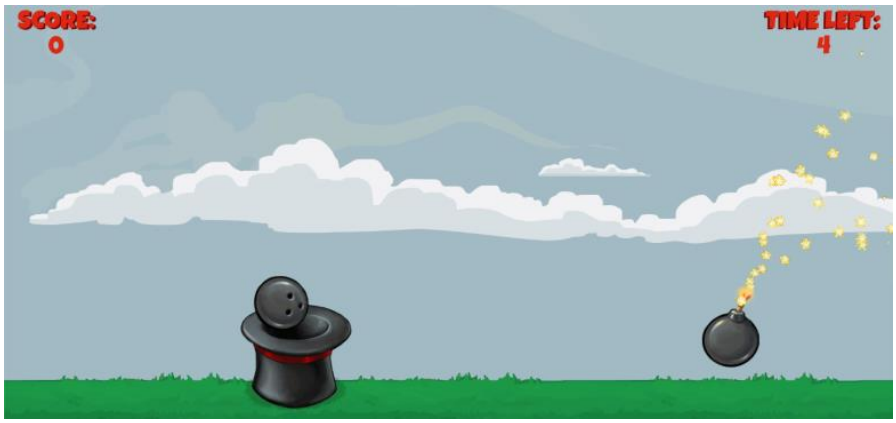


Figure 6.2 Catch-and-avoid game – falling balls and bombs. This represents a basic catch-and-avoid task, focusing on subject's motor engagement.

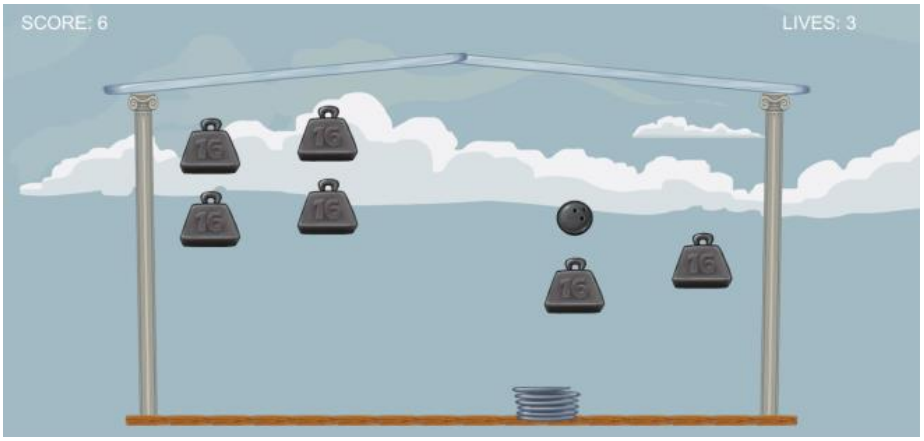


Figure 6.3 Catch-and-avoid game - rebounding of bowling balls. This game is similar to game shown in Fig. 6.2, it again requires mainly motor engagement, but as an upgrade to the previous game, provides also occasional cognitive engagement.

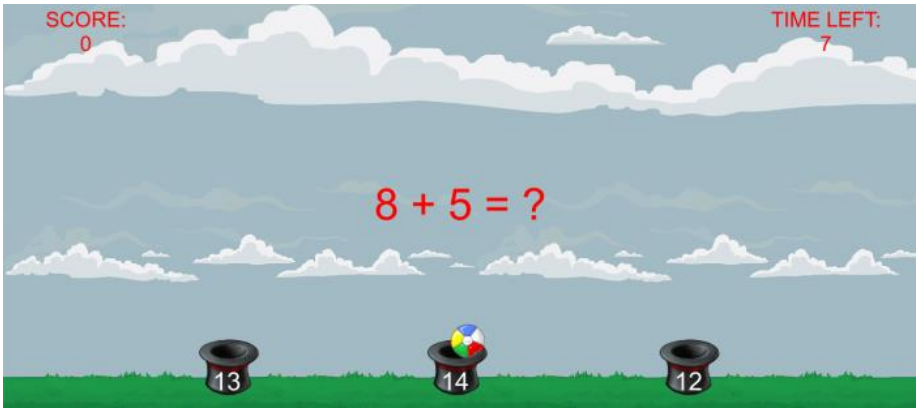


Figure 6.4 Drop the ball in correct hat. This game was developed to ensure shared motor and cognitive effort engagement.

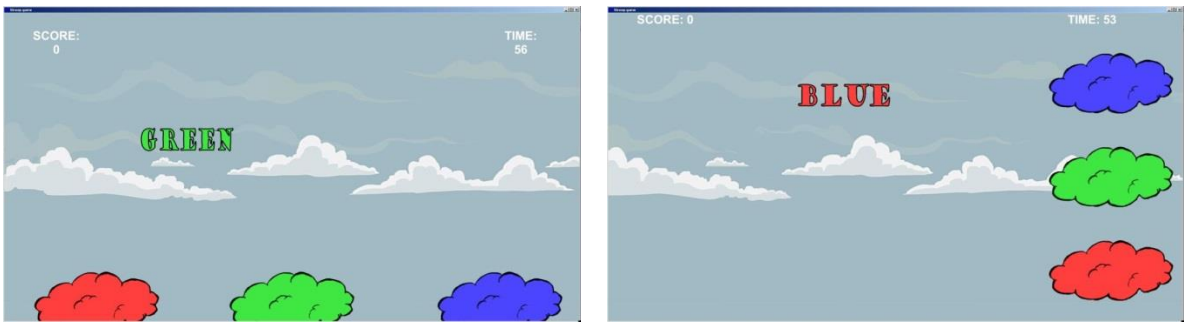
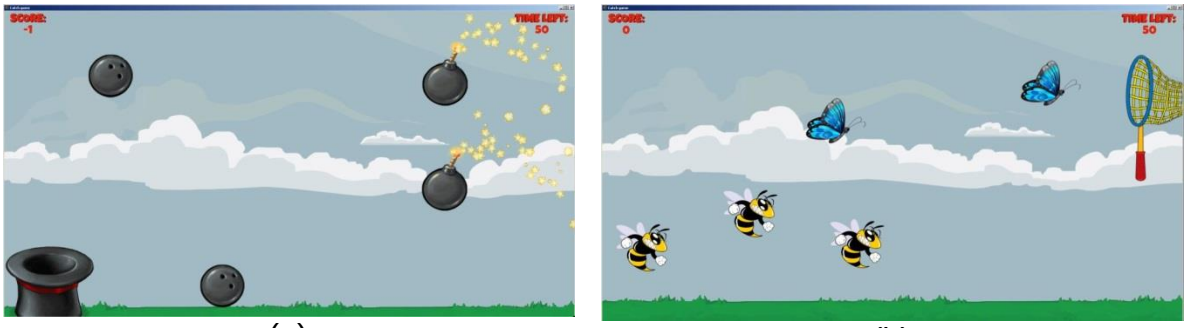
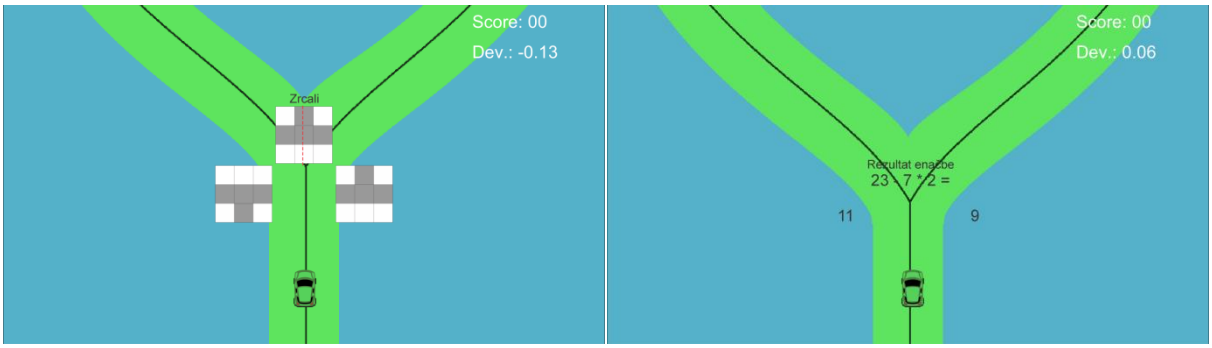


Figure 6.5 Figure shows Stroop physiological task game for two orientation of the robot: (a) horizontal and (b) vertical orientation. The game combines a motor task and a cognitive task, for motor task the game requires the movements left-right or up-down and for a cognitive task a custom version of well-known Stroop test was implemented. The game was primarily used for validation of the "patient model" on healthy subjects.



(a) (b)

Figure 6.6 Figure shows the Catch-and-avoid game for two orientation of the robot: (a) horizontal and (b) vertical orientation.



(a) (b)

Figure 6.7 Driving a car. The task is to drive the car along the road with junctions. The task combines both motor and cognitive task. The motor task is to move the car left-right, while the car is moving forward along the road (for practical reasons the car is positioned in the bottom of the scene and the scene with the road is moving from upper edge of the scene downwards to bottom edge of the scene). The cognitive task is to correctly answer the question presented before each junction.

Subfigures (a) and (b) show two tasks, which need to be solved during the training b): mirroring the image (above), solving a mathematical task (below). The task was validated with approximately 40 healthy subjects in order to test physiological responses to different training conditions.

7 Patient model

Note: *Patient model is described in more details in deliverables D2.1 and D4.2.*

Patient model provides the core functionality for adaptation of training to individual patient's needs. The task can be adapted in terms of motor and cognitive challenges. The model integrates clinical information about the patient with the automatic observations of patient's activities during the training with the LINarm++ device.

Patient model is constructed as a decision tree with five layers (see Figure 7.1). The inputs to the decision tree are the following parameters:

- patient's clinical assessment scores,
- patient's task performance (scores, ability to complete single subtask, time to complete single subtask),
- patient's motor performance (force, velocity, power, smoothness, …), and
- patient's physiological assessment (heart rate, skin conductance, peripheral skin temperature).

The output of the decision tree is the value α that defines the robot support. At the input to the top layer the robot support α can have any value from full assistance ($\alpha = 1$) to full resistance ($\alpha = -1$), thus $-1 \leq \alpha \leq 1$. Each layer limits α values to a subset of $-1 \leq \alpha \leq 1$. In the final layer the parameter α gets a single scalar value that is sent to the robot controller.

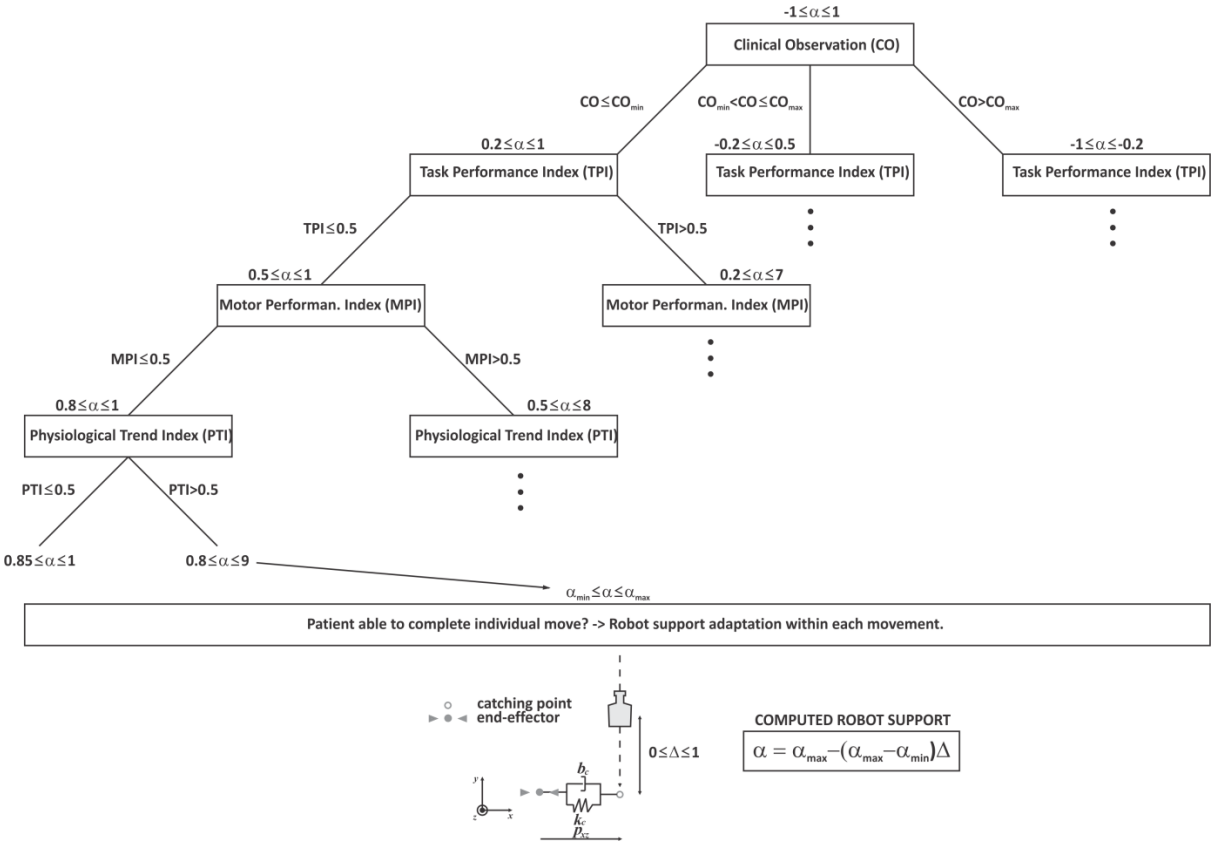


Fig 7.1 Decision tree for determining the level of robot support.

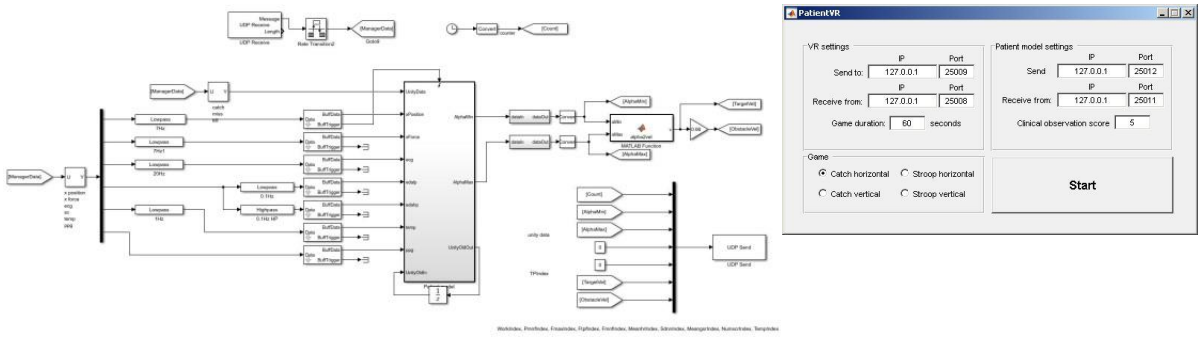


Figure 7.2 Matlab/Simulink implementation of the patient model later compiled into an executable and the graphical user interface for setting parameters for virtual scenario and starting games and the patient model.

Figures 7.3 to 7.5 show the results of the validation of "patient model" on healthy subjects. Detailed explanations are provided in figure captions.

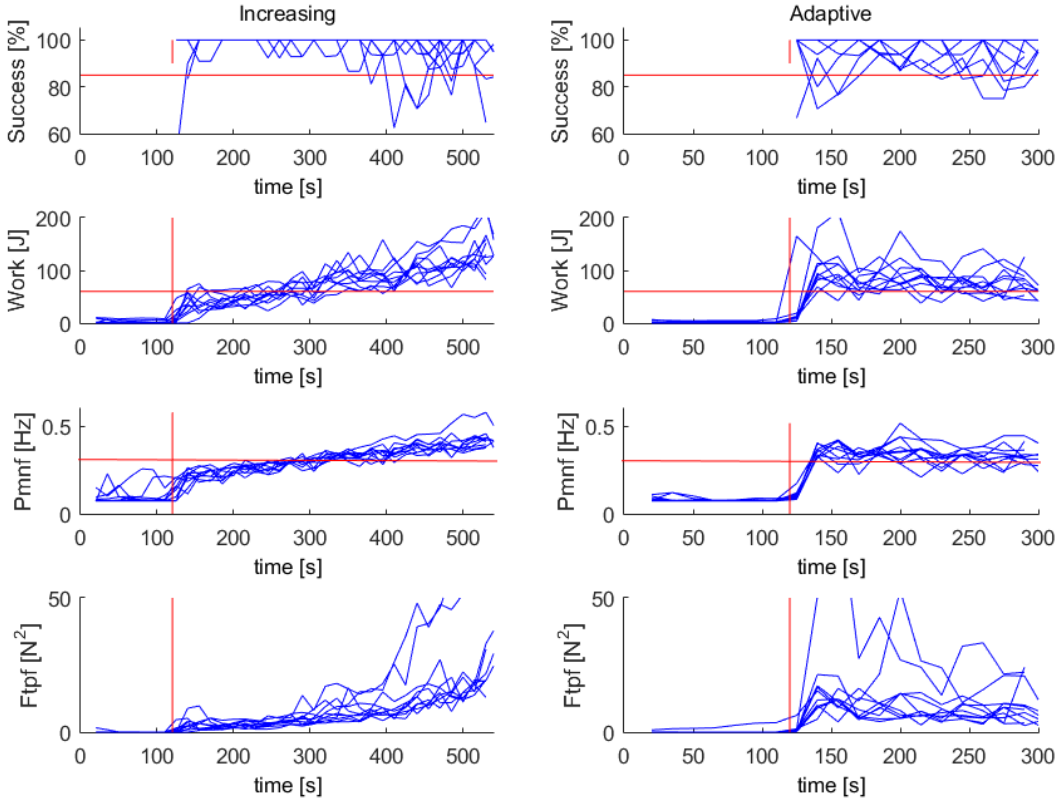


Figure 7.3 Validation of patient model. Figure shows various parameters for patient model validation experiments. Vertical red line designates the moment of the start of the experiment. Time before that is a baseline measurement. Horizontal red line shows the threshold for the parameter set for the patient model. Signals in blue are signals measured for individual subjects that participated in the study. First column shows results for experiment in which the difficulty was steadily increased over time. Second column shows the experiments during which the difficulty was set by a patient model. Results show that the most important parameters are: (1) Success rate (2) Work, (3) Pmnf (position mean frequency) and (4) Ftpf (total power of force signal). Subfigure in first column, second row shows the work during the experiment of increasing the difficulty. From the subfigure it is clearly seen that work performed by the user increases with the increased level of difficulty. From the subfigure in second column, second row it can be seen that the work fluctuates around constant value, which was the aim of the

experiment. The fluctuation is expected since the parameters need to change so that the patient model can either increase or decrease the difficulty based on results in last time window. Similar observations can be made for Pmnf and Ftpf parameters.

First inspection of the results showed that the subjects that participated in the experiments can be divided into two groups based on how their temperature changed during the increasing difficulty experiment:

- group 1 – subjects with positive trend of temperature and
- group 2 – subjects with negative trend of temperature.

Coincidently both groups were the same size of 5 subjects. Results are shown in Figures 7.4 and 7.5.

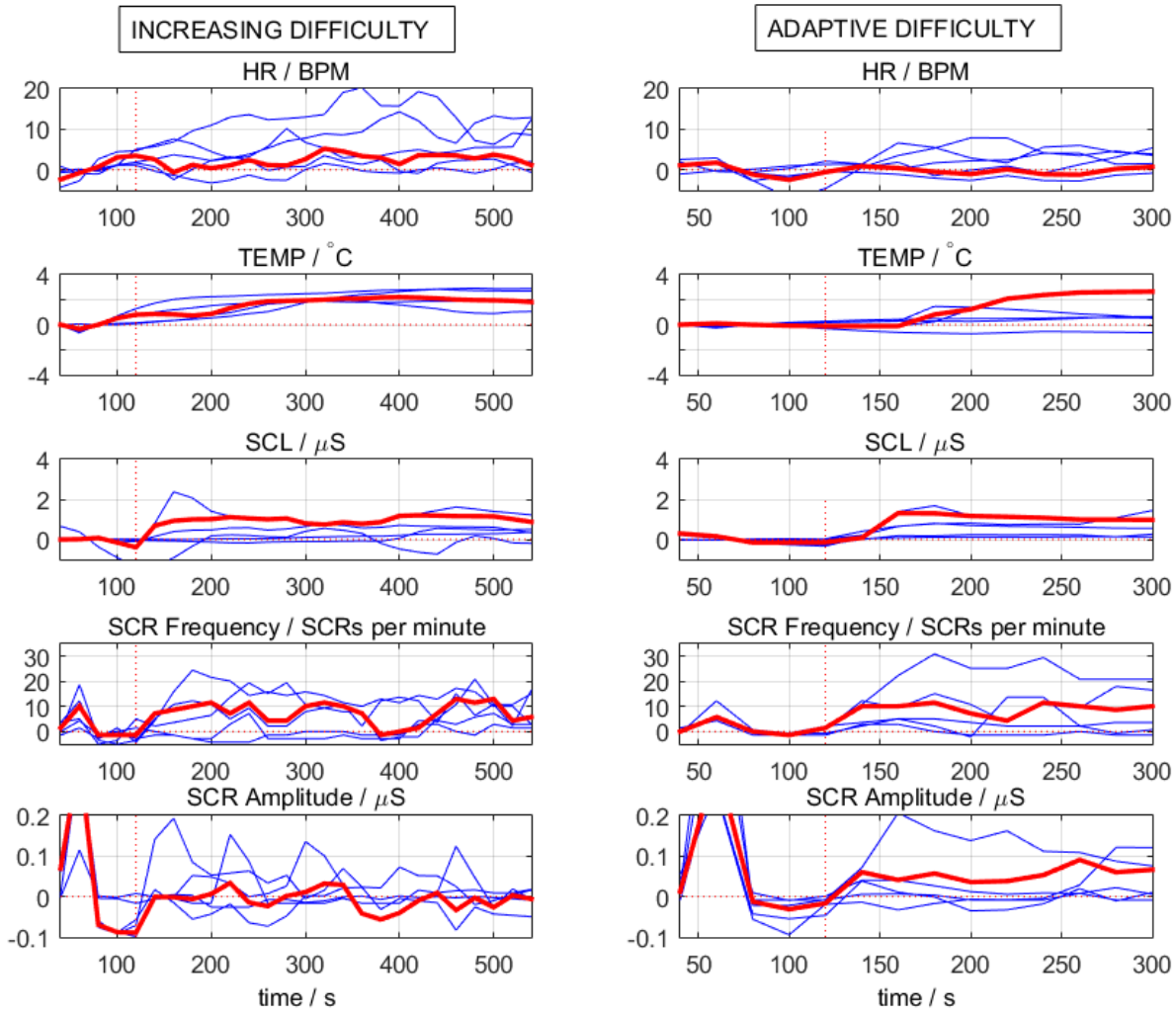


Figure 7.4 Validation of patient model. Signals in blue are signals measured for individual subjects that participated in the study. Red line is a mean of all signals (blue lines). Physiological parameters for subjects of group 1 – subjects with positive trend of temperature. Left column shows results for increasing difficulty experiment and right column shows results for adaptive difficulty experiment. Parameters are: (1) HR – heart rate, (2) TEMP – temperature, (3) SCL – skin conductance level, (4) SCR frequency – skin conductance response frequency, (5) SCR amplitude - skin conductance response amplitude. Parameter SCL seems also to be the only parameter that increases in group 1. Other parameters remain relatively unchanged compared to baseline values in group 1. Second column shows the results of the adaptive difficulty adaptation for physiological parameters. In group 1 heart rate does not change during the experiment compared to baseline values. The temperature parameter in group 1 increases significantly in the second part of the experiment. After the beginning of the experiment the skin conductance parameters values increase compared to baseline values and after the second half of the experiment they remain relatively constant.

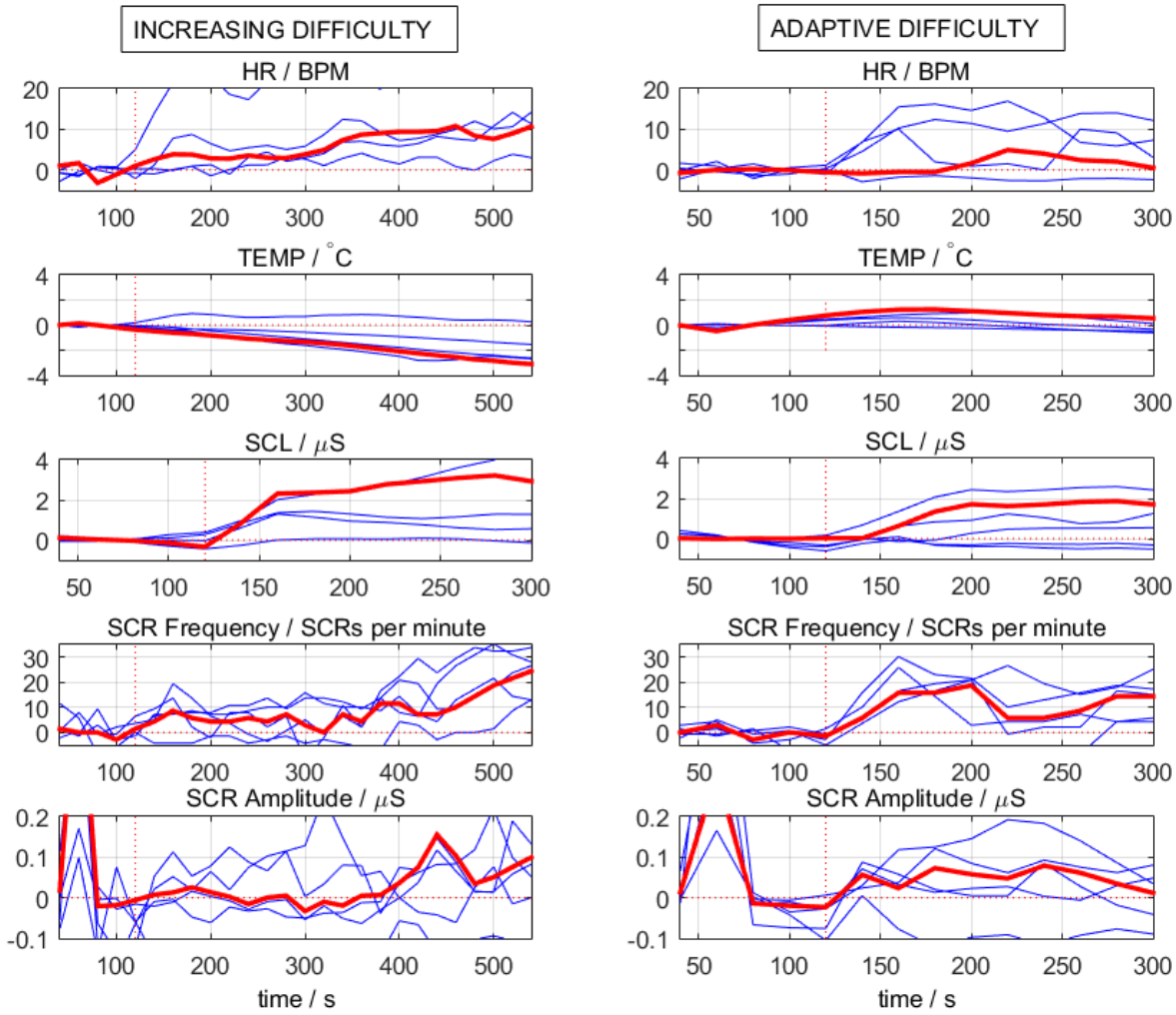


Figure 7.5 Validation of patient model. Signals in blue are signals measured for individual subjects that participated in the study. Red line is a mean of all signals (blue lines). Physiological parameters for subjects of group 2 – subjects with negative trend of temperature. Left column shows results for increasing difficulty experiment and right column shows results for adaptive difficulty experiment. Parameters are: (1) HR – heart rate, (2) TEMP – temperature, (3) SCL – skin conductance level, (4) SCR frequency – skin conductance response frequency, (5) SCR amplitude - skin conductance response amplitude. Heartrate in group 2 is increasing proportionally with the increased difficulty, temperature is decreasing, parameters of skin conductance (SCL, SCR frequency and SCR amplitude – see first column rows 3 to 5) are all increasing from the baseline values. Second column shows the results of the adaptive difficulty adaptation for physiological parameters. In group 2 heart rate increases (second column, first row) after the beginning of the experiment and it stabilizes in the second half of the experiment. In groups 2 in adaptive difficulty experiment temperature slightly, but not significantly increases during the first half of the experiment, but remains mostly constant compared to baseline values. After the beginning of the experiment the skin conductance parameters values increase compared to baseline values and after the second half of the experiment they remain relatively constant.

The patient model was designed to be robust in all circumstances for different types of patients. Therefore, the decision tree is organized in such a way to put more weight on reliable input parameters and less weight on input parameters that might be affected by training scenarios and movement artefacts. The concept was validated and demonstrated reliable on healthy subjects. Experimental data will be required for fine-tuning the model to the specifics of patient population.

8 Control system

Note: *The Control system is described in details in deliverables D2.1 and D2.2.*

The whole platform is managed by the central control system, in charge of controlling devices, synchronizing and updating rehabilitation parameters in accordance with the patient model.
The communication among the nodes of the architecture (Figure 8.1) is performed exploiting USB and UDP communication protocol. In order to facilitate the sharing of data among nodes also the Robotic Operating System (ROS) framework has been exploited.

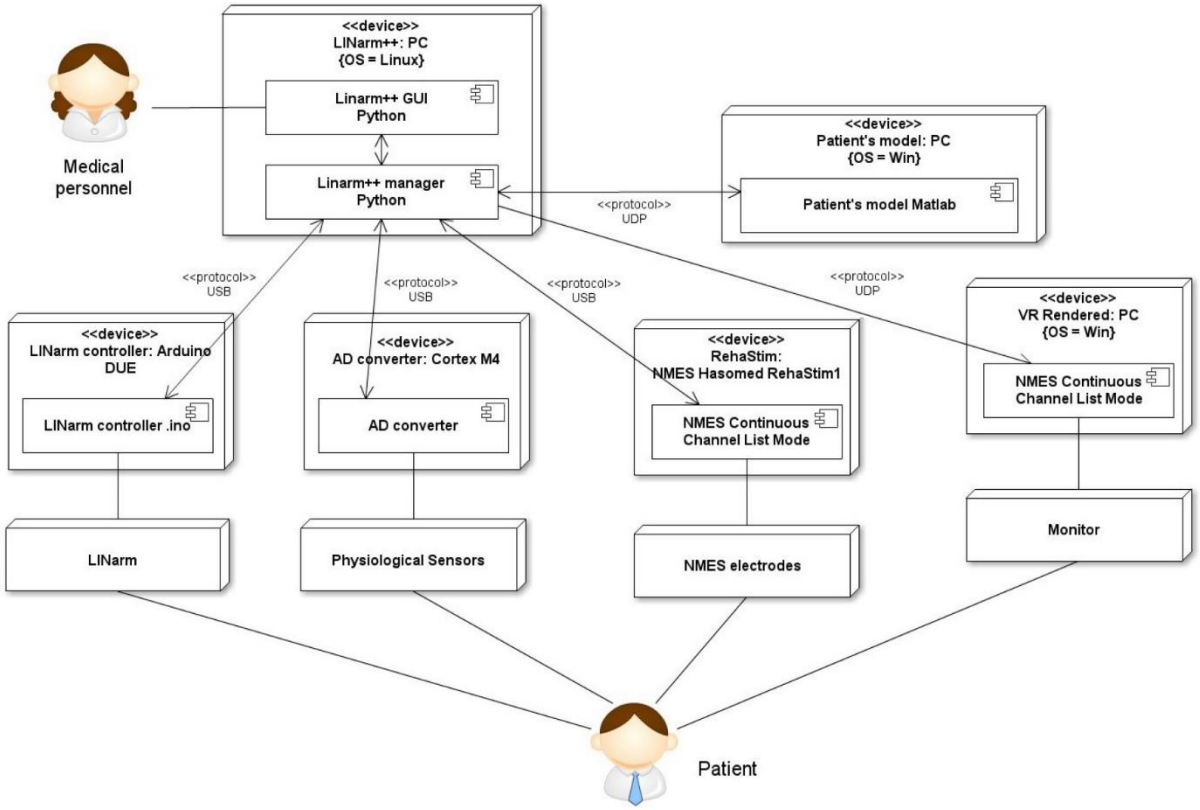


Figure 8.1 – UML representation of the LINarm++ architecture.

A documentation describing in details the software architecture is available. All the ROS nodes of the system have been integrated (Figure 8.2). A simulation mode has been implemented to allow to simulate the functioning of the whole architecture. Graphic user interfaces have been implemented in order to facilitate the interaction of the user with the platform (Figure 8.3). The whole platform can be controlled using the Python scripting language and a set of classes which virtualize the devices of the architecture (Figure 8.4).

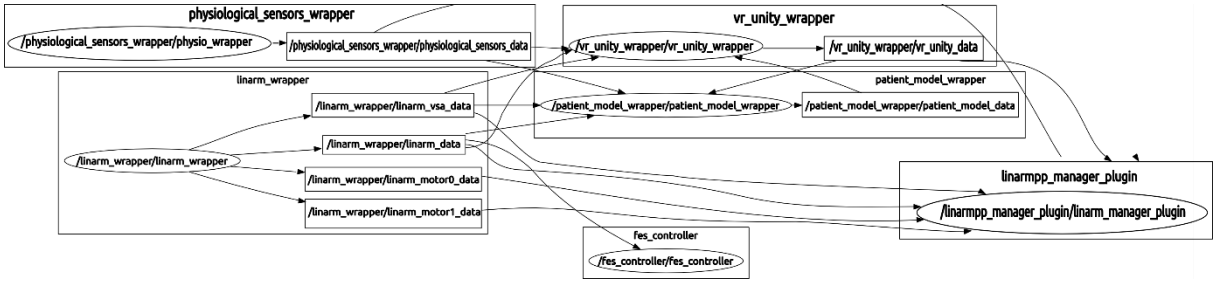


Figure 8.2 – ROS graph for the Linarm++ Manager software and of the "wrappers" in charge of communicating with the devices of the system.

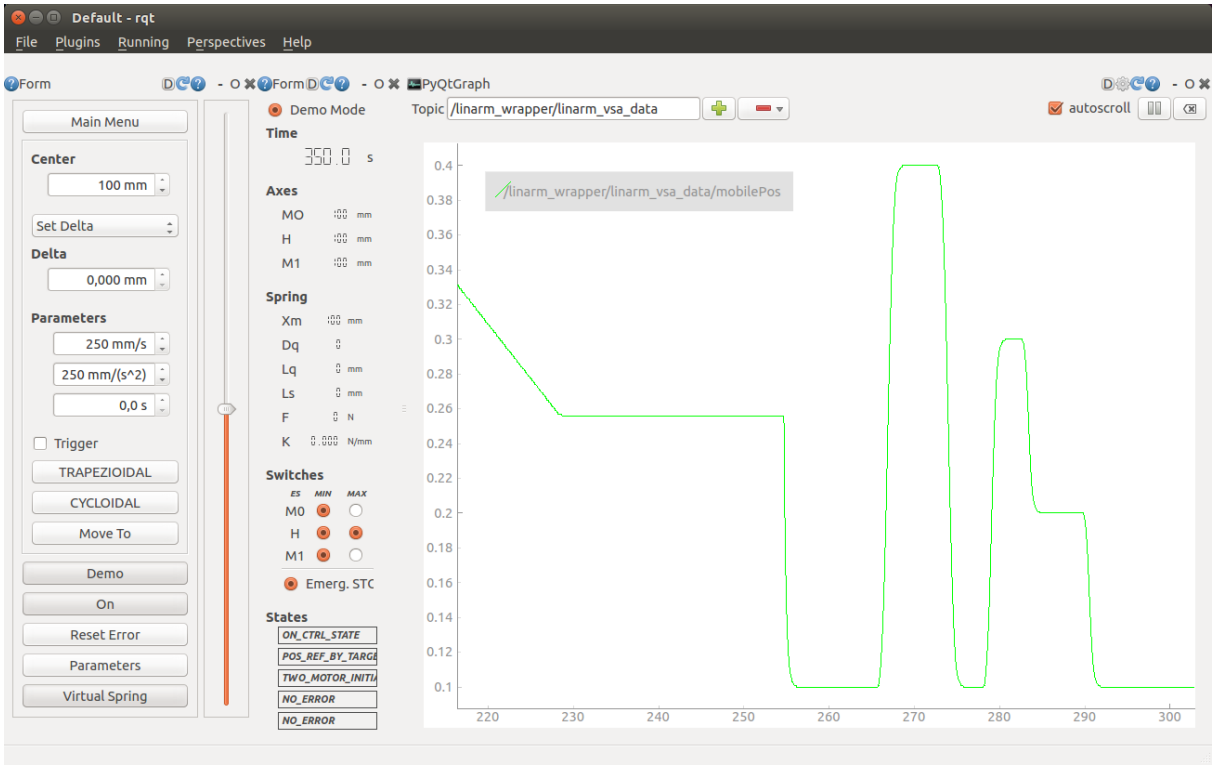


Figure 8.3 – The LINarm++ manager GUI is made up of a control panel (left), a monitor panel (center), a real-time reconfigurable plot to monitor all the system data.

```
# -*- coding: utf-8 -*-
################################################################################
# Parametri
import time
startPoint = 0.10
endPoint = 0.40

# Velocità ciclo
maxVelTrap = 0.150
maxAccTrap = 0.150
minAccTrap = 0.50

numCicli = 4

DeltaPos1 = 0.010

forceThreshold = 30
################################################################################

def waitCondition(cond, loc, timeoutReaching):
    import time
    old = time.time()
    while (eval(cond, globals(), loc) == False and (time.time()- old < timeoutReaching)):
        pass

linarm.setCtrlState(linarm.CONTROL_STATES.IDLE_CTRL_STATE)
linarm.setCtrlMode(linarm.CONTROL_MODES.POS_REF_BY_TARGET_POS)
linarm.setCtrlState(linarm.CONTROL_STATES.ON_CTRL_STATE)

linarm.moveDeltaTo(DeltaPos1,maxVelTrap,maxAccTrap,0, False)


for i in range(0,numCicli):
    waitCondition("physiologicalSensors.physiologicalSensorsData.force>20", locals(), 10)
    linarm.moveCentreTo(endPoint,maxVelTrap,maxAccTrap,0, False)
    waitCondition("abs(endPoint-linarm.linarmVsaState.equilibriumPos)<5e-3", locals(), 10)
    waitCondition("physiologicalSensors.physiologicalSensorsData.force>20", locals(), 10)
    linarm.moveCentreTo(startPoint,maxVelTrap,maxAccTrap,0, False)
    waitCondition("abs(startPoint-linarm.linarmVsaState.equilibriumPos)<5e-3", locals(), 10)

linarm.setCtrlState(linarm.CONTROL_STATES.IDLE_CTRL_STATE)
```

Figure 8.4 – Example of a script to execute a rehabilitation task. The LINarm++ Manager embeds the Python interpreter which allows to program the whole platform exploiting a set of classes which virtualize the devices, i.e. the `linarm` *object allows to control all the funcionalities of the LINarm mechatronic device.*

9 LINarm++ prototype

In this chapter, figures depicting the rehabilitation platform in different configurations with The LINarm++ prototype has been assembled and tested in partner's laboratories,

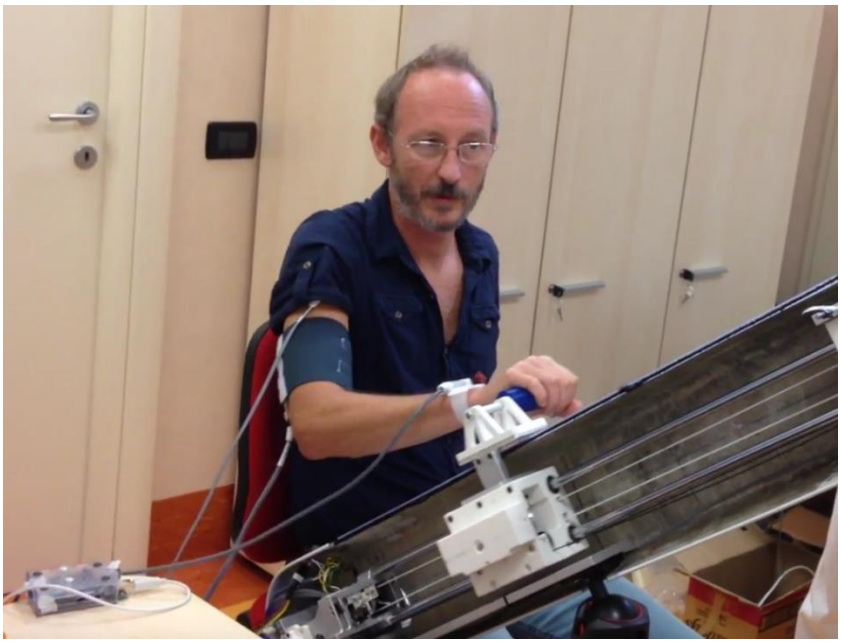


Figure 9.1 – The LINarm++ mechatronic device configured to perform reaching movements. The device is partially covered by the covering in order to show the mechanical structure of the device. The action of the subject is supported by the FES system.

Figure 9.2 – Hand-to-mouth movement performed by a handle connected by a proper mechanism. The connection allows to perform complex functional movements (e.g. hand-to-mouth) exploiting a simple linear movement of the robotic device.

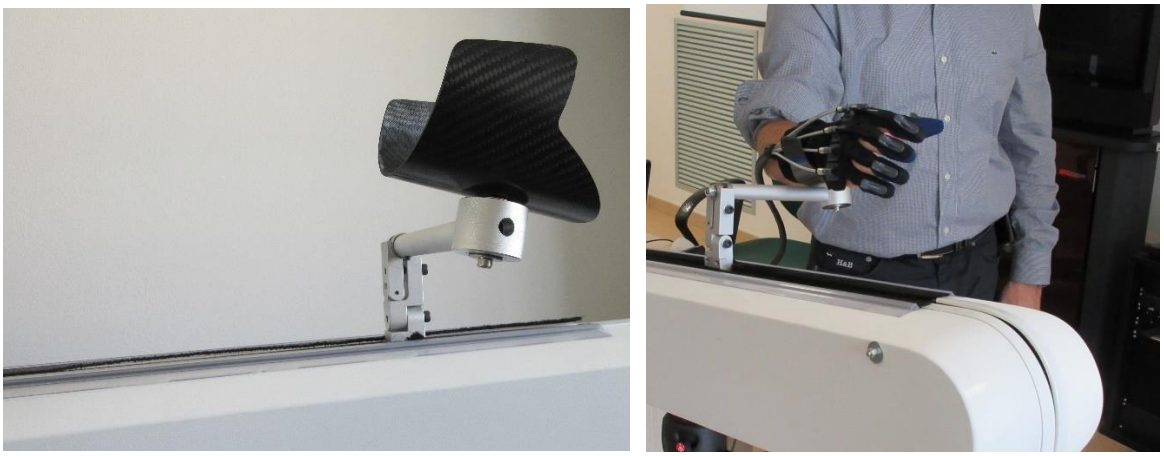


Figure 9.3 – Left: Support for the Gloreha rehabilitation device mounted on the mobile carriage of the LINarm++ device. Right: Use of Gloreha supported by the LINarm++ platform.

Figure 9.4 –LINarm++ prototype interfaced to the virtual environment and with the user holding the cylindrical sensorized handle. In the right picture the rehabilitation task is supported also by the wearable embedding a set of electrodes.

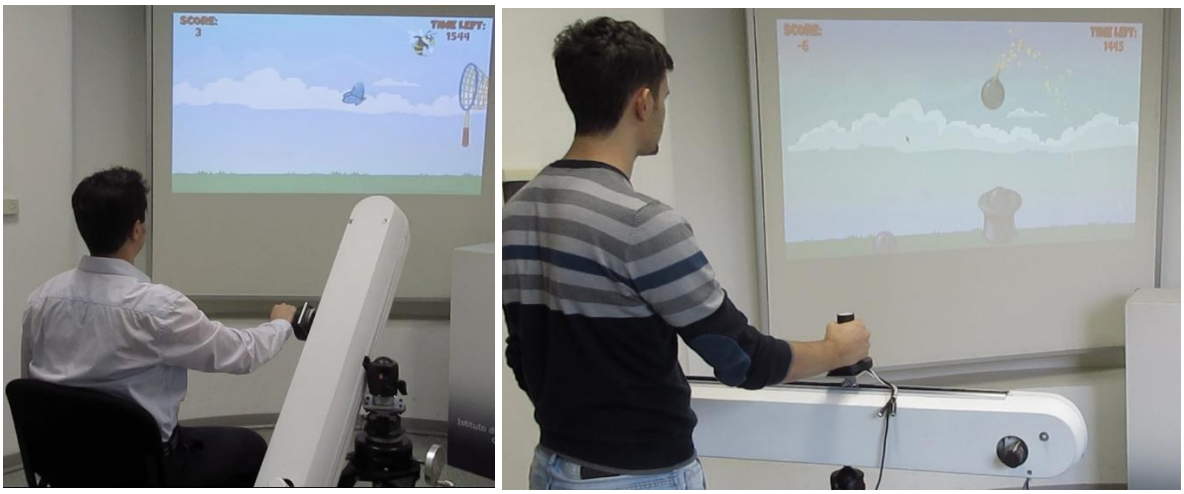


Figure 9.5 – LINarm++ in two different configurations to perform reaching movements (left) and movements normal to the sagittal plane (right).

10 Conclusions

The LINarm++ experimental platform has been completed embedding the functionalities foreseen within the Description of Work and specified at the first stages of the project (Deliverables D1.1, D2.1, D3.1, D4.1, D5.1). In order to analyse the possible actual relapses of the rehabilitation platform, LINarm++ has been the subject of an Exploitation plan (Deliverable D1.2), which included marketability and strategic aspects, including an opinion leader survey by presenting the device in three medical centers, to facilitate an actual exploitation of the results of the project. Moreover, the platform has been subject of a preliminary experimental assessment from the medical point of view: system usability tests have been performed in the Pisa RIF at the Volterra Auxilium Vitae Hospital which involved physiotherapists which tested the various functionalities of the device (Deliverable RIF).

11 References

[1]. Malosio M., et al., "An affordable, adaptable and hybrid assistive device for upper-limb neurorehabilitation", *Journal of Rehabilitation and Assistive Technologies Engineering*, accepted

[2]. Spagnuolo G., et al., "Analysis and synthesis of LinWWC-VSA, a variable stiffness actuator for linear motion", *Mechanism and Machine Theory,* accepted