Journal of Medical Systems

IoT architecture for smart control of an exoskeleton robot in rehabilitation by using a Natural User Interface based on gestures.

--Manuscript Draft--

Manuscript Number:	JOMS-D-19-01434R1
Full Title:	IoT architecture for smart control of an exoskeleton robot in rehabilitation by using a Natural User Interface based on gestures.
Article Type:	SI: Artificial Intelligence in Medicine (AIM)
Section/Category:	Mobile & Wireless Health
Keywords:	Intelligent Robots; IoT; ROS; Exoskeleton; Rehabilitation
Corresponding Author:	Nieves Pavón-Pulido, Ph.D. Universidad Politecnica de Cartagena Cartagena, Murcia SPAIN
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Universidad Politecnica de Cartagena
Corresponding Author's Secondary Institution:	
First Author:	Nieves Pavón-Pulido, Ph.D.
First Author Secondary Information:	
Order of Authors:	Nieves Pavón-Pulido, Ph.D.
	Juan Antonio López-Riquelme, Ph. D
	Jorge Juan Feliú Batlle, Ph. D
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	This paper describes a system for allowing a therapist to record specific motions, as a part of a rehabilitation program, mainly aimed at the elderly people, by using a Natural User Interface based on gestures. Motions are sent to an exoskeleton robot which reproduces them in the patient's lower limbs. The proposed system is an early prototype implemented as of a set of interconnected components, which run independently and remotely, under a distributed software architecture. Such prototype has been properly validated through different tests in a controlled environment. The obtained results and a discussion about benefits and drawbacks are also presented, together with preliminary ideas about future research for overcoming such drawbacks and transforming the prototype into a functional system to be tested in a real scenario with real patients.

IoT architecture for smart control of an exoskeleton robot in rehabilitation by using a Natural User Interface based on gestures.

Pavón-Pulido, Nieves

Technical University of Cartagena. Campus Muralla del Mar C/ Doctor Fleming, s/n. 30202 Cartagena.

+34968325335

+34968325345

nieves.pavon@upct.es

López-Riquelme, Juan Antonio

Technical University of Cartagena. Campus Muralla del Mar C/ Doctor Fleming, s/n. 30202 Cartagena.

+34968325466

+34968325345

jantonio.lopez@upct.es

Feliú-Batlle, Jorge J.

Technical University of Cartagena. Campus Muralla del Mar C/ Doctor Fleming, s/n. 30202 Cartagena.

+34968325390

+34968325345

jorge.feliu@upct.es

Abstract

This paper describes a system for allowing a therapist to record specific motions, as a part of a rehabilitation program, mainly aimed at the elderly people, by using a Natural User Interface based on gestures. Motions are sent to an exoskeleton robot which reproduces them in the patient's lower limbs. The proposed system is an early prototype implemented as of a set of interconnected components, which run independently and remotely, under a distributed software architecture. Such prototype has been properly validated through different tests in a controlled environment. The obtained results and a discussion about benefits and drawbacks are also presented, together with preliminary ideas about future research for overcoming such drawbacks and transforming the prototype into a functional system to be tested in a real scenario with real patients.

Keywords: Intelligent Robots, IoT, ROS, Exoskeleton, Rehabilitation

Introduction.

Population aging is an unquestionable fact that is occurring throughout the world, and life expectancy is also growing, both in developed and developing countries. Although these events are good news, many challenges are emerging from an economic and social point of view [1, 2].

Despite life expectancy is higher, elderly people quality life is not yet as satisfactory as expected since, in many cases, certain diseases affect them more virulently and trigger problems related to mobility, whose lack worsens other illness and conditions related to mental and emotional health [3].

When lack of mobility is not yet very severe, applying suitable rehabilitation programs is usually one of the best options to help patients to recover certain degree of autonomy. Applying some rehabilitation programs involves patients to be displaced to specialized centers because well-trained staff is needed for this purpose; specific assistive tools and equipment are required; and, typically, a direct contact between patient and therapist is needed.

On the other hand, saturation of waiting lists, due to the medical staff deficient size and the growth of the number of patients, and often related to lack of budget, and logistics drawbacks, is also a problem. In these circumstances, technical advances based on the application of Information and Communication Technologies (ICTs), in particular, Artificial Intelligence, Internet of Things (IoT) and Cloud Computing, among other related paradigms, could help the aforementioned difficulties to be minimized and, even, overcome.

Nowadays, many well-established research lines are focused on developing solutions for improving elderly people quality life. Many of them are based on the application of Ambient Assisted Living (AAL) techniques [4], combined to other disciplines, such as Robotics [5, 6] and Artificial Intelligence (AI), or Telemedicine [7], which contributes to the remote assistance of patients, avoiding unnecessary trips and optimizing medical resources. However, in the field of rehabilitation, specialized staff and equipment are unavoidable requirements. Consequently, solutions based on developing complex remote rehabilitation systems are not actually frequent.

An interesting research line, focused on improving rehabilitation technology by using ICTs, is based on the use of external exoskeleton robots [8], designed as enablers for rehabilitation exercises. However, the integration of hardware, firmware and low and high level software components is still a challenge; even more, if usable systems (for both, therapists and patients), are intended to be developed. In [9], an early prototype for remote rehabilitation that uses an exoskeleton (in the patient side), and a Kinect v2 sensor [10] used as a Natural User Interface (NUI), based on gestures [11] (in the therapist side), is described. However, several drawbacks were identified, mainly related to the compatibility of the selected NUI sensor.

In this paper an improvement of such system is presented. The architecture of the system has been enhanced by including a more compatible RGB-D sensor, and new algorithms have been developed for improving the transference of motion between the therapist and the patient.

The outline of the paper is as follows. Section 2 briefly describes the previous system architecture proposed in [9], highlighting which components have been modified. Section 3 shows the tests designed for testing the system and the obtained results. An analysis and a discussion about such outcomes is presented in Section 4. Finally, Section 5 presents the main conclusions and the work expected to be carried out in the near future.

System architecture.

The system presented in [9] comprises a collection of software modules, running under the Robotic Operating System (ROS) [12], which allow an exoskeleton robot to be remotely controlled through sending angular position commands corresponding to each joint. Such commands are generated by recording and processing the therapist’s motion, in real time, using a NUI based on gestures, whose main component is a Kinect V2 RGB-D sensor.

Among other reasons, the discontinuation of such sensor, its lack of compatibility with the Linux Operating System (OS), and the privative character of the Kinect Software Development Kit (SDK), have led to design and release a new version of the system.

Hardware architecture.

The exoskeleton robot is the key element that enables the remote contact between therapist and patient. Specifically, an exoskeleton developed under EXO-LEGS [13], a European project (AAL-010000-2012-15), included in the Ambient

Assisted Living Joint Programme, has been used. It consists of two legs with three identically motorized joints per leg (hip, knee and ankle). Table 1 shows the main features of the hardware elements [14], which configure a specific joint.

As the purpose of the system, in this first stage, is helping people to strengthen the lower body, the exoskeleton is supported by an auxiliary structure. It is expected the subject to remain standing while he or she performs the programmed motions, helped by the exoskeleton.

The way in which the therapist defines exercises through specific motions should be, somehow, recorded and sent to the exoskeleton in a reliable way. The RGB-D sensor is another key element of the hardware architecture, since it allows the therapist to record the motion expected to be remotely reproduced by the patient with the help of the exoskeleton. The previous version of the system was equipped with a Kinect V2 sensor, which uses time-of-flight (ToF) for depth sensing, and only works under Windows. The current version uses a structured-light Asus Xtion Pro Live [15] camera compatible with different operating systems and with ROS (see Fig. 1). Table 2 summarizes the main differences between both cameras.

Note that, the Asus Xtion Pro Live model was selected because it was available at the starting point of the prototype development, and a functional skeleton tracking SDK was still accessible. However, nowadays, both, the sensor and the skeleton tracking SDK are also discontinued and, consequently, a new model of camera will be included in the final version of the system. In particular, one marketed by the Orbbec company which offers an RGB-D solution similar to the Asus Xtion Pro Live.

Software architecture.

The proposed software architecture is distributed according to the kind of user: therapist or patient. The motion of the therapist’s legs is sensed through an RGB-D camera; then, the angular positions of the main representative joints of his/her legs are acquired, processed and sent to the exoskeleton, to be reproduced in the patient lower limbs.

Fig. 2 shows the set of software components (and their relations), which comprise the early version of the architecture described in [9], alongside the modifications included in the new system.

ROS ecosystem.

ROS is a “flexible framework for writing robot software”, which could be considered, at the lowest level, as a middleware that provides facilities for robot software development. These features help the goals for the proposed rehabilitation system to be achieved.

The main software modules of the new proposed architecture, specifically, the exoskeleton controller and the RGB-D sensor controller, are designed as packages within the ROS ecosystem and, consequently, they are executed as ROS nodes, i.e. processes that perform computation, all of them combined into a graph, centralized in the ROS Master node, which unifies communications over XMLRPC (eXtensible Markup Language Remote Procedure Call). These nodes exchange information through ROS topics (named buses with an anonymous publish/subscribe semantics, which decouples production and consumption of information).

In the very first version of the system, since the Kinect V2 sensor is only compatible under Windows, the compatibility layer WSL (Windows Subsystem for Linux), was used for including the camera controller as part of the ROS ecosystem; however, this approach made the architecture more complex, necessarily forced the installation of Windows 10 and using the Kinect V2 sensor demanded several specific features, in terms of performance, in the computer which was connected to.

These drawbacks led an alternative RGB-D sensor to be selected. In particular, the second (and currently used), version of the system is equipped with the above-mentioned Asus Xtion Pro Live model, which is directly connected to a PC under Linux and whose controller is implemented as a ROS node. However, the therapist computer could run another OS. Another solution for avoiding such prerequisite (which is being included in the next release version of the software), is developed by using the Rosbridge suite, designed to provide a JSON API to ROS functionality for non-ROS programs running in any OS. Thus, any RGB-D sensor could be used by simply writing a controller capable of acquiring and processing the visual information (including the skeleton detection and tracking), and sending it to the ROS ecosystem through Rosbridge.

Human skeleton detection and tracking.

An RGB-D sensor is a depth sensing device that works in association with an RGB camera for augmenting a conventional image with depth information, in a per-pixel basis. In the context of rehabilitation, the interesting information about depth is related to the position of the main human body joints, and it is obtained by applying skeleton tracking techniques.

If only a Kinect V2 sensor is present, the Microsoft SDK for skeleton tracking is used. However, if the system is equipped with an Asus Xtion Pro Live camera, the NiTE2 middleware based on OpenNI2 (an open source project focused on facilitating access to NUI devices) [16], is used for skeleton tracking.

Both, Microsoft SDK and NiTE2 [17] are not open-source resources, and the techniques used for skeleton recognition cannot be well-known and, consequently, modified. According to literature, skeleton tracking with a Kinect V2 sensor could be a procedure similar to that described in [18], which proposes a method to predict 3D positions of body joints from a single depth image, through an intermediate body parts representation, obtained from using a deep randomized decision forest classifier, trained with hundreds of thousands of training images (both real and synthetic ones). On the other hand, references about NiTE2, beyond those explaining how to use the API, are not available.

In both cases, the tracking process starts when the target is detected with its first motion, and such target is lost when it leaves the field of view of the sensor or it is almost totally occluded.

The ROS package for controlling the exoskeleton.

Each exoskeleton joint is implemented by a hardware module controlled by its own EPOS2 50/5 controller, which can be programmed by using a Linux Shared Object Library libEPOSCmd.so, provided by the manufacturer. The implemented ROS package allows control commands and data about the state to be exchanged with the exoskeleton by using such library.

When the ROS node is launched, each motor configuration is loaded and two topics per joint are created, one for sending the control command (angular position), and one for receiving the state of the joint. The designed ROS package is based on using the ros_control framework [19], which main advantage is that robot interfaces and controllers needed in the system are already implemented.

The exoskeleton controller uses two hardware interfaces per joint (Joint State Interface and Position Joint Interface), defined in ros_control. Therefore, it is possible to launch an existing Joint Position Controller to set each joint position, and a Joint State Controller for publishing each joint state. All the functionality of the exoskeleton mechanism has been implemented by a C++ class that inherits of the hardware_interface::RobotHW class, defined in the hardware_interface package. The code of the main program follows the typical outline defined by the ros_control framework, that is, a loop for reading the state, updating the controller and writing the actuation commands for each joint. Fig. 3 shows how the exoskeleton is controlled through the ros_control framework.

The ROS package for controlling the RGB-D camera.

The ROS package that controls the Asus Xtion Pro Live camera uses the NiTE2 middleware to access the detected skeleton. The ROS node creates six topics for sending angular positions corresponding to the hip, knee and ankle joints calculated from the detected skeleton, which are continuously generated over time. Each list of angles, corresponding to a specific part of the body, is considered as a noisy signal with a high frequency noise. It is necessary to avoid the transmission of such noise to the exoskeleton; therefore, a previous filtering stage is needed for ensuring the exoskeleton carries out a smooth motion. In fact, a low pass filter, based on the statistical mean, which also takes into account if the difference between consecutive angles is high enough for sending a new value to the exoskeleton, according to a heuristically defined threshold, is continually applied over each list of angles generated during the last past short interval of time. Thus, the application of the filter does not have influence in the final performance of the exoskeleton, when it reproduces a specific exercise, since the computational cost of the used algorithm is very low.

Fig. 4 summarizes how this node works. Note that, under Windows, instead of directly launching a ROS node, a program with the same functionality for controlling the camera is executed, and the Rosbridge suite is used to create the needed topics.

Results.

During the software development process, a test plan has been designed for validating, first, each component separately and, then the integration of all the modules. As this early prototype could not be validated in a medical context, yet, several tests have been designed to demonstrate the performance of the system, mainly focused on verifying that it is possible to reliably operate the exoskeleton by simply using an RGB-D sensor in real time, in a remote manner. Table 3 summarizes the set of tests carried out during the time of development of the current version of the prototype.

First, the empty exoskeleton has been tested by sending position commands to each joint within a wide range of possible poses, using the ROS tools for managing topics and analyzing the results through the ROS Robot Visualizer Rviz. The software component that controls the RGB-D sensor has also been tested, both independently and together with the Rviz exoskeleton simulator (see Fig. 5), by recording several motions, which have been analyzed to know how noise affects and how filtering solves the problem (see Fig. 6). Finally, the global system has been verified with a user in the role of therapist, when the exoskeleton is empty, as Fig. 7 shows, and with another user in the role of patient that wears the exoskeleton (see Fig. 8). This process has been repeated with different people, with the aim of analyzing how the physical characteristics of each person could influence the performance of the skeleton tracking algorithm, and testing how difficult is the control of the exoskeleton by using this NUI. In all the tests, the camera module is capable of detecting the lower body and calculating and send the angles of the joints to the exoskeleton in a reliable way.

Discussion.

The obtained preliminary results are positive and the main benefits offered by the system are:

- It could be highly usable for therapists, and the usage learning curve seems to be shallow. However, more tests will be needed when the system is deployed in a medical environment.
- Therapists could record motion exercises to be reproduced by the exoskeleton, for different patients, in different moments.

- Therapists could evaluate the results of the exercises, since the joints state of the exoskeleton could be properly saved during exercise execution.
- Patients could use the exoskeleton without the need of the presence of the therapist, under the supervision of an assistant, at any local clinic.

The new version of the prototype has overcome some problems detected in the previous version, in particular:

- The system architecture has been modified to include a more compatible RGB-D sensor. As mentioned before, since the Asus Xtion Pro Live sensor model is already discontinued, the next release of the proposed system will include a new camera, marketed by the Orbbec company, with the same features, and compatible with OpenNI. Moreover, such company also provides its own SDK that includes body tracking capabilities, thus solving the problem with the NiTE2 SDK, which is also discontinued.
- If the hardware architecture is distributed through Internet (but not in the same Intranet), a Virtual Private Network (VPN), is needed (see Fig. 2), if all the components are written as ROS nodes. The Rosbridge suite enhances the compatibility of ROS and non ROS components and partially solves this problem. However, it is necessary to improve the software architecture to make the system more usable for both, therapists and patients and, therefore, it is being extended by using a Cloud Computing-based solution, by adding a backend based on Google Cloud Endpoint services, and a frontend (graphical user interface), based on HTML5 and JavaScript, which is compatible with the Rosbridge library needed for enabling the access to the ROS components that control the exoskeleton. The extended software architecture will enable therapists to program and design rehabilitation exercises, and patients to perform such exercises, which could be synchronously and/or asynchronously evaluated by the therapist.

Conclusions and future work.

A system for allowing therapists to remotely control an exoskeleton for rehabilitation purposes has been developed, demonstrating that it is possible to apply remote rehabilitation and so, it is feasible to solve, or at least minimize, the logistic issues that handicap the application of certain treatments.

The very first prototype has been improved in terms of compatibility and simplicity and, although it remains an early prototype, it paves the path to get a better functional solution, ready to be tested in a real environment. That is precisely the goal of the future research, whose main agenda is: (i) avoiding the dependence of using existing privative algorithms for skeleton tracking, by developing a new Deep Learning-based method focused on detecting the lower limbs in all kind of poses; (ii) using Cloud Computing for improving the compatibility of the system and (iii) testing it with real people, after including new biomedical sensors to measure the effectiveness of the rehabilitation exercises done while wearing the exoskeleton.

Acknowledgements

The engineer Jesús Damián Blasco deserves a special mention in this work for his support in several technical aspects. This article is the result of the activity carried out under the “Research Programme for Groups of Scientific Excellence at Region of Murcia” of the Seneca Foundation (Agency for Science and Technology of the Region of Murcia – 19895/GERM/15) and with the support of the research group Neurocor (Neurotechnology, Control and Robotics) at the Technical University of Cartagena, Spain.

Compliance with Ethical Standards:

Conflict of Interest: Author Nieves Pavón-Pulido declares that she has no conflict of interest. Author Juan Antonio López-Riquelme declares that he has no conflict of interest. Author Jorge J. Feliú-Batlle declares that he has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Lutz, W., Sanderson, W., Scherbov, S.: The coming acceleration of global population ageing. Nature Int. J. Sci. 451(7179), 716–719 (2008)
2. United Nations: World Population Ageing. Report, New York (2017)
3. Tinetti, M.E.: Performance-oriented assessment of mobility problems in elderly patients. J. Am. Geriatr. Soc. 34, 119–126 (1986)
4. Memon, M., Wagner, S.R., Pedersen, C.F., Beevi, F., Hansen, F.O.: Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. Sensors 14(3), 4312–4341 (2014)

5. Pavón-Pulido, N., López-Riquelme, J.A., Ferruz-Melero, J., Vega Rodríguez, M.A., BarriosLeón, A.J.: A service robot for monitoring elderly people in the context of Ambient Assisted Living. J. Ambient Intell. Smart Environ. 6(6), 595–621 (2014)
6. Pavón-Pulido, N., López-Riquelme, J.A., Pinuaga-Cascales, J.J., Morais, R.: Cybi: a smart companion robot for elderly people: improving teleoperation and telepresence skills by combining cloud computing technologies and fuzzy logic. In: Proceedings on IEEE International Conference on Autonomous Robot Systems and Competitions, Vila Real, pp. 198–203 (2015)
7. Bujnowska-Fedak, M.M., Grata-Borkowska, U.: Use of telemedicine-based care for the aging and elderly: promises and pitfalls. Smart Homecare Technol. TeleHealth 3, 91 –105 (2015)
8. Younbaek, L., et al.: Biomechanical design of a novel flexible exoskeleton for lower extremities. IEEE/ASME Trans. Mechatron. 22(5), 2058–2069 (2017)
9. Pavón-Pulido, N., López-Riquelme, J.A., Feliú-Batlle, J.: Intelligent Control of an Exoskeleton for Rehabilitation Purposes Using a ROS-Based Software Architecture. EPIA 2019. Conference on Artificial Intelligence. Lectures Notes in Artificial Intelligence 11084, 349-360 (2019)
10. Microsoft Kinect Homepage. https://developer.microsoft.com/es-es/windows/kinect. Accessed 21 Nov 2016
11. Pirker, J., Pojer, M., Holzinger, A., Gütl, C.: Gesture-based interactions in video games with the leap motion controller. In: Kurosu, M. (ed.) HCI 2017. LNCS, vol. 10271, pp. 620–633. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58071-5_47
12. Quigley, M., et al.: ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software, vol. 3 (2009)
13. ExoLegs Homepage. http://www.aal-europe.eu/projects/exo-legs/. Accessed 29 Oct 2019
14. Maxon Motor Homepage. https://www.maxonmotor.com/maxon/view/product/control/Positionierung/347717. Accessed 29 Oct 2019
15. Asus Xtion Pro Live Homepage. https://www.asus.com/es/3D-Sensor/Xtion_PRO_LIVE/. Accessed 29 Oct 2019
16. OpenNI2 Homepage. https://structure.io/openni. Accessed 29 Oct 2019
17. Sinthanayothin, C., Bholsithi, W., Wongwaen, N.: Skeleton Tracking using Kinect Sensor & Displaying in 3D Virtual Scene. International Journal of Advancements in Computing Technology. 4(11), 213-223 (2012)
18. Shottont, J., et al.: Real-Time Human Pose Recognition in Parts from Single Depth Images. Proceedings of CVPR. Computer Vision and Pattern Recognition, pp. 1297-1304, (2011)
19. Chitta, S., et al.: ros_control: a generic and simple control framework for ROS. J. Open Source Softw. 2(20), 456, 1–5 (2017)

Table 1. Main features of the Maxon devices that comprise each joint.

Table 2. Main features of the RGB-D cameras: Res. represents resolution, F represents the number of frames per second, FOV is Field of View (H for horizontal and V for vertical), measured in degrees, and OMR is Operative Measuring Range in meters (related to depth).

Table 3. Summary of the test plan developed for validating the performance of the current system, using the Asus Xtion Pro Live model as RGB-D sensor, the real exoskeleton (empty and worn by users), and running all the ROS nodes under the same intranet.

Fig. 1. Hardware architecture of the rehabilitation system based on an exoskeleton robot remotely controlled by using gestures for motion recording through a NUI.

Fig. 2. Software architecture of the rehabilitation system. In the previous version, a Kinect V2 sensor is used. The current version is equipped with an Asus Xtion Pro Live RGB-D sensor, directly compatible with the Linux environment. For processes running Windows, it is possible to use Rosbridge to access the ROS ecosystem.

Fig. 3. Outline of the ROS package that allows the exoskeleton control to be carried out. Since the Epos2 50/5 internally executes a control feedback loop to reach the position, the ROS software controller is really empty. Only it is necessary to read the state, to prove if the position is reached and to write the control command. The behavior of each hardware module for controlling each joint is implemented by the Epos2 class.

Fig. 4. Outline of the RGB-D camera software controller. The Linux version of the RGB-D controller is directly implemented as a ROS node. However, if the RGB-D controller version for Windows is executed, the program uses the same angles extraction algorithm than in the Linux version, but applies the commands within the ROS ecosystem through the Rosbridge suite.

Fig. 5. Experiment to test the performance of the RGB-D camera controller in the Linux environment, executed as a ROS node. A set of joint angles, describing the motion trajectory, has been recorded, filtered and sent to the simulation of the exoskeleton implemented by Rviz under ROS.

Fig. 6. Experiment that shows the result of filtering the original signal generated as the result of the evolution of the knee angle, after repeating the motion of elevating the knee several times.

Fig. 7. Experiment to test the real exoskeleton empty. The user that acts as a therapist is in front of the RGB-D camera. A set of Position Joint Commands, which define the trajectory after executing the motion of elevating and descending the knee, are calculated from the human skeleton, detected taking the depth image captured by the camera as input. The trajectory is properly reproduced by the exoskeleton in real time.

Fig. 8. Experiment to test the system when a user is wearing the exoskeleton. The user acting as a therapist has carried out several motions elevating and descending the knee. The exoskeleton reproduces the motion in real time.

Device marketed by Maxon	**Main features**
Brushless EC 60 flat motor.	400 W, 48 V, Hall sensors, 470 g.
Planetary gearhead.	Reduction relation of 169:9.
MILE inductive rotatory encoder.	1024 pulse per revolution.
EPOS2 50/5 smart motion controller	USB, RS232 and CAN interfaces

Table 1. Main features of the Maxon devices that comprise each joint.

Sensor	**Color**		**Depth**		**FOV**		**OMR meters**
	Res.	**F**	**Res.**	**F**	**H**	**V**	
Kinect V2	1920×1080	30 fps	512×424	30 fps	70º	60º	[0.5-4.5]
Asus Xtion Pro Live	1280×1024	30 fps	640×480	30 fps	58º	45º	[0.8-3.5]

Table 2. Main features of the RGB-D cameras: Res. represents resolution, F represents the number of frames per second, FOV is Field of View (H for horizontal and V for vertical), measured in degrees, and OMR is Operative Measuring Range in meters (related to depth).

Test	Description	Result	Stage
1	Sending commands to each motor controller independently.	Validated	Implementation of the exoskeleton controller
2	Sending commands to all the motor controllers by using ros_control.	Validated	
3	Skeleton tracking with different users in different settings.	Validated	Implementation of the camera controller
4	Skeleton tracking for sending angles to the simulated exoskeleton in Rviz.	Noise detected	
5	Skeleton tracking for sending angles, after filtering, to the simulated exoskeleton in Rviz.	Validated	
6	Testing the usability of the NUI.	Validated	
7	Testing the system with the empty exoskeleton.	Validated	Deployment
8	Testing the system with a user in the role of therapist and another in the role of patient.	Validated	

Table 3. Summary of the test plan developed for validating the performance of the current system, using the Asus Xtion Pro Live model as RGB-D sensor, the real exoskeleton (empty and worn by users), and running all the ROS nodes under the same intranet.

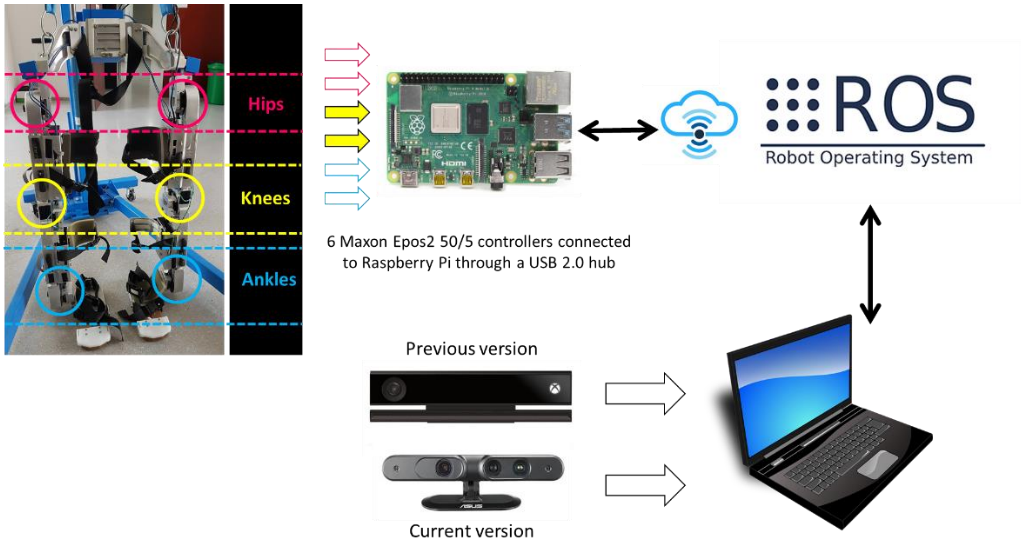

Fig. 1. Hardware architecture of the rehabilitation system based on an exoskeleton robot remotely controlled by using gestures for motion recording through a NUI.

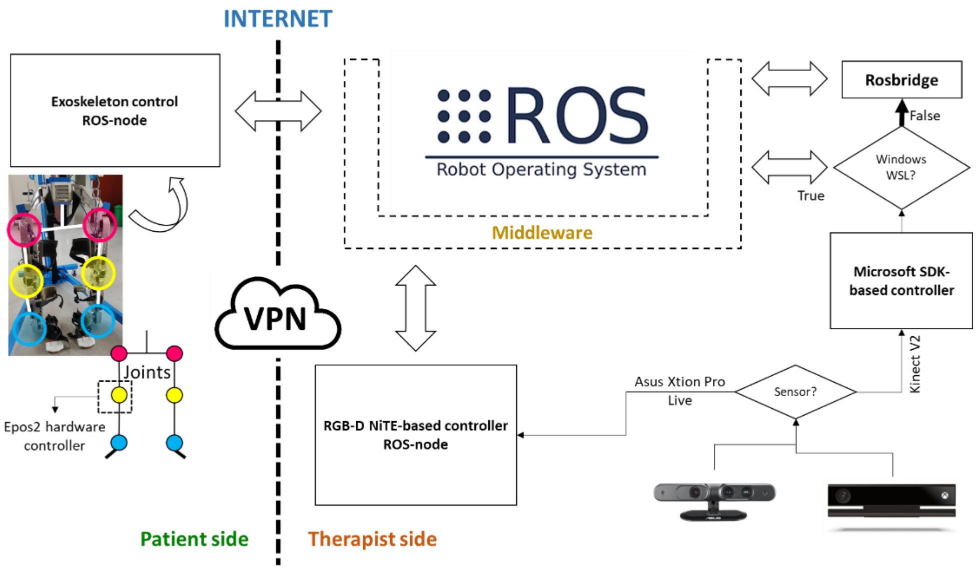

Fig. 2. Software architecture of the rehabilitation system. In the previous version, a Kinect V2 sensor is used. The current version is equipped with an Asus Xtion Pro Live RGB-D sensor, directly compatible with the Linux environment. For processes running Windows, it is possible to use Rosbridge to access the ROS ecosystem.

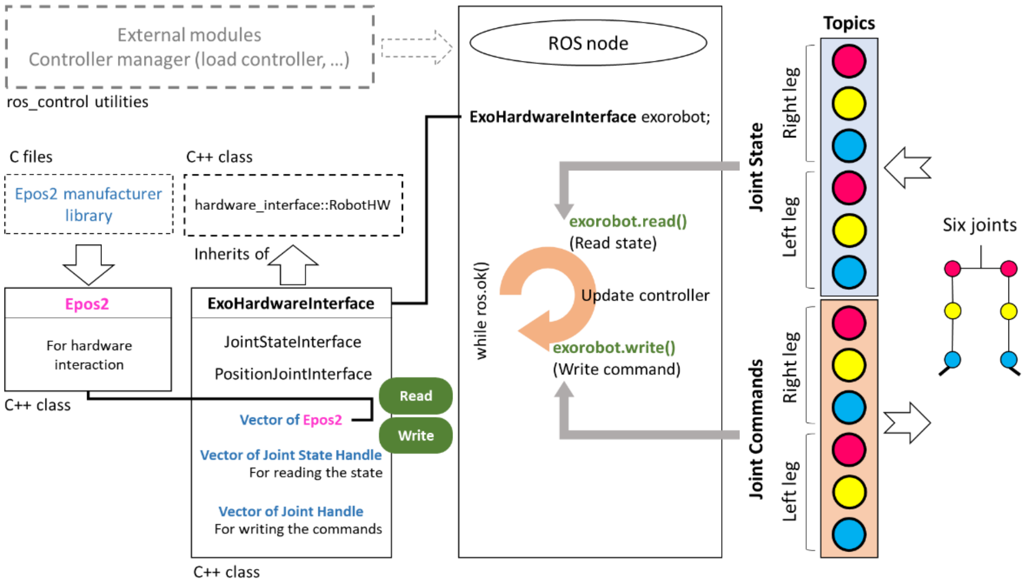

Fig. 3. Outline of the ROS package that allows the exoskeleton control to be carried out. Since the Epos2 50/5 internally executes a control feedback loop to reach the position, the ROS software controller is really empty. Only it is necessary to read the state, to prove if the position is reached and to write the control command. The behavior of each hardware module for controlling each joint is implemented by the Epos2 class.

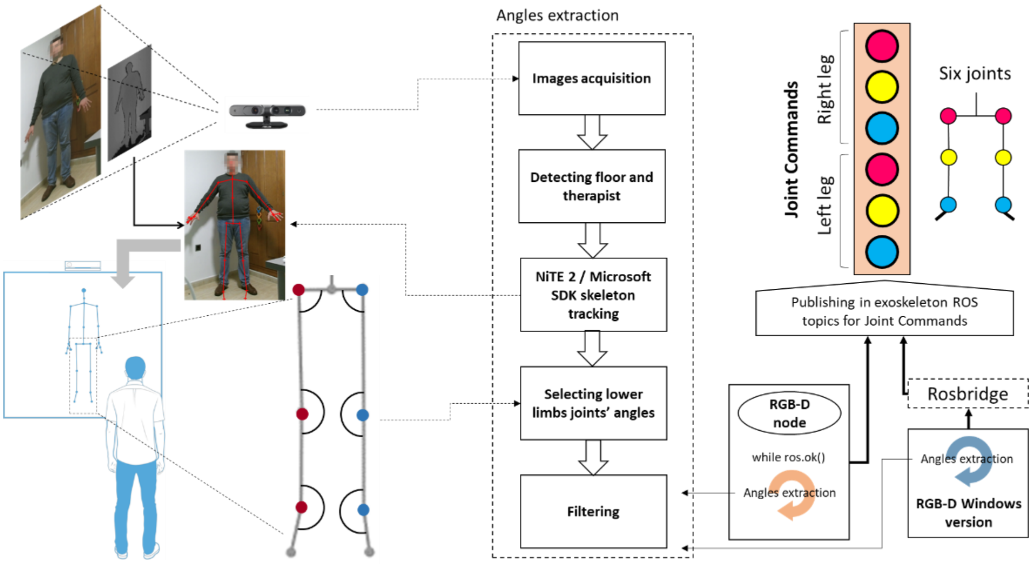

Fig. 4. Outline of the RGB-D camera software controller. The Linux version of the RGB-D controller is directly implemented as a ROS node. However, if the RGB-D controller version for Windows is executed, the program uses the same angles extraction algorithm than in the Linux version, but applies the commands within the ROS ecosystem through the Rosbridge suite.

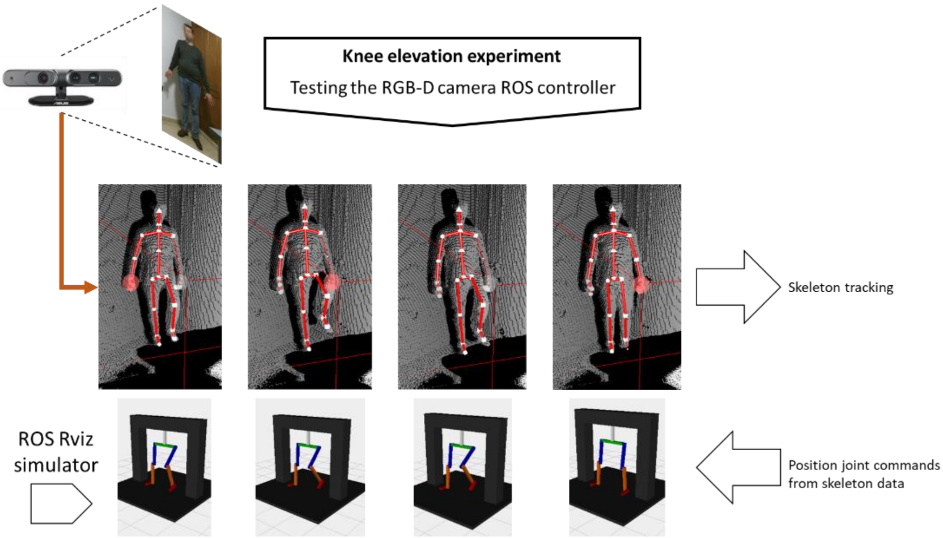

Fig. 5. Experiment to test the performance of the RGB-D camera controller in the Linux environment, executed as a ROS node. A set of joint angles, describing the motion trajectory, has been recorded, filtered and sent to the simulation of the exoskeleton implemented by Rviz under ROS.

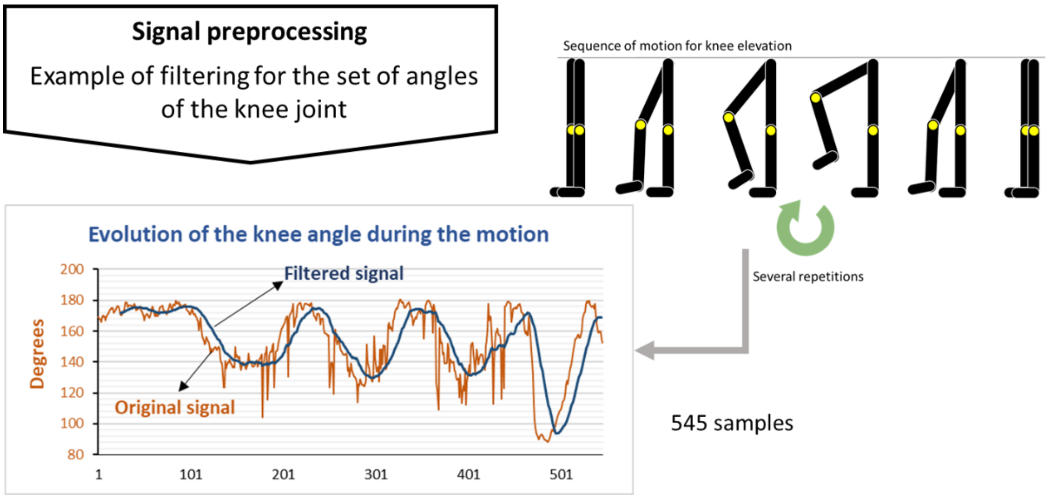

Fig. 6. Experiment that shows the result of filtering the original signal generated as the result of the evolution of the knee angle, after repeating the motion of elevating the knee several times.

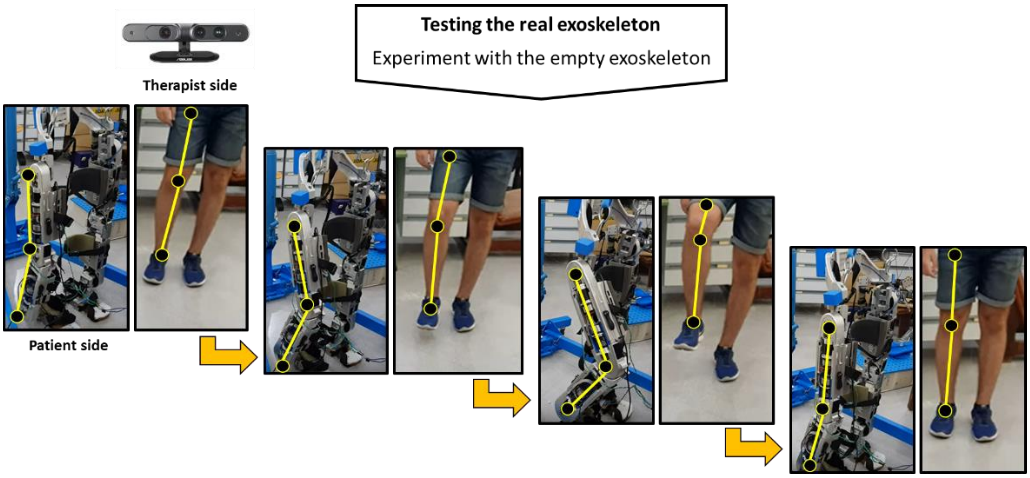

Fig. 7. Experiment to test the real exoskeleton empty. The user that acts as a therapist is in front of the RGB-D camera. A set of Position Joint Commands, which define the trajectory after executing the motion of elevating and descending the knee, are calculated from the human skeleton, detected taking the depth image captured by the camera as input. The trajectory is properly reproduced by the exoskeleton in real time.

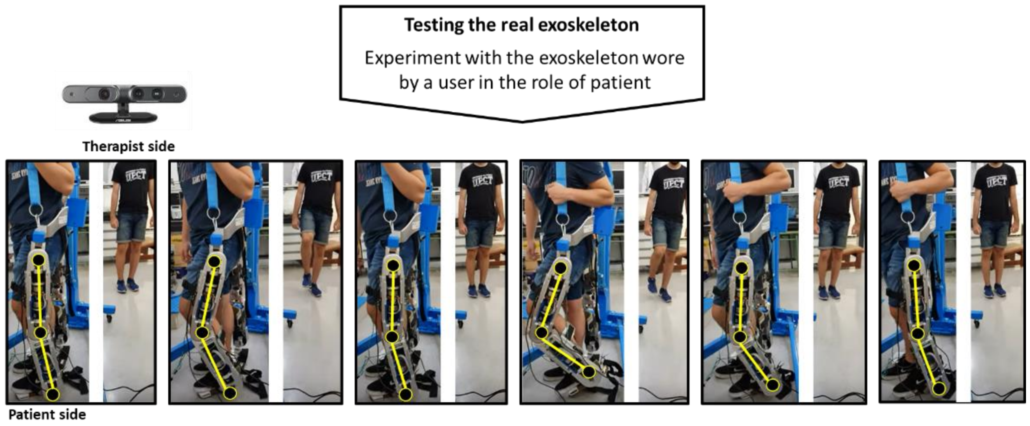

Fig. 8. Experiment to test the system when a user is wearing the exoskeleton. The user acting as a therapist has carried out several motions elevating and descending the knee. The exoskeleton reproduces the motion in real time.