

## Article

# Implementation and Assessment of an Intelligent Motor Tele-Rehabilitation Platform

Yves Rybarczyk <sup>1,2,3,\*</sup> , Jorge Luis Pérez Medina <sup>1</sup> , Louis Leconte <sup>4</sup>, Karina Jimenes <sup>1</sup>, Mario González <sup>1</sup>  and Danilo Esparza <sup>1</sup>

<sup>1</sup> Intelligent & Interactive Systems Lab (SI<sup>2</sup> Lab), Universidad de Las Américas, Quito 170125, Ecuador; jorge.perez.medina@udla.edu.ec (J.L.P.M.); karina.jimenes@udla.edu.ec (K.J.); mario.gonzalez.rodriguez@udla.edu.ec (M.G.); wilmer.esparza@udla.edu.ec (D.E.)

<sup>2</sup> Department of Electrical Engineering, CTS/UNINOVA, Nova University of Lisbon, 2829-516 Monte de Caparica, Portugal

<sup>3</sup> School of Informatics, University of Skövde, 54128 Skövde, Sweden

<sup>4</sup> Ecole Normale Supérieure, 94235 Paris-Saclay, France; louis.leconte@ens-paris-saclay.fr

\* Correspondence: y.rybarczyk@fct.unl.pt; Tel.: +351-2-1294-8545

Received: 18 October 2018; Accepted: 24 December 2018; Published: 4 January 2019



**Abstract:** Over the past few years, software applications for medical assistance, including tele-rehabilitation, have known an increasing presence in the health arena. Despite the several therapeutic and economic advantages of this new paradigm, it is important to follow certain guidelines, in order to build a safe, useful, scalable, and ergonomic tool. This work proposes to address all these points, through the case study of a physical tele-rehabilitation platform for patients after hip replacement surgery. The scalability and versatility of the system is handled by the implementation of a modular architecture. The safeness and effectiveness of the tool is ensured by an artificial intelligence module that assesses the quality of the movements performed by the user. The usability of the application is evaluated by a cognitive walkthrough method. Results show that the system (i) is able to properly assess the correctness of the human's motion through two possible methods (Dynamic Time Warping and Hidden Markov Model), and (ii) provides a good user experience. The discussion addresses (i) the advantages and disadvantages of the main approaches for a gesture recognition of therapeutic movements, and (ii) critical aspects to provide the patient with the best usability of a tele-rehabilitation platform.

**Keywords:** eHealth; software engineering; gesture recognition; Dynamic Time Warping; Hidden Markov Model; usability

## 1. Introduction

Over the past few years, alternative healthcare deliveries have been developed. For instance, the advances in the telecommunications have permitted the emergence of telehealth systems. One specialized field of telehealth is tele-rehabilitation, which allows for the implementation of a therapeutic program via an interactive multimedia web-based platform [1,2]. A remarkable increase in the number of patients treated by tele-rehabilitation has been noticed since the beginning of the 21st century, especially in physiotherapy [3]. Despite the several medical and economic advantages of this new paradigm, the development of a tele-rehabilitation platform has to follow specific rules to be safe and efficient. Two fundamental rules are (i) to build a platform based on an affordable system to capture the human movements and (ii) to make sure the patient is performing the therapeutic exercises correctly [4].

The present work proposes a web-based platform for motor tele-rehabilitation applied to patients after hip arthroplasty surgery. This orthopedic procedure is an excellent case study, because it

involves people who need a postoperative functional rehabilitation program to recover strength and joint mobility. The development of a tele-rehabilitation system is justified by the condition of these individuals that makes difficult their transportation to and from the physiotherapist's office. The proposed approach considers two fundamental conditions for the development of a suitable tele-rehabilitation platform. First, the system is based on a modular architecture composed of a low-cost motion capture device, in order to ensure the viability and scalability of the tool. Second, the platform detects automatically the correctness of the executed movement to provide the patient with real-time feedback [5]. Since exercises are carried out in front of a machine instead of a therapist, the subjects may produce incorrect movements that could be harmful, especially after a surgery. In addition, the lack of human presence diminishes the motivation, which may slow down the recovery process. Here, two possible approaches are tested to assess the movements and display an appropriate feedback: (i) Dynamic Time Warping (DTW) [6], and (ii) Hidden Markov Models (HMMs) [5].

The remainder of the manuscript is organized into six parts. Section 2 is a presentation of related work. Section 3 is a general description of the web-based platform. Section 4 focuses on a first study to assess the movements and it is based on DTW. Section 5 presents a second study on movement assessments, which uses HMMs. The reliability of both approaches to discriminate between correctly and incorrectly performed movements is tested through laboratory experiments that compare the evaluation of the computer to the evaluation of the therapists. Section 6 consists of an analysis of the usability of the system, through an experiment that involves end users. Finally, the last section (i) discusses the results of the automatic assessments and the usability tests, and (ii) draws conclusions on the most suitable solutions to develop an efficient tele-rehabilitation system.

## 2. Related Work

Different approaches are proposed to provide the patient with a tele-rehabilitation system. The motion capture can be based on inertial wearable sensors or visual sensors. [7] propose a cloud-assisted wearable system (Rehab-aaS) that enables a general motor rehabilitation, even if the application is optimized for the upper limbs. The platform is scalable and can be integrated into a body sensor network (BodyCloud) [8] for the monitoring of different physiological parameters. However, the system does not have an artificial intelligent module that allows for a rigorous assessment of the rehabilitation exercises. In addition, the wearable devices require the inertial sensors to be precisely placed on the body and/or to involve a calibration stage. Thus, another approach consists of using a vision-based motion capture. This system also presents certain limitations, such as the occlusion problem, but it has the advantage to provide an easy setup. The occlusion refers to a configuration in which a body joint disappears temporarily behind a part of the individual's body. This situation can be overcome (at least partially) by asking the individuals to change the orientation of their body according to the plane in which the movement is performed. Most of the systems use the Kinect, because it is an affordable piece of equipment and its accuracy is good enough for functional assessment activities [9]. An experiment that consists of assessing the accuracy of the Kinect by comparison to an accelerometer shows a percentage of correlation between the two sensors equal to 96% [4]. An avatar evolving in a serious game is usually used to motivate the users to regularly practice their therapeutic exercises [10]. This gamification approach is also applied for physical exercises, such as Tai Chi [11]. The application can indicate how well the player imitates the instructor, even if the system is not designed to classify gestures. Another type of Tai Chi prototype platform is built to rehabilitate patients with movement disorders [12]. The patient's movements are compared with pre-recorded movements of a coach and further evaluated by using a fuzzy logic algorithm [13]. Additionally, a component-based application framework is proposed by [14] for the development of 3D games by combining already existing 3D visual components. The preliminary results indicate that the prototypes can be used as serious game for physical therapy. However, this framework is not open source and the matching between the avatar and the user is not robust, especially when occlusions occur.

In general, it is very rare to find a system that provides both an attractive virtual environment and an algorithm to assess the quality of the rehabilitation movements executed by the patients. Instead of evaluating a spontaneous movement, other studies propose to guide the gesture through visual [15] or haptic [16] feedback, in order to avoid wrong motions. The disadvantage of these approaches is to induce a too stereotyped movement, which reduces the functional benefit of the rehabilitation. Thus, the technological implementation of an appropriate program of motor rehabilitation must involve an expert system that could substitute the therapist and provide the patient with feedback. Recent studies applied algorithms based on rules [17] or Dynamic Time Warping [18] for the recognition of therapeutic movements, and fuzzy logic [19] for the diagnosis of physical impairments. Nevertheless, the current systems are essentially able to discriminate between a correct and an incorrect gesture, but they do not give a targeted feedback on the type of error or they do not consider compensatory movements when an exercise is wrongly executed. To get such a feedback, it seems necessary to build a model of the therapeutic exercises as suggested by [20], who propose a theoretical modeling in UML for the reeducation of the upper limbs. Our work proposes a more advanced approach, since we developed and implemented a statistical model to assess the rehabilitation movements, which can be applied on any part of the body (upper and lower limbs) and can precisely identify the cause of a bad performance to provide the users of the platform with comprehensive feedback regarding the corrections to be made to the gesture.

### 3. Functionalities and Architecture of the Platform

As represented in Figure 1, the platform is composed of two main blocks dedicated to (i) the Core and (ii) the Real Time Evaluation.

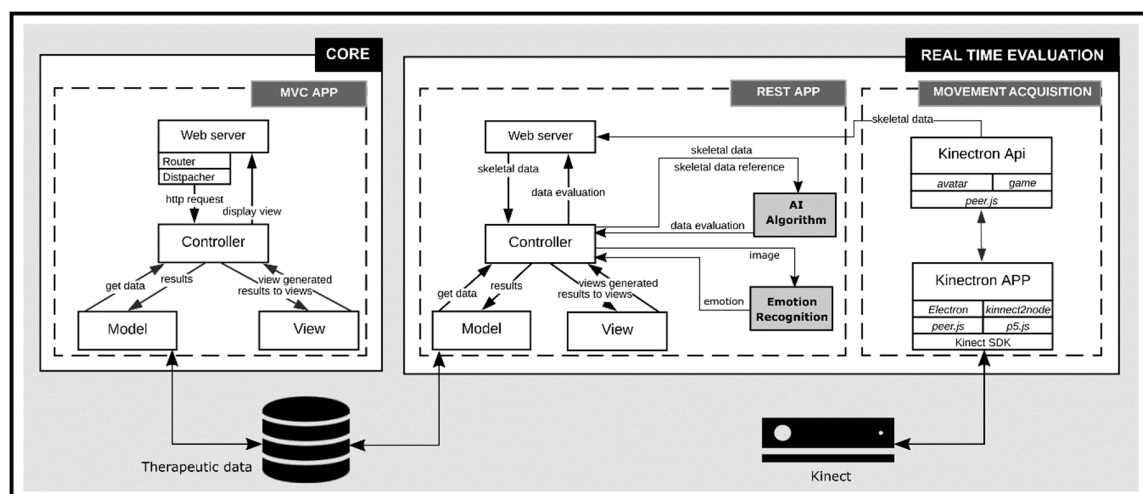


Figure 1. Architecture of the tele-rehabilitation platform.

#### 3.1. Core

The core block allows for empowering patients, who can perform autonomously a set of rehabilitation activities from a program elaborated by health professionals (e.g., physiotherapists) [21]. Several functionalities are implemented in the platform, in order to support patients in their self-recovery process. They have access to a set of multi-modal learning resources on the appropriate use of the platform and the correct procedure to complete the therapeutic program. The patient can initially visualize a list of learning resources, select one of them and display an additional list of associated multimedia files. Then, it is possible to choose the preferred learning resources among different modalities, review the information and return to the general learning menu.

In addition, this block is connected to the database that contains a set of rehabilitation exercises. The computational implementation is based on Django 2.0.5. This web framework

uses the Model-View-Controller (MVC) design pattern and works under the two central principles: (i) maintaining loose coupling between the layers of the framework, and (ii) Don't Repeat Yourself (DRY) [22]. According to this pattern, the application functionality is divided into three kinds of components. The model represents the knowledge and has the logic to update the controller. The view represents the visualization of the data. The controller determines the data flow into a model object and updates the view whenever data change. It keeps view and model separated and acts on both.

### 3.2. Real Time Evaluation

#### 3.2.1. Functionalities

The rehabilitation sessions are planned by the physiotherapist. They are composed of a set of exercises. An exercise is based on the composition pattern [23], which means that it is a composition of simple or compound exercises. Before performing the exercises, patients have to answer a questionnaire regarding their health status. The responses are processed by the application to authorize or disavow them to start a rehabilitation session. If they are authorized to initiate a session, the system activates the execution interface of the exercises. An exercise consists of a set of repetitions predefined in the recovery plan. There is a short break between each repetition to provide the patient with a feedback regarding the quality of the movement. The trials are assessed in terms of range of motion (ROM) and compensatory movements. Before starting an exercise, the patients can consult a set of instructions to execute the movements correctly. During the completion of the exercise, an avatar that maps in real time the patient's movement is displayed in a game-based virtual environment (Figure 2). The movement quality and facial expressions are also assessed in real time by artificial intelligence algorithms. Once a trial is completed, the performance is graphically presented to the user. Finally, the patients have the possibility to suspend a rehabilitation session at any time. In this case, they must provide the physiotherapist with a reason to abort the session.

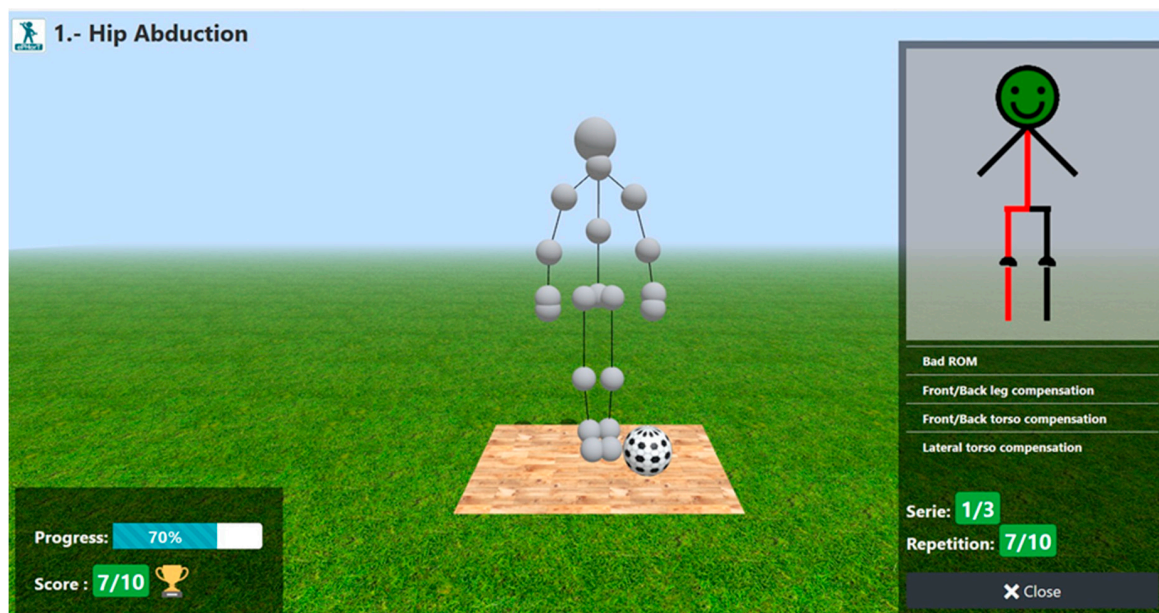


Figure 2. User's interface of the game-based exercises.

#### 3.2.2. Implementation

The real time evaluation block is composed of two modules, the first one for processing skeleton data with a RESTful API, and the second one for acquiring movement data as a client application. They communicate to each other for processing the movement data and for giving the results using JSON data format. The RESTful API was developed using Django Rest Framework 3.8.2. It is a



powerful and flexible toolkit of Django to build RESTful Web Services [24]. The client application is built with the Kinectron 1.4.2 and Processing 1.4.8. Kinectron is an open source tool to capture the movement data. It has two components: a server that broadcasts Kinect data, and an API that receives Kinect data [25]. A Kinect v2 camera is used to capture the movements of the patient and extracts the coordinates of the main body joints. A cross-browser JavaScript Library is incorporated (Three.js) to display in a web browser the game-based exercises, which are composed of 3D animated objects and the user's avatar. Additional technologies, such as Bootstrap 4.1.13 and JQuery 3.1.14 are used to build the interface. Bootstrap is a CSS framework that provides a characteristic of responsive web design. It allows developing interfaces compatible with different kinds of browsers and terminal platforms [26]. JQuery is a Javascript library which has CSS selector, flexible animation system and event system, rich plugins, and solutions for browser compatibility issues.

The API REST package works with the exercises packages and contains the intelligent modules to assess both the quality of the movements and the emotions of the patients when performing an exercise. It has a couple of services to process the skeleton and the images frames through different artificial intelligence algorithms. The first service processes the skeleton to evaluate if the patient is exercising correctly and the result is transferred as a JSON structure with a set of angles. Each angle has a type, a DTW metric that compares the performed movement to the reference movement. A graphical-based feedback is displayed on the user's interface to inform the patients on the quality of their performance (Figure 2). A red color represents bad movement amplitude for a determined limb or body part. This visual feedback is reinforced by a textual description of potential errors in the motion. Other information, such as a progress bar and a score are also displayed. The second service processes the image frame every second to determine if the patient feels pain. A machine-learning algorithm based on Support Vector Machine (SVM) is implemented to operate a Boolean classification of facial expressions in pain (1) vs. no pain (0). A sequence of 1 and 0 is obtained at the end of each repetition. This information is used by the therapist to identify if the intensity of the exercises is appropriate or not for a determined patient.

#### 4. Movement Assessment: DTW Approach

##### 4.1. Dynamic Time Warping

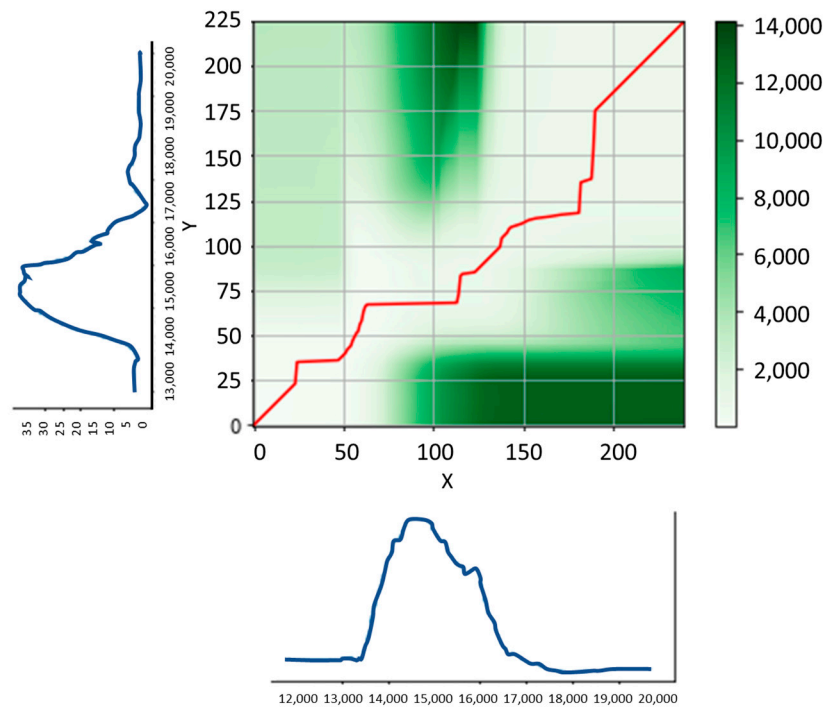
Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two temporal sequences, which may vary in speed. It is perfectly adapted for the assessment of therapeutic movements for which the velocity of the execution is not relevant. This characteristic explains why we have chosen this technique, because it enables us to assess the quality of the movement by a single comparison to a reference. Any distance (Euclidean, Manhattan, ...) which aligns the  $i$ -th point on one-time series with the  $i$ -th point of another produces a poor similarity score. On the contrary, a non-linear alignment produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis. The implementation of such an algorithm is obtained through a matrix sized with the length of each signal (Figure 3). On each point of the matrix, the distance between the points associated with each of the two signals is calculated. The best alignment of two signals is given by the path through the grid that minimizes the total distance between these two signals. Our algorithm implements a quadratic distance ( $D$ ), as defined in Equation (1).

$$D_{(i,j)} = (x_{(j)} - y_{(i)})^2 \quad (1)$$

Thus, a distance matrix is obtained. Then, a cumulated distance matrix ( $C$ ) is created, based on the distance matrix ( $D$ ). The latter is built thanks to a definition of distance very close to the 8-connectivity in a square tiling (mathematic topology) for each point of the matrix, as presented in Equation (2).

$$C_{(i,j)} = \min\{C_{(i-1,j-1)}, C_{(i-1,j)}, C_{(i,j-1)}\} + D_{(i,j)} \quad (2)$$

Finally, we have to look for the path that minimizes the cumulated distance. To do so, we start from the last point and look for the point with the lower cumulated distance between all its three inferior neighbors (within the meaning of Moore neighborhood). We reiterate the process until we reach a side or the originate point  $C(0, 0)$ . Then, we just have to calculate the length of this path. The result is small when signals are close, and becomes larger when signals differ. An optimization can be provided to this algorithm, using a warping window. It is based on the fact that a good alignment path is unlikely to wander too far from the diagonal and, thus, limits the window length. Since the Kinect frequency is about 33 Hz and the movements are short, we do not need to implement this optimization, because only few points (between 100 and 200) have to be analyzed.



**Figure 3.** Cumulated distance matrix and its optimal path (in red), between a reference movement and a movement to be tested (blue signals).

#### 4.2. Trigonometric Parametrization

As we expect our algorithm to be invariant to 3D translational motion and to different bodies, we have to focus not on coordinates of each joint, but on angles. This section describes the mathematical calculation of the (i) working angles (ROM) and (ii) compensation angles for the four main therapeutic movements: hip abduction, slow flexion hip and knee, hip extension, and forward-sideway-backward sequence. During a rehabilitation session it is important to identify both the movement amplitude the patient should reach (ROM) and the compensatory movements that should be avoided for an appropriate re-education of the injured limbs.

For hip abduction, we identified one working angle (Figure 4, picture on the left) and three compensation angles (Figure 4, three pictures on the right). To obtain the angle of the hip abduction ( $\theta_{aw}$ ), we have to project  $\overrightarrow{16_{18}}$  on the frontal plane, and proceed with the calculation of Equations (3)–(5).

$$n_1 = \left[ \sum_{k=x,y,z} (k_{18} - k_{16})(k_{16} - k_0) \right]^2 \quad (3)$$

$$d_1 = (x_{16} - x_0)^2 + (y_{16} - y_0)^2 + (z_{16} - z_0)^2 \quad (4)$$

$$\cos \theta_{aw} = \frac{|y_{16} - y_{18}|}{\sqrt{\frac{n_1}{d_1} + (y_{16} - y_{18})^2}} \quad (5)$$

where  $n_1$  and  $d_1$  are intermediate variables used to calculate the angle  $\theta_{aw}$ , and  $k$  is successively equal to  $x$ ,  $y$ , and  $z$  in the calculation of the sum. '0' stands for spine base, '16' stands for left hip, and '18' stands for left ankle. The leg compensation angle ( $\theta_{lc}$ ) is obtained by the calculations of Equations (6)–(8).

$$n_2 = \left[ \sum_{k=x,z} (k_{18} - k_{16})(k_{15} - k_{14}) \right]^2 \quad (6)$$

$$d_2 = (x_{15} - x_{14})^2 + (z_{15} - z_{14})^2 \quad (7)$$

$$\cos \theta_{lc} = \frac{|y_{16} - y_{18}|}{\sqrt{\frac{n_2}{d_2} + (y_{16} - y_{18})^2}} \quad (8)$$

where  $n_2$  and  $d_2$  are intermediate variables used to calculate the angle  $\theta_{lc}$ . '14' stands for right ankle, and '15' stands for left foot. Equations (9)–(11) are used to calculate the frontal torso compensation angle ( $\theta_{ftc}$ ).

$$n_3 = \left[ \sum_{k=x,z} (k_0 - k_{20})(k_{15} - k_{14}) \right]^2 \quad (9)$$

$$d_3 = (x_{15} - x_{14})^2 + (z_{15} - z_{14})^2 \quad (10)$$

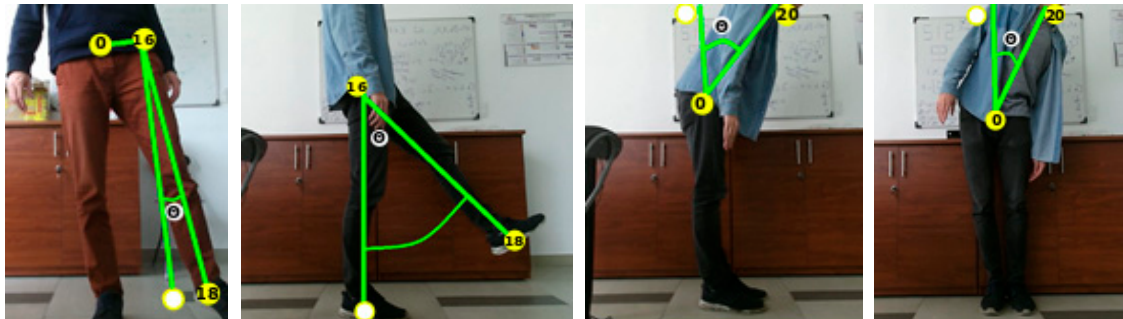
$$\cos \theta_{ftc} = \frac{|y_0 - y_{20}|}{\sqrt{\frac{n_3}{d_3} + (y_0 - y_{20})^2}} \quad (11)$$

where  $n_3$  and  $d_3$  are intermediate variables used to calculate the angle  $\theta_{ftc}$ . '20' stands for spine shoulder. Equations (12)–(14) are used to calculate the lateral torso compensation angle ( $\theta_{ltc}$ ).

$$n_4 = \left[ \sum_{k=x,y,z} (k_{20} - k_0)(k_{16} - k_0) \right]^2 \quad (12)$$

$$d_4 = (x_{16} - x_0)^2 + (y_{16} - y_0)^2 + (z_{16} - z_0)^2 \quad (13)$$

$$\cos \theta_{ltc} = \frac{|y_0 - y_{20}|}{\sqrt{\frac{n_4}{d_4} + (y_0 - y_{20})^2}} \quad (14)$$



**Figure 4.** Expected and possible movements involved in hip abduction (from left to right): working angle, leg compensation, front torso compensation, and lateral torso compensation.

For slow flexion of hip and knee, we identified two working angles (Figure 5, picture on the left) and two typical compensation angles (Figure 5, two pictures on the right). Equations (15)–(17) are used to calculate the hip-working angle ( $\theta_{hw}$ ).

$$n_5 = \left[ \sum_{k=x,y,z} (k_{17} - k_{16})(k_{15} - k_{14}) \right]^2 \quad (15)$$

$$d_5 = (x_{15} - x_{14})^2 + (y_{15} - y_{14})^2 + (z_{15} - z_{14})^2 \quad (16)$$

$$\cos \theta_{hw} = \frac{|y_{17} - y_{16}|}{\sqrt{\frac{n_5}{d_5} + (y_{17} - y_{16})^2}} \quad (17)$$

where  $n_5$  and  $d_5$  are intermediate variables used to calculate the angle  $\theta_{hw}$ . ‘14’ stands for right ankle, ‘15’ stands for left foot, ‘16’ stands for right hip, and ‘17’ stands for left knee. Equations (18)–(20) are used to calculate the knee-working angle ( $\theta_{kw}$ ).

$$n_6 = \left[ \sum_{k=x,y,z} (k_{18} - k_{17})(k_{15} - k_{14}) \right]^2 \quad (18)$$

$$d_6 = (x_{15} - x_{14})^2 + (y_{15} - y_{14})^2 + (z_{15} - z_{14})^2 \quad (19)$$

$$\cos \theta_{kw} = \frac{|y_{18} - y_{17}|}{\sqrt{\frac{n_6}{d_6} + (y_{18} - y_{17})^2}} \quad (20)$$

where  $n_6$  and  $d_6$  are intermediate variables used to calculate the angle  $\theta_{kw}$ . ‘14’ stands for right ankle, ‘15’ stands for left foot, ‘17’ stands for left knee, and ‘18’ stands for left ankle. Equations (21)–(23) are used to calculate the thigh compensation angle ( $\theta_{thc}$ ).

$$n_7 = \left[ \sum_{k=x,y,z} (k_{17} - k_{16})(k_{12} - k_{16}) \right]^2 \quad (21)$$

$$d_7 = (x_{12} - x_{16})^2 + (y_{12} - y_{16})^2 + (z_{12} - z_{16})^2 \quad (22)$$

$$\cos \theta_{thc} = \frac{|y_{17} - y_{16}|}{\sqrt{\frac{n_7}{d_7} + (y_{17} - y_{16})^2}} \quad (23)$$

where  $n_7$  and  $d_7$  are intermediate variables used to calculate the angle  $\theta_{thc}$ . ‘12’ stands for left hip, ‘16’ stands for right hip, and ‘17’ stands for left knee. Equations (24)–(26) are used to calculate the tibia compensation angle ( $\theta_{tic}$ ).

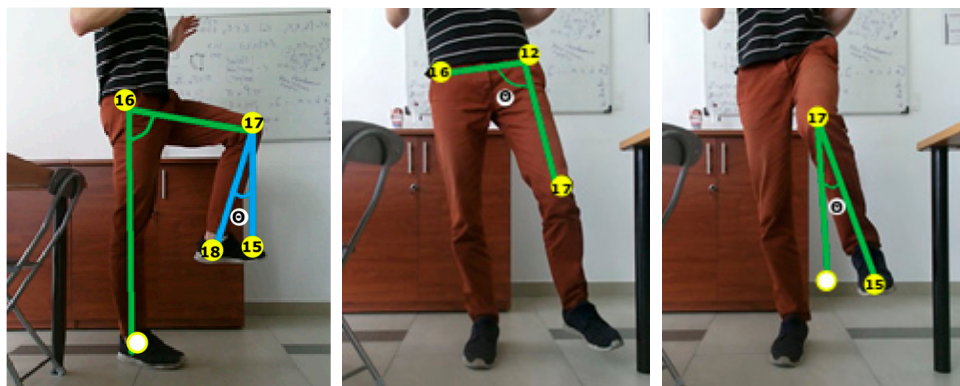
$$n_8 = \left[ \sum_{k=x,y,z} (k_{18} - k_{17})(k_{12} - k_{16}) \right]^2 \quad (24)$$

$$d_8 = (x_{12} - x_{16})^2 + (y_{12} - y_{16})^2 + (z_{12} - z_{16})^2 \quad (25)$$

$$\cos \theta_{tic} = \frac{|y_{18} - y_{17}|}{\sqrt{\frac{n_8}{d_8} + (y_{18} - y_{17})^2}} \quad (26)$$

where  $n_8$  and  $d_8$  are intermediate variables used to calculate the angle  $\theta_{tic}$ . ‘12’ stands for left hip, ‘16’ stands for right hip, ‘17’ stands for left knee, and ‘18’ stands for left ankle. The same frontal and lateral compensation movements of the trunk as described in hip abduction are also calculated for this exercise (Equations (9)–(14)).





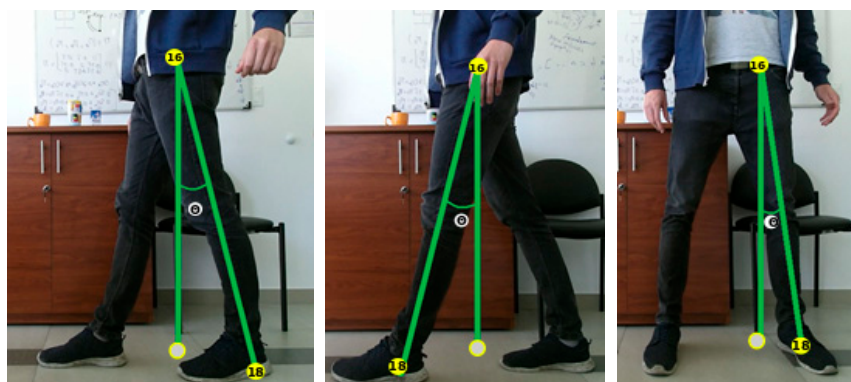
**Figure 5.** Expected and possible movements involved in slow flexion of hip and knee (from left to right): working angle for hip (in green) and knee (in blue), thigh compensation, and tibia compensation.

For hip extension, the working angle is the same as the compensation angle of hip abduction, and the compensation angle is equal to the working angle of hip abduction (Figure 6). The same frontal and lateral compensation movements of the trunk as described in hip abduction are also calculated for this exercise (Equations (9)–(14)).



**Figure 6.** Working angle for hip extension.

There are two working angles for the forward-sideway-backward sequence, which are the working and leg compensation angles of the hip abduction (Figure 7). Again, the frontal and lateral compensation movements of the trunk described in hip abduction are considered for this exercise (Equations (9)–(14)).



**Figure 7.** Working angle for forward-sideway-backward sequence.

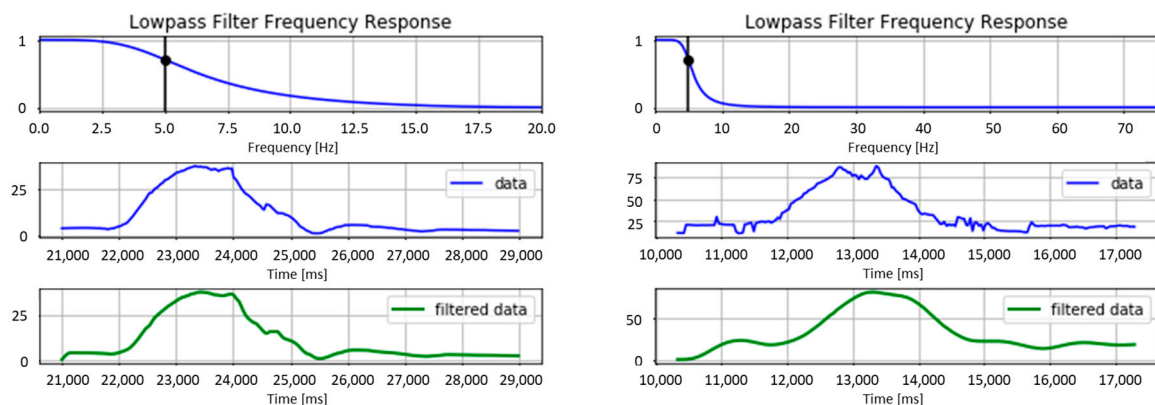
### 4.3. Filtering and Implementation

Since the raw input data provided by the Kinect are too noisy, a low pass filter is applied to the signal (Figure 8). The Butterworth Filter available in the python library ‘scipy.signal’ is used to smooth the signal. It is necessary to create two different filters according to the noisiness of the data (a moderate and a strong one). The best ratio signal/noise for the moderate filter is obtained empirically with the parameters as follows:

- filter order = 2
- sample rate (Hz) = 40
- cutoff frequency (Hz) = 5

The settings of the strong filter are:

- filter order = 4
- sample rate (Hz) = 150
- cutoff frequency (Hz) = 5



**Figure 8.** Examples of the filtering effect for the moderate (**left panel**) and the strong (**right panel**) filter.

The assessment module is implemented to compare the DTW cost of each described angle (ROM and compensations) to an empirically defined threshold (the optimum value is defined in the next section). If the distance calculated by the DTW is higher than the threshold, the angle of the movement is classified as wrong. Since different DTW run in parallel (one for each specific working and compensation movements), it is possible to provide the patients with a detailed feedback of their errors for helping them to correct their performance in the next trial. Whatever the correctness of the movement, a graphical message is always displayed to the users at the end of the trials (see Figure 2). Once a session is finished, a JSON file is created with every angle data cost, pictographic feedback (good, bad, or neutral smileys), and six possible messages that are displayed on the patient’s interface (evaluation of the ROM and potential compensation movements). Then, these data are stored in the platform database.

### 4.4. Validation

#### 4.4.1. Experimental Protocol

The experiment consisted of comparing the assessment performed by four physiotherapists to the assessment made by the algorithm. Seven healthy subjects participated in the study. They were informed about the purpose of the study and signed a consent form to take part to the experiment. They were asked to take place at approximately two meters distance from a single Kinect camera. The motion capture device was placed at the height of the subject’s xiphoid apophysis. Each of the four movements

described in Section 4.2 was repeated eleven times (one of them was used as reference). The participant had to introduce a different error in the execution of the movement at each repetition (e.g., wrong range of motion or different kinds of compensation). To do so, they followed a script unknown to the therapists and different from one participant to another. The only similarity between the scripts is the fact that they were composed of six accurate executions, two movements with an incorrect ROM, and three movements with compensation errors. For each trial, the physiotherapists were asked to evaluate all the angles (ROM and compensations) on a 4-point Likert scale (0–3). The data were processed with the python library *Pandas*. Due to the inter-individual differences between the assessments made by the physiotherapists, the scale of the marks was reduced to two levels. All the assessments of 0 and 1 were grouped in a single category that corresponded to a bad execution of the exercise. And the assessments of 2 and 3 were all classified as a good movement. The final therapist's evaluation was obtained by averaging the score of the four professionals. In case of draw, the trial was discarded. Then, these Boolean variables (bad vs. good) were compared with the assessment performed by the algorithm. The outcome was expected to be different depending on the decision threshold.

#### 4.4.2. Results

Overall, the outcome of the algorithm matches at 88% with the assessment made by the physiotherapists. The accuracy of this matching depends on the threshold of tolerated difference between the reference movement and the analyzed movement (Figure 9). A certain variability in the performance of the classification is observed from one exercise to another. For instance, the best results are obtained for the assessment of the ROM for hip abduction and forward-sideway-backward sequence (percentage of accuracy >90%).

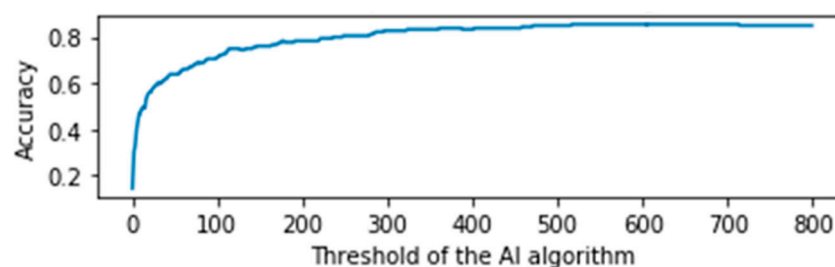


Figure 9. Overall variation of accuracy depending on the threshold.

A confusion matrix permits to identify the percentage of movements assessed as correct by the therapists and predicted as correct by the algorithm (True Positives—TP), the incorrect movements predicted as incorrect (True Negatives—TN), the correct movements predicted as incorrect (False Negatives—FN), and the incorrect movements predicted as correct (False Positives—FP). The accuracy of the algorithm to predict the TP and TN is almost the same, even if it is slight higher for the former than the latter (Table 1). This small difference can be explained by the fact that there is more diversity in the execution of the bad (errors of ROM and compensations) than the good movements.

Table 1. Confusion matrix providing a detailed comparison between the movement assessments of the physiotherapists and the algorithm.

|                      |          | therapists assessment |          |
|----------------------|----------|-----------------------|----------|
|                      |          | good mvt              | bad mvt  |
| algorithm prediction | good mvt | TP = 91%              | FP = 9%  |
|                      | bad mvt  | FN = 15%              | TN = 85% |

The misclassified movements (FP and FN) can be explained by two main reasons. First, there is only a partial consensus between the assessments carried out by the four therapists. Nevertheless, the discrepancy between the health professionals is less than 8% and, consequently, the human judgment is considered as an acceptable reference to evaluate the algorithm. Second, some angles suffer from occultation and cannot be evaluated correctly by the computer. This situation happens occasionally when a body joint (e.g., the hip) disappears temporarily behind a part of the participant's body (e.g., the hand).

Table 2 provides a quantification of the missing data in terms of the percentage of sampling errors and occultations. Sampling errors are defined as technical issues caused by faults in the capture sampling rate of the Kinect. The occultations refer to situations in which the participants cross their arms, cross their arms in front of the torso or cross the legs while performing the exercises. They are computed considering the pose arrangement [27] that identifies intersections between the torso, the arms, and/or the legs. The results show that the sampling errors are almost insignificant (<0.01%) and the occultations represent less than 6% of the missing data. Some movements are more affected by occultation than others, which suggests defining an optimal orientation of the body with respect to the vision-based motion capture sensor, for each exercise.

**Table 2.** Missing data defined in terms of the percentage of sampling errors and joint occultations, for each exercise (slow flexion hip and knee—SFHK; hip abduction—HA; hip extension—HE; forward-sideway-backward—FSB).

| Exercises       | Sampling Error (%) | Occultation (%) | Total (%)    |
|-----------------|--------------------|-----------------|--------------|
| SFHK            | 0.01               | 14.35           | <b>14.36</b> |
| HA              | 0.00               | 0.26            | <b>0.26</b>  |
| HE              | 0.01               | 1.17            | <b>1.18</b>  |
| FSB             | 0.00               | 7.61            | <b>7.61</b>  |
| <b>Mean (%)</b> | <b>0.005</b>       | <b>5.85</b>     | <b>5.855</b> |

## 5. Movement Assessment: HMM Approach

### 5.1. Hidden Markov Model

Hidden Markov Model is a probabilistic approach that aims to model a given action into hidden states. It is composed of initial, transitional and emission probabilities. Initial probabilities are the distribution of probabilities of 'being in a state' before a sequence is observed. Transitional probabilities are represented by a matrix, in which the probabilities indicate the possible changes from one state to another. Finally, the emission probabilities model the variance of each state's associated values (mostly Gaussian Probability Density Functions—PDFs) obtained from continuous variable observations. These model parameters can be learned with the use of the Expectation-Maximization (EM) algorithm. Signals can be classified by looking at the probability that a signal is generated from a trained HMM. This is done by the use of the forward algorithm.

Learning the model parameters (states and transitions) by optimizing the likelihood is essential to make meaningful use of the HMM in classification. The distribution function defined by a Gaussian, Mixed Gaussian or multinomial density function, as well as the covariance type, needs to be characterized prior to this process. An observation is merely a noisy and variable representation of a related state. A state is a clustering of observations that relates to a distribution with a specific mean in the parameter space. A likely state is retrieved by finding the cluster that the observation is member of. Also, the transition probabilities between states creates a sequence of the most likely temporal succession of states. Estimating the model parameters is done by utilizing the Baum-Welch Expectation-Maximization algorithm, which is based on a forward-backward algorithm used in classifying Hidden Markov Chains [28,29]. The probabilities are calculated at any point of a sequence by inspecting previous observations, to find out how well the model describes the data, and following observations, to conclude how well the model predicts the rest of the sequence. This is an iterative

process, in which the objective is to find an optimal solution (state sequence) for the HMM. This optimal sequence of states is inferred using the Viterbi algorithm. In addition, the forward algorithm can be used to calculate the probability that a sequence is generated by a specific trained HMM, making it applicable for classification.

This classification is based on training an individual HMM per subclass of an exercise. For instance, one HMM could be trained on ‘running’ while another one would be trained on ‘walking’ (both subclasses of the human locomotion class). When calculating the forward probabilities of a sequence of observations and comparing the probabilities of all the HMMs, the sequence is classified as the category that provides the highest probability, as described in Equation (27).

$$\text{Class} = \arg_{i=1}^n \max [\Pr(O|\lambda_i) * \Pr(\lambda_i)] \quad (27)$$

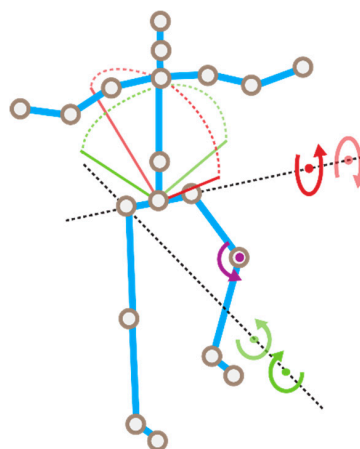
where  $\lambda_i$  represents a determined model and  $O$  is a sequence of observations. The amount of states in a HMM is a free parameter. The Bayesian Information Criteria (BIC) is a technique that aids to define a determined number of parameters by taking into account the possibility of overfitting the data when the number of states increases. BIC penalizes HMMs that have a high number of states, as described in Equation (28).

$$\text{BIC} = \ln(n)s - 2 \ln(\text{MLE}) \quad (28)$$

where  $n$  is the data size and  $s$  the amount of states. Therefore, the optimal amount of states is retrieved by selecting the model with the lowest BIC score. HMMs trained with multiple states are evaluated by cross-validation on their Maximum Likelihood Estimation (MLE) and the previously mentioned penalizing term.

## 5.2. Gesture Representation

A skeletonized 3D image from a Kinect camera provides Cartesian  $x$ ,  $y$  and  $z$  coordinates of twenty body joints. The gesture representation is chosen to be a skeletonized image as this has been shown to improve the model accuracy [30]. This representation depends on the position of the subject in relation to the camera and the roll, yaw and pitch angles of the device. The causal relationships between different joints are not captured by this representation. This means that physical constraints, such as a movement of the ankle that could be influenced by bending the knee, are not accounted for. To overcome these limitations, the joints are used to create a new representation that contains angles of multiple joints in respect to the frontal and sagittal planes, as well as multiple angles between relevant limbs. Figure 10 shows a graphical representation of the features in relation to the skeleton image. Table 3 describes the feature vector of the joint movements according to the anatomical terminology.



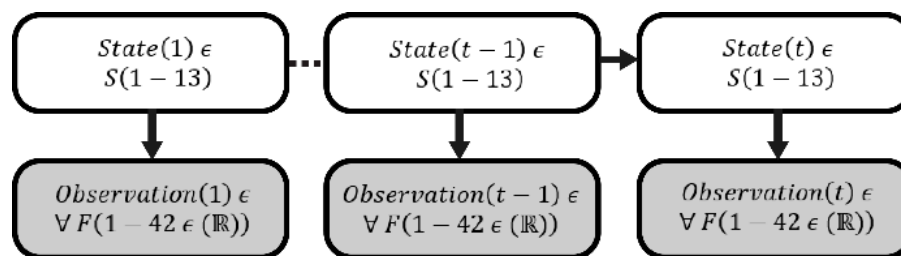
**Figure 10.** Graphical representation of the features used in the study. The movements in the egocentric frontal plane and sagittal plane are represented in green and red, respectively. The purple arrow represents the angle of the knee (independent from any plane).



**Table 3.** Feature vector describing the joint movements. For the purpose of the assessment these features are also transformed into speed and acceleration.

| Right Hip   | Left Hip   | Spine Center  | Knee                    |
|---|--|---|-------------------------|
| Right hip<br>Frontal plane rotation<br>(abduction)  | Left hip<br>Frontal plane rotation<br>(abduction)  | Spine center<br>Frontal plane rotation<br>(lateral left)  | Right knee<br>(flexion) |
| Right hip<br>Frontal plane rotation<br>(adduction)  | Left hip<br>Frontal plane rotation<br>(adduction)  | Spine center<br>Frontal plane rotation<br>(lateral right) | Left knee<br>(flexion)  |
| Right hip<br>Sagittal plane rotation<br>(flexion)   | Left hip<br>Sagittal plane rotation<br>(flexion)   | Spine center<br>Sagittal plane rotation<br>(flexion)      |                         |
| Right hip<br>Sagittal plane rotation<br>(extension) | Left hip<br>Sagittal plane rotation<br>(extension) | Spine center<br>Sagittal plane rotation<br>(extension)    |                         |

In this study, the motion is defined from the following joints: ankles, knees, hips, and spine. The angles of the knees are obtained by calculating the angle between ankle, knee and hip. The orientation of the knees induced by hip activity are expressed in four angular representations, following the two opposite directions for both sagittal and frontal planes. The same method is applied to describe the orientation of the torso, by finding the displacement of the center between the two shoulders in relation to the hips. This leads to a description of the movement into fourteen features. It is the principal representation followed by a first order and second order derivatives of these features that provide speed and acceleration of the movement. Overall, a total amount of 42 features is used. Figure 11 represents a diagram of the HMM's implemented in this study, in which  $State_{(t)}$  and  $Observation_{(t)}$  are state id and associated feature values at t time, respectively.

**Figure 11.** Graphical representation of the HMM for the exercise assessment.  $\epsilon$  stands for: takes value out of; and  $\forall$  stands for: out of all.  $F$  and  $S$  are the collection of features (42) and states (13), respectively. The definition of the optimal number of states is explained in Section 5.3.4. Each state is dependent on its previous state and observations are samples of the associated current state.

### 5.3. Experiment

#### 5.3.1. Protocol

Four healthy subjects participated in the experiment. They were informed about the purpose of the study and signed a consent form to take part in the experiment. They were asked to stand at approximately two meters distance front of a Kinect camera. The motion capture device was placed at the height of the subject's xiphoid apophysis. Each participant executed 70 movements leading to a total of 280 records. The rehabilitation exercise was a sequence, in which the subjects had to do one step forward, one step sideways and one step backward, with variations. These variations are staged executions of errors or compensatory movements that can occur during the rehabilitation in practice. The exercise was performed in batches of ten and the experiment was divided into two independent parts. In the first part, the subjects had to execute the movements as follows: (I) correct

execution, (II) steps too short, (III) execution without moving the center of mass, (IV) steps too large, (V) steps with bended knee, (VI) steps with bended knee and flexed torso. The objective of this first experimental part was to assess the accuracy of the models to classify the movements. In the second part, the subjects had to perform ten trials of a partially wrong executions of the exercise (VII), in which the faults II to VI are only occurring in the beginning, middle or end of the sequence. This second experiment is used to evaluate the real-time applicability of the HMM technique.

An application was created to capture the skeletonized image of the subjects performing an exercise. Python 2.7 was used to create a graphical user interface with the option to name, start and stop a recording. In addition, Python was used for the later processing steps, which were feature transformations and classifications. The application communicated with the Kinect SDK and wrote the data into a CSV file, with a frequency of 60 Hz, during the recording mode. The developed programs and raw data are available at <http://docentes.fct.unl.pt/y-rybarczyk/files/programs.rar> and <http://docentes.fct.unl.pt/y-rybarczyk/files/data.rar>, respectively. The HMMLearn package for python 2.7 was used for the training and application of the HMMs (<https://github.com/hmmlearn/hmmlearn>).

### 5.3.2. Evaluation Method

Using the BIC score to select the appropriate amount of states was done for each type of trained HMMs (I–VI). It provided insight on the semantic variation within the exercises. For instance, less states assigned to a faulty movement relative to the good execution implies that ‘there was something missing in the execution’, whereas the detection of extra states implied that ‘there was something added to the movement’.

For each type of execution (I–VI) an HMM was trained, leading to a total of six distinct HMMs. In order to build a general model that could assess the movements of any subject, models that classified the executions of a subject were exclusively trained on the recordings of the other remaining subjects. This led to a total of 24 trained models (6 per subject). The initial model parameters were set with a Gaussian density function and a full covariance matrix type (the initializations of the model parameters were done randomly, HMMLearn standard initialization was used, and EM iterated a 1000 times unless log-likelihood gain was less than 0.01). The HMM topological structure was fully connected, because priori knowledge about the expected state sequence could not be estimated with sufficient certainty. In addition, as the outcome of six classifiers determined the most likely model that was associated with the sequence, unpredicted variance in a signal (or noise) should not drastically influence the likelihood of the signal. The outcome of the six classifiers was calculated by means of the forward probabilities. Then, these probabilities were ranked from 1 to 6, where 1 and 6 were assigned to the highest and the lowest probability, respectively. A confusion matrix was used to map these values in terms of average prediction rank of each type of execution. In addition, an indication of the similarity of a type of execution in relation to the combination of all the other executions was provided. Finally, a range of sliding temporal windows was used to evaluate the real-time suitability of the approach. It was applied to assess the correct detection of the present types of faults in executions VII. These windows classified a subsequence in a fixed number of samples, which partially overlapped over time.

### 5.3.3. Validation Method

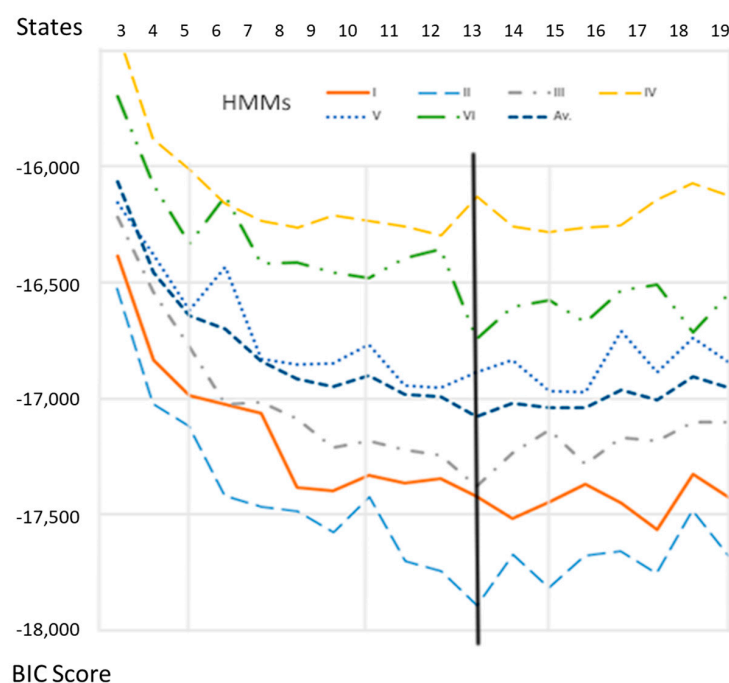
To get a reliable result that validated the models, it was important that the test data were different from the training data. Both training and test sets had to be produced by independent sampling from an infinite population, which avoided a misleading result that would not reflect what it was expected when the classifier is deployed. This section describes the used method that enabled us to apply this mandatory rule in machine learning. A 10 times repeated random sub-sampling, Monte Carlo Cross-Validation (MCCV) was used to evaluate the performance of each model (6 per subject and 24 in total). Results with Gaussian mixtures on real and simulated data suggested that MCCV provided genuine insight into cluster structure [31]. This method selects the most appropriate model for classification. To assure the ability of the model to generalize well, the validation was executed by

applying, for each subject, the other three subject's recordings. This means that each trained HMM was used as classifier of the data of an unrelated subject. Each fold contained a trial of the three subjects. The split (80% train, 20% validation) was newly created during every validation (10 times) with a random assignment of the trials in the training and test sets. This led to a model trained with 24 exercises (8 of each subject). To perform the random assignment, the python built-in random function that implements the Mersenne Twister regenerator method was used. During each validation, 6 HMMs (models I–VI) were trained such as the best performing (based on the validation score) set of HMMs was selected as models set for classification. Forward probabilities were calculated for each HMM. When the correct HMM output the highest probability, the classification value became 1 and contrary 0. Per fold, each HMM classified the remaining 6 exercises where the performance per fold was the fraction correctly classified exercises (sum of classification results) of the total classifications (36) of the 6 HMMs combined. The model's parameters differed slightly between the sets as the random data selection altered the learned state Probability Density Functions (PDFs) per fold. The best performing model set out of the 10 validations was then selected to perform the classification for the test subject.

### 5.3.4. Results

#### 1. State assignment and classification

The MLE for each HMM up to twenty states was used to define the BIC scores (see Equation (28)) against the amount of states (Figure 12). The profile of the BIC score against the amount of states was similar between the HMMs. The consensual lowest BIC score was obtained for an amount of states equal to thirteen (value that corresponds to the minimum of the average curve—the blue bold broken line in Figure 12). Thus, thirteen states were used to model the six movements. This makes intuitively sense as the exercise was constructed out of three distinctive parts (a multiple of three is expected), plus an initial/ending part (inactive state). Hence, each part in the exercise was described by four states.



**Figure 12.** BIC scores for each type of execution (I–VI) and an averaged BIC score over these executions (blue bold broken line). The black vertical line indicates the optimal amount of states.

The classification performance shows a high level of accuracy (Table 4) in classifying a whole sequence into the classes (I–VI). A value of 1 means the model always gave the highest probability,

with respect to the other models and for any sequence of the related movement, whereas a value of 6 indicates the lowest probability. The values in this table are averaged prediction ranks for each model of each movement (I–VI). The average prediction rank of HMM I is the highest (2.78), which means that the execution type I (correct movement) is most closely related to all the other types. The overall performance of the classification for each class (I–VI) is shown in Table 5. It is to note that the execution type III (i) is more likely to be classified as type I, and (ii) has the lowest prediction accuracy compared with the other classes. Movement III is the one that presents the subtlest difference from movement I. The only distinction between these two executions is the fact that the participants do not move their center of mass in the former case. In other words, the trunk does not go with the rest of the movement. This fine difference could explain the limitation of the algorithm to discriminate between the two types of movement. A solution to overcome this issue could be to implement a better staging of the execution III and/or getting a higher descriptive power in the gesture representation (e.g., including additional features related to the upper-part of the body).

**Table 4.** Confusion matrix of executions (I–VI). Each column represents the types of movement and each row the output prediction ranks of the HMMs (I–VI). The closer is the value to 1 (green cells) the better is the prediction.

|             |     | Movements    |      |      |      |   |    |      |
|-------------|-----|--------------|------|------|------|---|----|------|
|             |     | I            | II   | III  | IV   | V | VI |      |
| Predictions | I   | 1            | 2.27 | 1.4  | 3.97 | 4 | 4  | 2.78 |
|             | II  | 2.7          | 1    | 2.8  | 6    | 6 | 5  | 3.92 |
|             | III | 2.3          | 2.74 | 1.57 | 5    | 5 | 6  | 3.77 |
|             | IV  | 4.74         | 4    | 4.9  | 1.04 | 2 | 3  | 3.28 |
|             | V   | 4.27         | 5    | 4.34 | 1.97 | 1 | 2  | 3.1  |
|             | VI  | 6            | 6    | 6    | 3.04 | 3 | 1  | 4.18 |
|             |     | Average rank |      |      |      |   |    |      |

**Table 5.** Performance of the classification of movements I–VI.

| I    | II   | III | IV  | V    | VI   |
|------|------|-----|-----|------|------|
| 100% | 100% | 57% | 97% | 100% | 100% |

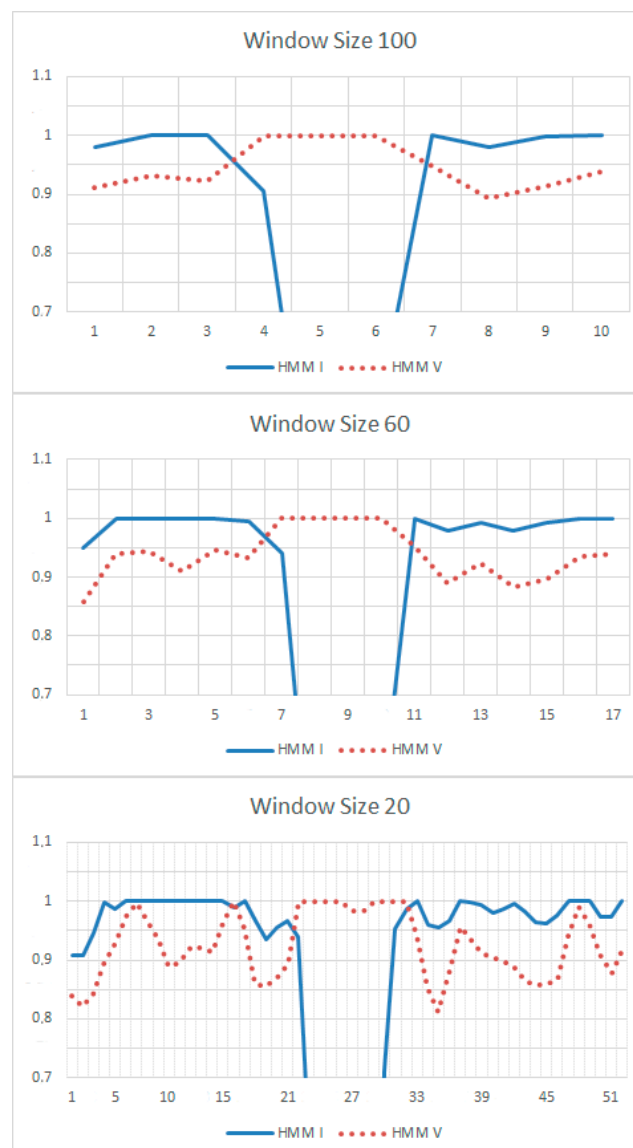
## 2. Real-time testing

The aim for the platform is not only to provide an overall classification of the movement (correct vs. types of fault), but also to identify a potential real-time transition from one classification to another during a single exercise. This can aid the patient to be aware on the phases of the movement in which certain errors tend to occur. The result of this instantaneous classification could be displayed as a real-time feedback when the patient is executing any exercise.

The samples of execution type VII are used to evaluate the ability for the trained models I–VI to classify partially incorrect movement (i.e., only certain phases of the movement are wrong). This execution is a rehabilitation movement in which the individual performs one step to the front, one step to the side and one step to the back, in a continuous sequence. The classification takes place over a selection of frames within the movement. The forward algorithm, as described previously, is used to perform this classification. Three different sizes of windows are used for the classification: 100, 60 and 20 frames. These different samplings are made to study the effect of the window size on the consistency

and accuracy of the assessment. After classifying the frames of a determined window size, the window shifts half the number of frames in the total sequence and the classification is repeated until the end of the sequence is reached. This so-called overlapping window is used to obtain a smoother classification path over time. There are multiple classification values during the full exercise. At each newly created classification moment in the exercise the values of the six classifiers are normalized in a fashion that the highest value becomes 1 and the other values are expressed as a fraction of this value. Detection is considered accurate if the majority of the movement's phase where the error occurred assigns the value of 1 to the expected error type. In the case of the execution type VII, there are three phases: step forward, step sideways, and step backward.

Figure 13 shows that whatever the window size is the performance maintains quite accurate, even though the result is a little bit noisier when the sizes get smaller. For instance, there is a very high detection rate (21/24) when errors of types IV to VI are present in the sequence of the movements, for any sampling size. Detecting execution types II and III are less successful (9/16). It can be explained by the fact that these two types share high similarities with execution I (see Table 4).



**Figure 13.** Execution of type VII where the middle part (i.e., step to the side) is performed as type V. Three different window sizes are represented: 100, 60 and 20 samples. The red dotted line represents the prediction of HMM V and the blue line indicates the prediction of HMM I (correct movement).



There is a certain trade-off for choosing the window size. A smaller window can provide a frequent feedback, but a slightly noisier prediction. Figure 13 presents the example of a correct sequence, except the middle part is performed as execution V (step with bended knee). In this figure the amount of feedback moments is represented on the  $x$ -axis and a normalized classification value on the  $y$ -axis. The sampling rate is 50 Hz (20 ms per sample). Window sizes of 100, 60 and 20 represent approximately every second, twice a second and five times a second feedback, respectively. This example shows that the accuracy of the prediction (identification of correct vs. incorrect executions) is not significantly altered by the window sizes, which confirms the pertinence of an HMM approach for real-time applications.

## 6. Usability of the System

### 6.1. Participants and Procedure

In order to evaluate if the platform can be easily used by the end users, a usability test was carried out. A total of 41 right handed participants (distributed in 32 males and 9 females) volunteered for the experiment. No specific skill was necessary to take part to the study. 92.3% of the subjects was between 18 and 30 years old. All participants reported that they had never used a tele-rehabilitation system before.

Participants were informed about the purpose of the study and signed a consent form to participate. Then, subjects were asked to take an accommodation phase, in which they followed a tutorial on the use of the platform. Once the scenario of tele-rehabilitation was clarified and after the participants were confident with the application, they were asked to complete five tasks described in the next section. The total duration for the completion of the tasks was around 15 min. Once the whole tasks were completed, the subjects were requested to fill in the IBM Computer System Usability Questionnaire (CSUQ) described in Table 6 [32]. CSUQ is a questionnaire based on a Likert scale, in which a statement is made and the respondent then indicates the degree of agreement or disagreement with the statement on a 7-point scale, from 'strongly disagree' (1) to 'strongly agree' (7).

### 6.2. Tasks

The rehabilitation program is implemented in the platform according to 3 chronological stages. The intensity of the therapeutic exercises increases progressively from stage 1 (the week following the surgery) to stage 3 (about 3 weeks after the surgery). During the usability test, participants had to perform five tasks related to activities implemented in these different stages.

#### 6.2.1. Task 1: Consult Therapeutic Instructions of Stage 1

In this task, subjects were asked to access and understand the documentation provided by the core component regarding 'how to use the platform'. They could select among different learning modalities (text, audio, and video).

#### 6.2.2. Task 2: Perform Rehabilitation Exercises of Stage 1

This task involved several subtasks. First, participants had to answer a questionnaire to self-assess their health status and unlock the exercise interface (meaning that they are in physical condition to pursue). Second, they had to select an exercise of stage 1 (low intensity exercises) and consult recommendations to perform it properly. Finally, the exercise had to be executed in front of the Kinect.

#### 6.2.3. Task 3: Consult Therapeutic Instructions of Stage 2

Again, participants had to access and consult general recommendation documents developed for the therapeutic education of the patients in stage 2 (medium intensity exercises). At the end, and before continuing to the rehabilitation program, they had to accept the conditions by clicking a check-box.

#### 6.2.4. Task 4: Perform Rehabilitation Exercises of Stage 2

As in task 2, subjects were requested to perform an exercise available in stage 2. An addition requirement for this task was to abort the series of repetition by using the functionality ‘cancel’ and, then, to fill in the questionnaire to justify the cause of the suspension.

#### 6.2.5. Task 5: Consult and Send a Message to the Therapist

In this task, participants had to access the mailbox integrated in the platform, in order to consult new messages and send a text-based or audio-based email to their physiotherapists.

### 6.3. Results

Overall, the participants mostly agreed with the statements. In most cases, the averages of the responses were greater than 5 with a standard deviation (SD) less than 1.19. The global result to the questions (#09, #10 and #11) that belong to the information quality or INFOQUAL (Mean = 5.49, Median = 6.00, SD = 1.15) suggests the platform provides a good online aid and documentation. Nevertheless, question #09 is assessed less positively (Mean = 4.77, Median = 5.00, SD = 1.51), which means that users tend to consider the error messages as not helpful to solve problems. Finally, the result to question #15 (Mean = 5.92, Median = 6.00, SD = 0.84) clearly indicates that the participants agree on the organization of the information of the user interface.

**Table 6.** Structure and results of the IBM Computer System Usability Questionnaire.

| Category  | #  | Question  | Mean | SD   |
|-----------|----|---|------|------|
| SYSUSE    | 01 | Overall, I am satisfied with how easy it is to use this system.   | 5.82 | 0.88 |
|           | 02 | It was simple to use this system.   | 5.92 | 0.74 |
|           | 03 | I could effectively complete my work using this system.   | 6.03 | 1.01 |
|           | 04 | I was able to complete my work quickly using this system.   | 5.74 | 0.97 |
|           | 05 | I was able to efficiently complete my work using this system.   | 5.79 | 1.10 |
|           | 06 | I felt comfortable using this system.   | 5.82 | 0.99 |
|           | 07 | It was easy to learn to use this system.  | 5.90 | 0.94 |
|           | 08 | I believe I could become productive quickly using this system.  | 5.72 | 0.92 |
| INFOQUAL  | 09 | The system gave error messages that clearly tell me how to fix problems.  | 4.77 | 1.51 |
|           | 10 | Whenever I made a mistake using the system, I could recover easily and quickly.   | 5.31 | 1.51 |
|           | 11 | The information (such as online help, on-screen messages, and other documentation) provided with this system was clear. | 5.38 | 1.04 |
|           | 12 | It was easy to find the information I needed.   | 5.72 | 1.15 |
|           | 13 | The information provided for the system was easy to understand.   | 5.69 | 0.95 |
|           | 14 | The information was effective in helping me complete the tasks and scenarios.   | 5.64 | 1.01 |
|           | 15 | The organization of information on the system screens was clear.  | 5.92 | 0.84 |
| INTERQUAL | 16 | The interface of this system was pleasant.  | 5.54 | 1.19 |
|           | 17 | I liked using the interface of this system.   | 5.38 | 1.07 |
|           | 18 | This system has all the functions and capabilities I expect it to have.   | 5.59 | 1.09 |
| OVERALL   | 19 | Overall, I am satisfied with this system.   | 5.74 | 0.88 |

The average values for the IBM CSUQ measures are presented in Table 6. Concerning the category SYSUSE (Mean = 5.84, Median = 6.00, SD = 0.94), the results show that the participants tend to consider the system as useful. The results to the INFOQUAL category (Mean = 5.49, Median = 6.00, SD = 1.15) also suggest a good quality of the information presented on the user interface. The participants have indicated that the information provided by the system was easily understood and helpful to complete the tasks and scenarios. Nevertheless, future work will concentrate on improving the feedback provided by the system, especially when errors occur. Interface quality or INTERQUAL (Mean = 5.50, Median = 6.00, SD = 1.11) shows that the user interfaces of the platform are adapted to the end user. However, it seems that the tool requires some additional improvements, such as: (i) a simplification of the visual interface for the execution of the rehabilitation exercises, and (ii) the incorporation of a speech-based remote interaction with the application. Overall satisfaction or OVERALL (Mean = 5.74, Median = 6.00, SD = 0.88) suggests that the system is positively evaluated by the users. It can be concluded that the system usability of the current version of the platform is assessed positively by the

participants, who seem to like the general organization and functionalities of the application, even if some upgrades are required.

## 7. Discussion

This study describes the development of a tele-rehabilitation platform to support motor recovery. Exercises designed for patients after hip arthroplasty are presented as a case study. Nevertheless, this platform is built on a modular architecture that facilitates its adaptation for any other physical rehabilitation programs (e.g., hemiparetic patients). In addition, an artificial intelligence module is integrated in the platform, in order to assess the quality of the therapeutic movements performed by the patients. Two possible methods are tested: Dynamic Time Warping and Hidden Markov Models.

First, a DTW approach was used to evaluate the motion by comparison to a reference. The experimental results show a high correlation between the assessment performed by the computer and by the health professionals. The slight discrepancy between the two evaluations has both technical and human explanations. The accuracy is affected by partial occultations of some joints inherent to the motion capture device. In addition, we observed certain inconsistencies in the judgment of the therapists, which limit the reliability of the human reference.

Second, a HMM approach is tested to detect variance within movement, caused by errors or compensatory movements that may occur during the completion of the therapeutic exercise. HMMs are trained on these errors and compensatory actions. Although the setting of the experiment was controlled, the classification also included intrapersonal and interpersonal variances, as a model that classified a determined subject was merely trained with the data of the other participants. It suggests that the proposed assessment algorithm has a fair capability of generalization. A high classification accuracy of the movements is obtained by building a general model that can be applied to any subject. A real-time analysis enables us to detect four out of five faulty movements, when these errors briefly occur in the beginning, middle or end of a correct execution of the exercise. The same level of accuracy is maintained whatever the detection rate. These findings demonstrate that the HMM is an appropriate method to provide real-time feedback regarding the correctness of the rehabilitation movement performed by a patient. This approach is successfully applied on a real-time assessment of components of the movement, which are discriminated in several classes that differ on extremely subtle aspects.

Both methods, DTW and HMM, have advantages and disadvantages. It is not necessary to collect lots of data to perform an assessment based on DTW, because just a single reference is required to apply the technique. On the contrary, HMM is a probabilistic approach, in which the accuracy of the classification depends on the quantity of available data. The bigger is the dataset, the higher is the precision. This aspect is particularly relevant if you presume a difficult access to collect enough data to train the model. On the other hand, HMM is more adapted than DTW to perform a real-time evaluation, as it creates a model that can be dynamically generated and adjusted. Another advantage of utilizing HMM is the fact that it is more robust than DTW when the quantity of features to discriminate increases.

Finally, the usability of the platform is positively assessed by end users. The empirical results based on subjective perception and self-reported feedback show that the application is useful, effective, efficient, and easy to use. In addition, the evaluation of the user experience enables us to identify usability aspects that should be implemented, in order to improve the visual interface. The experiment indicates that the error messages should be as detailed as possible. Our results suggest that providing the patients with a systematic and non-ambiguous feedback is a critical aspect to ensure a perfect acceptance of a tele-rehabilitation system.

The main contribution of this manuscript is to provide a holistic description of a solution to implement a tele-rehabilitation platform that presents three advantages rarely available in other similar systems. The first is the implementation of a scalable and modular architecture to facilitate (i) the adaptation to other physical rehabilitation programs, and (ii) the evolution of the system for integrating new software (e.g., game engines) and hardware (e.g., motion capture sensor) technologies.

The second is the integration of an artificial intelligence (AI) module that assesses the quality of the movement through two possible solutions: DTW or HMM. This aspect, which is absent in comparable applications [12,33,34], is however crucial to ensure the efficiency and safety of a remote motor rehabilitation. The third is the gamification of the therapeutic exercises, in order to enhance the motivation of the patients to complete the rehabilitation program at home.

Nevertheless, the current version of the platform presents certain limitations that will be overcome in the future. The vision-based motion capture implies occlusion issues that alter the accuracy of the assessment algorithms. A possible solution would be to implement a hybrid system that integrates both Kinect and inertial sensors [35]. In addition, it will be necessary to include additional features to get a more advanced discrimination of the movements assessed by a HMM approach. In particular, these features should stand for kinematic variables related to the upper part of the body, which will improve the identification of compensatory movements. In terms of evaluation of the AI algorithms, the future experimental protocols will simplify the assessment tasks of the therapists, in order to reduce the discrepancies. For instance, this assessment could be based on videos and not performed on the fly, and/or each expert could have to gauge fewer parameters simultaneously. Finally, we are aware that it is required to carry out usability tests on real patients and during the whole completion period of the rehabilitation program, as indicated in [36]. For this purpose, we plan to apply usability questionnaires specifically designed to measure the user experience with e-Health platforms, such as USEQ [36] and TUQ [37].

To conclude, we would like to highlight the fact that the application is built considering both the health professional requirements and the patient limitations. The architecture of the system is modular and flexible enough to be integrated in a global smart home. In addition, the proposed approach intends to deliver a therapeutic program as a service over the Internet. It means that the therapists can implement a personalized rehabilitation program that their patients will be able to execute anywhere through the platform. In that sense, the system promotes a smart medical environment, in which both patients and health professionals can interact remotely from home, for the former, and the medical office, for the latter; transforming the application into a ubiquitous tool to support the patient's recovery.

**Author Contributions:** Conceptualization, Y.R., J.L.P.M. and D.E.; methodology, Y.R., J.L.P.M. and L.L., software, K.J., L.L., Y.R. and J.L.P.M.; validation, Y.R., L.L. and J.L.P.M.; data curation, L.L., J.L.P.M. and D.E.; data analysis, Y.R., L.L., J.L.P.M. and M.G.; writing and editing, Y.R., J.L.P.M., L.L. and K.J.; project administration, Y.R.; funding acquisition, Y.R.

**Funding:** This research was funded by REDCEDIA, grant number CEPRA XI-2017-15.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rybarczyk, Y.; Kleine Deters, J.; Cointe, C.; Esparza, D. Smart web-based platform to support physical rehabilitation. *Sensors* **2018**, *18*, 1344. [[CrossRef](#)] [[PubMed](#)]
2. Rybarczyk, Y.; Goncalves, M.J. WebLisling: A web-based therapeutic platform for rehabilitation of aphasic patients. *IEEE Latin Am. Trans.* **2016**, *14*, 3921–3927. [[CrossRef](#)]
3. Mani, S.; Sharma, S.; Omar, B.; Paungmali, A.; Joseph, L. Validity and reliability of Internet-based physiotherapy assessment for musculoskeletal disorders: A systematic review. *J. Telemed. Telecare* **2017**, *23*, 379–391. [[CrossRef](#)] [[PubMed](#)]
4. Rybarczyk, Y.; Kleine Deters, J.; Gonzalvo, A.; Gonzalez, M.; Villarreal, S.; Esparza, D. ePHoRt project: A web-based platform for home motor rehabilitation. In Proceedings of the 5th World Conference on Information Systems and Technologies, Madeira, Portugal, 11–13 April 2017; pp. 609–618. [[CrossRef](#)]
5. Kleine Deters, J.; Rybarczyk, Y. Hidden Markov Model approach for the assessment of tele-rehabilitation exercises. *Int. J. Artif. Intell.* **2018**, *16*, 1–19.

6. Rybarczyk, Y.; Kleine Deters, J.; Aladro Gonzalo, A.; Esparza, D.; Gonzalez, M.; Villarreal, S.; Nunes, I.L. Recognition of physiotherapeutic exercises through DTW and low-cost vision-based motion capture. In Proceedings of the 8th International Conference on Applied Human Factors and Ergonomics, Los Angeles, CA, USA, 17–21 July 2017; pp. 348–360. [\[CrossRef\]](#)
7. Fortino, G.; Gravina, R.A. Cloud-assisted wearable system for physical rehabilitation. In *Communications in Computer and Information Science*; Barbosa, S.D.J., Chen, P., Filipe, J., Kotenko, I., Sivalingam, K.M., Washio, T., Yuan, J., Zhou, L., Eds.; Springer: Heidelberg, Germany, 2015; Volume 515, pp. 168–182.
8. Fortino, G.; Parisi, D.; Pirrone, V.; Fatta, G.D. BodyCloud: A SaaS approach for community body sensor networks. *Future Gener. Comput. Syst.* **2014**, *35*, 62–79. [\[CrossRef\]](#)
9. Bonnechère, B.; Jansen, B.; Salvia, P.; Bouzahouene, H.; Omelina, L.; Moiseev, F.; Sholukha, V.; Cornelis, J.; Rooze, M.; Van Sint Jan, S. Validity and reliability of the Kinect within functional assessment activities: Comparison with standard stereophotogrammetry. *Gait Posture* **2014**, *39*, 593–598. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Pedraza-Hueso, M.; Martín-Calzón, S.; Díaz-Pernas, F.J.; Martínez-Zarzuela, M. Rehabilitation using Kinect-based games and virtual reality. *Procedia Comput. Sci.* **2015**, *75*, 161–168. [\[CrossRef\]](#)
11. Morais, W.O.; Wickström, N. A serious computer game to assist Tai Chi training for the elderly. In Proceedings of the 1st IEEE International Conference on Serious Games and Applications for Health, Washington, DC, USA, 16–18 November 2011; pp. 1–8. [\[CrossRef\]](#)
12. Lin, T.Y.; Hsieh, C.H.; Der Lee, J. A kinect-based system for physical rehabilitation: Utilizing Tai Chi exercises to improve movement disorders in patients with balance ability. In Proceedings of the 7th Asia Modelling Symposium, Hong Kong, China, 23–25 July 2013; pp. 149–153. [\[CrossRef\]](#)
13. Hoang, T.C.; Dang, H.T.; Nguyen, V.D. Kinect-based virtual training system for rehabilitation. In Proceedings of the International Conference on System Science and Engineering, Ho Chi Minh City, Vietnam, 21–23 July 2017; pp. 53–56. [\[CrossRef\]](#)
14. Okada, Y.; Ogata, T.; Matsuguma, H. Component-based approach for prototyping of Tai Chi-based physical therapy game and its performance evaluations. *Comput. Entertain.* **2017**, *14*, 1–20. [\[CrossRef\]](#)
15. Da Gama, A.; Chaves, T.; Figueiredo, L.; Teichrieb, V. Guidance and movement correction based on therapeutic movements for motor rehabilitation support systems. In Proceedings of the 14th Symposium on Virtual and Augmented Reality, Rio de Janeiro, Brazil, 28–31 May 2012; pp. 28–31. [\[CrossRef\]](#)
16. Brokaw, E.B.; Lum, P.S.; Cooper, R.A.; Brewer, B.R. Using the Kinect to limit abnormal kinematics and compensation strategies during therapy with end effector robots. In Proceedings of the 2013 IEEE International Conference on Rehabilitation Robotics, Seattle, WA, USA, 24–26 June 2013; pp. 24–26. [\[CrossRef\]](#)
17. Zhao, W.; Reinthal, M.A.; Espy, D.D.; Luo, X. Rule-based human motion tracking for rehabilitation exercises: Realtime assessment, feedback, and guidance. *IEEE Access* **2017**, *5*, 21382–21394. [\[CrossRef\]](#)
18. Antón, D.; Goñi, A.; Illarramendi, A. Exercise recognition for Kinect-based telerehabilitation. *Methods Inf. Med.* **2015**, *54*, 145–155. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Gal, N.; Andrei, D.; Nemes, D.I.; Nadasan, E.; Stoicu-Tivadar, V. A Kinect based intelligent e-rehabilitation system in physical therapy. *Stud. Health Technol. Inf.* **2015**, *210*, 489–493. [\[CrossRef\]](#)
20. López-Jaquero, V.; Rodríguez, A.C.; Teruel, M.A.; Montero, F.; Navarro, E.; Gonzalez, P. A bio-inspired model-based approach for context-aware post-WIMP tele-rehabilitation. *Sensors* **2016**, *16*, 1689. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Rybarczyk, Y.; Vernay, D. Educative therapeutic tool to promote the empowerment of disabled people. *IEEE Lat. Am. Trans.* **2016**, *14*, 3410–3417. [\[CrossRef\]](#)
22. Jaiswal, S.; Kumar, R. *Learning Django Web Development*; O'Reilly: Sebastopol, CA, USA, 2015; ISBN 978-1783984404.
23. Riehle, D. Composite design patterns. In Proceedings of the 12th ACM SIGPLAN ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Atlanta, GA, USA, 5–9 October 1997; pp. 218–228. [\[CrossRef\]](#)
24. Hillar, G.C. *Django RESTful Web Services: The Easiest Way to Build Python RESTful APIs and Web Services with Django*; Packt: Birmingham, UK, 2018; ISBN 978-1788833929.
25. Kinectron: A Realtime Peer Server for Kinect 2. Available online: <https://kinectron.github.io/docs/server.html> (accessed on 5 April 2018).
26. Jakobus, B. *Mastering Bootstrap 4: Master the Latest Version of Bootstrap 4 to Build Highly Customized Responsive Web Apps*; Packt: Birmingham, UK, 2018; ISBN 978-1788834902.



27. Wu, Q.; Xu, G.; Li, M.; Chen, L.; Zhang, X.; Xie, J. Human pose estimation method based on single depth image. *IET Comput. Vis.* **2018**, *12*, 919–924. [[CrossRef](#)]
28. Yamato, J.; Ohya, J.; Ishii, K. Recognizing human action in time-sequential images using Hidden Markov Model. In Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15–18 June 1992. [[CrossRef](#)]
29. Fiosina, J.; Fiosins, M. Resampling based modelling of individual routing preferences in a distributed traffic network. *Int. J. Artif. Intell.* **2014**, *12*, 79–103.
30. Yao, A.; Gall, J.; Fanelli, G.; Van Gool, L. Does human action recognition benefit from pose estimation? In Proceedings of the 22nd British Machine Vision Conference, Dundee, UK, 29 August–2 September 2011. [[CrossRef](#)]
31. Smyth, P. Clustering using Monte Carlo cross-validation. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996.
32. Lewis, J.R. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum. Comput. Interact.* **1995**, *7*, 57–78. [[CrossRef](#)]
33. Antón, D.; Berges, I.; Bermúdez, J.; Goñi, A.; Illarramendi, A. A telerehabilitation system for the selection, evaluation and remote management of therapies. *Sensors* **2018**, *18*, 1459. [[CrossRef](#)] [[PubMed](#)]
34. Antón, D.; Kurillo, G.; Goñi, A.; Illarramendi, A.; Bajcsy, R. Real-time communication for kinect-based telerehabilitation. *Future Gener. Comput. Syst.* **2017**, *75*, 72–81. [[CrossRef](#)]
35. Gowing, M.; Ahmadi, A.; Destelle, F.; Monaghan, D.S.; O'Connor, N.E.; Moran, K. Kinect vs. low-cost inertial sensing for gesture recognition. In Proceedings of the 20th International Conference on Multimedia Modeling, Dublin, Ireland, 6–10 January 2014. [[CrossRef](#)]
36. Gil-Gómez, J.A.; Manzano-Hernández, P.; Albiol-Pérez, S.; Aula-Valero, C.; Gil-Gómez, H.; Lozano-Quilis, J.A. USEQ: A short questionnaire for satisfaction evaluation of virtual rehabilitation systems. *Sensors* **2017**, *17*, 1589. [[CrossRef](#)] [[PubMed](#)]
37. Parmanto, B.; Lewis, A.N., Jr.; Graham, K.M.; Bertolet, M.H. Development of the telehealth usability questionnaire (TUQ). *Int. J. Telerehabil.* **2016**, *8*, 3–10. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).