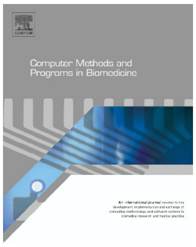


journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals

Aras Yurtman, Billur Barshan[*]

Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, TR-06800 Ankara, Turkey

ARTICLE INFO

Article history:
Received 15 October 2013
Received in revised form 9 July 2014
Accepted 14 July 2014

Keywords:
Physical therapy
Motion sensors
Inertial sensors
Dynamic time warping
Pattern recognition
Motion capture

ABSTRACT

We develop an autonomous system to detect and evaluate physical therapy exercises using wearable motion sensors. We propose the *multi-template multi-match dynamic time warping* (MTMM-DTW) algorithm as a natural extension of DTW to detect multiple occurrences of more than one exercise type in the recording of a physical therapy session. While allowing some distortion (warping) in time, the algorithm provides a quantitative measure of similarity between an exercise execution and previously recorded templates, based on DTW distance. It can detect and classify the exercise types, and count and evaluate the exercises as correctly/incorrectly performed, identifying the error type, if any. To evaluate the algorithm's performance, we record a data set consisting of one reference template and 10 test executions of three execution types of eight exercises performed by five subjects. We thus record a total of 120 and 1200 exercise executions in the reference and test sets, respectively. The test sequences also contain idle time intervals. The accuracy of the proposed algorithm is 93.46% for exercise classification only and 88.65% for simultaneous exercise and execution type classification. The algorithm misses 8.58% of the exercise executions and demonstrates a false alarm rate of 4.91%, caused by some idle time intervals being incorrectly recognized as exercise executions. To test the robustness of the system to unknown exercises, we employ leave-one-exercise-out cross validation. This results in a false alarm rate lower than 1%, demonstrating the robustness of the system to unknown movements. The proposed system can be used for assessing the effectiveness of a physical therapy session and for providing feedback to the patient.

© 2014 Elsevier Ireland Ltd. All rights reserved.

1. Introduction and background

Physical therapy is an important type of rehabilitation for patients with various disorders. Cardiopulmonary medicine, neurology, orthopedics, and pediatrics are branches of medicine that may benefit from physical therapy [1,2]. Physical therapy usually requires exercising in a hospital or a rehabilitation center under the supervision of a specialist who assigns one or more exercises to the patient [3]. After learning how to perform the exercises correctly, patients usually need to continue exercising at home, where they receive no feedback [4]. Even at the hospital, specialists cannot follow each patient continuously during their exercise sessions because the

[*] *Corresponding author*. Tel.: +90 3122902161; fax: +90 3122664192.
E-mail addresses: yurtman@ee.bilkent.edu.tr (A. Yurtman), billur@ee.bilkent.edu.tr (B. Barshan).

http://dx.doi.org/10.1016/j.cmpb.2014.07.003
0169-2607/© 2014 Elsevier Ireland Ltd. All rights reserved.

former often alternate between several patients or have other tasks to do, a situation that can result in insufficient, inaccurate, and often subjective feedback [5,6].

An important issue in physical therapy is to evaluate the exercises and assess the effectiveness of an exercise session. Previous studies consider energy expenditure [7], the total duration of the exercise session, or the time period during which the patient is physically active [8]. The system in [3] generates a warning whenever relevant parameters exceed predefined thresholds to reduce the risk of over-exercising. These approaches may be misleading or fail if the patient performs the exercises incorrectly or at an unusual pace. Previous studies have not considered detecting sequential exercise executions and providing an objective evaluation of their accuracy to assess the effectiveness of a physical therapy session. Such a process is also difficult and tedious for a specialist, even when he is responsible for only a single patient, and impossible when he is monitoring several. Patients and specialists would highly benefit from an automated system.

Several different sensor modalities are used in physical rehabilitation, including inertial [9–12], visual [13–15], strain [4,16], medical [6], physiological, kinetic, and environmental sensors [17], or some combination of these [18,19]. Automatic monitoring of the people performing physical therapy exercises should be done without restricting their independence, intruding on their privacy, or degrading their quality of life. A commonly used approach is to fix cameras in the environment, which intrudes on privacy and usually has a relatively high installation cost. The main advantage of this approach is that the person does not have to wear or carry any sensors or devices on their body. This approach may also eliminate problems related to misplacing sensors on the body, although some vision-based systems require wearing/pasting special tags or markers. This method may be acceptable when the person always performs exercises at the same place, but when the exercises are performed in different places, e.g., indoors and outdoors, this approach becomes unsuitable. We believe that wearable sensors are superior to camera systems in these respects.

In [20], participants 3–9 months post-stroke with mild to moderate motor impairment of one arm wore an accelerometer on each arm outside the laboratory for three days before and after treatment or an equivalent no-treatment period. The use of the more impaired arm in daily life was assessed using low-pass filtered accelerometer recordings. Other studies that focus on post-stroke rehabilitation are [21–23].

In numerous studies, a 3D real-time human body model is constructed to observe movements [5,24,25]. In [24,26,27], patients perform the given exercises to complete tasks in video-game-like virtual environments, making exercise sessions more enjoyable.

Another approach to observing movements is to use *biofeedback* devices that transform sensor measurements from the body into sound, a blinking LED, or an observable shape on a screen [28–30]. Biofeedback provides detailed information regarding the lengthening, shortening, and physical exertion of a muscle. It also allows comparing the data of a healthy muscle to a non-healthy one when performing the same exercise. Biofeedback devices are sometimes combined with electromyography. Although these devices have become portable recently, older devices are immobile and costly, and are mostly used in hospitals or rehabilitation centers [31]. In addition, most devices either do not evaluate a patient's performance or they evaluate the results using simple thresholding. Hence, they require a specialist or the patient to evaluate the feedback [31], both of whom may be highly subjective. Therefore, biofeedback devices cannot replace an attendant specialist most of the time.

Below, we provide a summary of studies aiming to assess the accuracy of physical therapy exercises or classify them as correct/incorrect:

Fergus et al. [5] propose a tele-rehabilitation system that collects and stores the patient's motion data, utilizing body area and sensor networks, including inertial sensors. The system virtually simulates the patient's body motions on a 3D human body model in real time or using the stored data. The physician or physical therapist monitors the motions remotely to evaluate the patient's progress. This proposed solution is impractical and does not significantly improve inspection time because the system provides no information regarding the patient's movement capability, movement accuracy, or progress.

Using five body-worn tri-axial accelerometers, Taylor et al. [11] build a classifier that labels incorrectly performed exercises for knee osteoarthritis, a degenerative disease of the knee joint. Three exercises are performed by nine healthy subjects. The exercises are performed in the correct way as well as with particular errors. Several features extracted from the accelerometer data are used in the AdaBoost classifier to classify the exercises as correct or as having a particular error type. However, multiple errors are not allowed by the methodology and the classification accuracy is about 70% in most cases, which is not very high.

In [12], an Android application estimating the accuracy (i.e., score) of balance board exercises is developed, using a smartphone's internal accelerometer and magnetometer. A complex rule-based algorithm is proposed to obtain a score value closest to the score given by an expert; the difference between human and automatic scores is found to be fewer than 10 points in more than 75% of the exercises on a 0 to 100 scale. However, the proposed methodology does not yield an optimal solution, and different rule-based scoring algorithms are used for different exercise types, which makes it difficult to add a new exercise to the system.

In the MyHeart system [16], the accuracy of arm movements for post-stroke rehabilitation is determined using strain sensors. Healthy subjects wearing tight-fitting garments with printed strain sensors imitate how post-stroke patients might perform each of seven exercise types correctly and incorrectly under the supervision of a physician or therapist. An exercise is considered correctly performed if the similarity between the recorded signal and a pre-recorded template, calculated by the open-end DTW (see Appendix A.2), exceeds a threshold [12]. The system provides real-time feedback to the patient with an average classification accuracy of 85%. The main disadvantage of the system is the difficulty of putting on the garment for a post-stroke patient, even with help.

In [4], strain sensors worn on the arm are used to provide real-time feedback to patients undergoing motor rehabilitation. Seven exercises are executed correctly and incorrectly

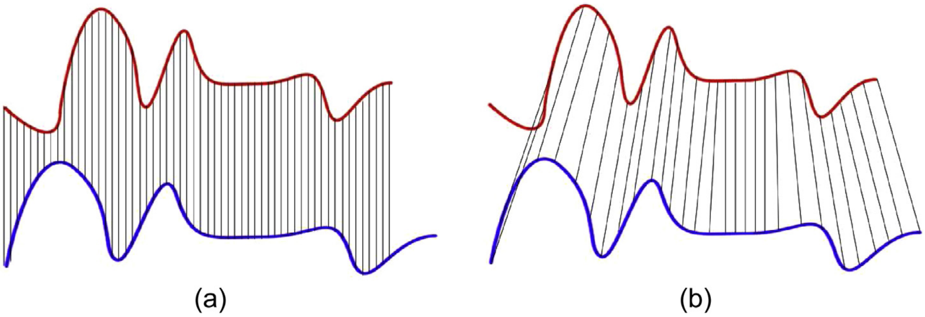


Fig. 1 – Comparison of the Euclidean and DTW distance measures. (a) The Euclidean measure compares the samples at the same time instants, whereas (b) the DTW measure compares samples with similar shapes to minimize the distance. Retrieved from http://upload.wikimedia.org/wikipedia/commons/6/69/Euclidean_vs_DTW.jpg.

at various speeds by a healthy subject wearing a left-handed sensorized long-sleeved shirt. The system checks whether the measured signals "match *at most once* a prefix of one of several stored reference templates" [4] to detect which exercise is being performed. The distance obtained by the open-end DTW algorithm is used as a similarity measure between the strain signals and fed to a one-nearest-neighbor classifier both for exercise recognition and correctness/error type classification.

The m-Physio platform [10] classifies physical rehabilitation exercises using an accelerometer. A smartphone with a tri-axial accelerometer is fixed to the patient's leg or arm depending on the type of exercise he performs. The patient first performs the exercises correctly under the supervision of a specialist who records the reference templates. The signals recorded from test exercises are compared with the reference templates using the ordinary DTW algorithm. One disadvantage of the m-Physio platform is the need to determine the values of four parameters in the exercise-capture phase: the minimum and maximum duration of each exercise, the sampling frequency of the accelerometer, and the amount of smoothing applied to the signals. The specialist sets the values by trial and error, which may easily alter the system performance. When using the system, the patient needs to touch the screen to mark the beginning and the end of each exercise execution, which is another disadvantage because some patients (e.g., elderly, disabled, and/or stroke patients) may not be able to touch the screen in a timely manner or may forget to do so. The system provides feedback as *correct, incorrect, too short/long in time* for each execution. The patient's statistics are uploaded to a centralized database so that the specialist can remotely check his progress through a web interface.

In the previous studies, exercise executions are usually cropped manually and considered in an isolated fashion; the subject needs to mark each execution by pressing a button or perform each exercise when informed by the system by a sound or on-screen notification. Sequential executions and idle time periods are not considered.

We propose a completely autonomous system to detect the sequential executions of one or multiple exercise types during an exercise session, classify the exercise type, evaluate each execution as correctly/incorrectly performed, and identify the error type and idle time periods, if any [32]. During the physical therapy session, the patient wears small, inexpensive motion sensor units. The sensors are lightweight and can be easily worn and carried, making home-based rehabilitation possible. The patient first executes the exercises under the supervision of a specialist who records reference templates. The patient can then perform the exercises anywhere, provided that he properly wears the sensors. He does not need to press a button to mark the beginning or end of an execution or select the exercise type he intends to perform. The system compares the detected executions with the templates recorded while under supervision, and quantifies their similarity. We develop the *multi-template multi-match dynamic time warping* (MTMM-DTW) algorithm as a natural extension of DTW. Using MTMM-DTW, multiple template sequences of different durations can be searched in a test sequence of any duration based on the DTW dissimilarity measure. The system can provide statistical information about the exercise session at any desired level of detail to the specialist and provide feedback to the patient. The primary applicability of the system is to the rehabilitation of orthopedic patients. This is an early proof-of-principle study in the pre-clinical stage and the methodology has not yet been demonstrated in clinical settings or to be clinically efficacious.

This paper is organized as follows: In Section 2, we describe the MTMM-DTW algorithm. We present the experimental methodology and the results in Section 3. In Section 4, we provide a discussion of the results and related issues. Finally, we draw conclusions and indicate directions for future research in Section 5. We provide background information on the DTW algorithm and its variations in Appendix.

2. Multi-template multi-match DTW algorithm

We develop an extension of the DTW algorithm for detecting multiple occurrences of multiple exercise types in a sequence recorded during a physical therapy session. This method makes it possible to classify exercise types, identify whether an exercise is performed correctly (while identifying the type of error, if any), and count the number of exercises performed.

We use an approach based on DTW which has more flexibility than using absolute and Euclidean distance measures because it allows time warping in comparing and matching

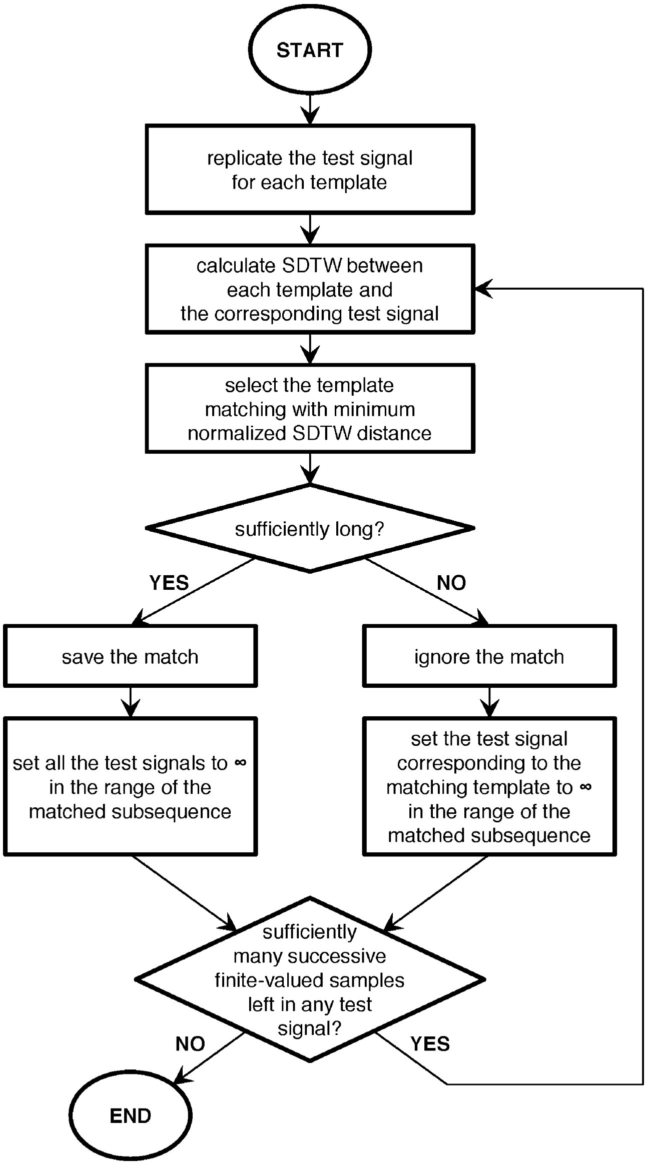

START
replicate the test signal for each template
calculate SDTW between each template and the corresponding test signal
select the template matching with minimum normalized SDTW distance
sufficiently long?
YES
NO
save the match
ignore the match
set all the test signals to ∞ in the range of the matched subsequence
set the test signal corresponding to the matching template to ∞ in the range of the matched subsequence
sufficiently many successive finite-valued samples left in any test signal?
NO
YES
END

Fig. 2 – The flowchart of the MTMM-DTW algorithm.

similar parts of two sequences (see Fig. 1 for an example), which may be beneficial when time variations in different executions of the same physical therapy exercise are considered. The patient may perform the exercise at different paces, which should be tolerated to some extent. On the other hand, the distance measure should not tolerate significant differences in amplitude that may occur; for example, if the patient performs the exercises at a low amplitude or holds for some time in the wrong position. If absolute or Euclidean distance measures are used, both time and amplitude variations affect the measured distance and it is not possible to allow one of them while penalizing the other. In contrast, the DTW algorithm naturally compensates for time variations but not for changes in amplitude, both of which are more desirable in this scheme.

The ordinary DTW algorithm matches the first and last samples of two sequences, $\mathbf{x}$ and $\mathbf{y}$ (not necessarily of the same length), to each other, possibly warping the time axes in between to obtain maximum overall similarity. On the other hand, the subsequence DTW (SDTW) algorithm matches the subsequence of the test sequence $\mathbf{y}$ that is most similar to the template sequence $\mathbf{x}$. If the template sequence may occur more than once in the test sequence, it may be desirable to detect *all* of the subsequences of $\mathbf{y}$ that resemble $\mathbf{x}$. In the *single-template multi-match DTW* (STMM-DTW) algorithm that we developed for this purpose [33], the SDTW algorithm is executed more than once to detect possibly multiple occurrences of a *single* template sequence $\mathbf{x}$ in the test sequence $\mathbf{y}$. In some applications, it may be necessary to search for *multiple* templates in a test sequence. To detect possibly multiple occurrences of each of the K templates $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$ in $\mathbf{y}$, we propose the MTMM-DTW algorithm (see Fig. 2).

Algorithm 1. MTMM-DTW algorithm

```
$M \leftarrow \text{length}(\mathbf{y})$
for $k = 1 \rightarrow K$ do
  $N^{(k)} \leftarrow \text{length}(\mathbf{x}^{(k)})$
  $M_{\text{left}}(k) \leftarrow M$ {$M_{\text{left}}$ is the maximum number of successive unmatched samples in $\mathbf{y}$ for the $k$th template $\mathbf{x}^{(k)}$}
  $\tilde{\mathbf{y}}^{(k)} \leftarrow \mathbf{y}$ {the test sequence $\mathbf{y}$ replicated as $\tilde{\mathbf{y}}^{(k)}$ to be used for each template $\mathbf{x}^{(k)}$}
end for
while $M_{\text{left}}(k) \geq \alpha N^{(k)}, \exists k \in \{1, \ldots, K\}$ do
  compute SDTW$(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}^{(k)})$ and save the SDTW$^{(k)}$ distance, $m_1^{(k)}$, and $m_2^{(k)}$ $\forall k$ ($m_1$ and $m_2$ are the first and the last sample indices of the matched subsequence of $\mathbf{y}^{(k)}$)
  $k^* \leftarrow \arg\min_{k \in \{1,\ldots,K\}} \frac{1}{N^{(k)}} \text{SDTW}^{(k)}$ {find matching subsequence of the template having the minimum DTW distance *per sample of the template* by normalizing DTW distances with respect to the template lengths}
  if $m_2^{(k^*)} - m_1^{(k^*)} + 1 \geq \alpha N^{(k^*)}$ {the last matched subsequence is sufficiently long}
  then add the last match with template number $k^*$ to the list
    $\tilde{\mathbf{y}}^{(k)}(m_1^{(k^*)} : m_2^{(k^*)}) \leftarrow \infty, \forall k$ {prevent the matched samples from being matched in the DTW executions that follow for all templates}
  else ignore the last match {the last matched subsequence is too short}
     $\tilde{\mathbf{y}}^{(k^*)}(m_1^{(k^*)} : m_2^{(k^*)}) \leftarrow \infty$ {prevent the last matched subsequence from being matched to the same template in the DTW executions that follow}
  end if
  for $k = 1 \rightarrow K$ do
    $M_{\text{left}}(k) \leftarrow$ the maximum number of successive finite-valued samples in $\mathbf{y}$ for the $k$th template $\mathbf{x}^{(k)}$
  end for
end while
```

Because the duration of each template is, in general, different, and the DTW distance is the cumulative sum of the pairwise costs between the samples of the warped template and test sequences, the DTW distances need to be normalized by the length of the template to be able to make a fair comparison. For this reason, when we execute the SDTW algorithm for a given template (line 8), we scale the DTW distances of the subsequences matching that template by the length of the template (line 9).[1] Then, we select the subsequence with the minimum *normalized* DTW distance.

An optional condition may be imposed on the matches if desired: The length of the matching subsequence must be at least some fraction α of the length of the matching template $N^{(k)}$ (in time samples). If a matched subsequence satisfies this length criterion, we save the matching template number, the corresponding sample interval in the test sequence, and the DTW distance in the list of matched subsequences. Then, we set the matched samples of the test sequence $\mathbf{y}$ to ∞ to prevent them from matching again to any template in the SDTW executions that follow (line 12). If the subsequence does not satisfy the length criterion, we do not save it, but still set the samples in $\mathbf{y}$ corresponding to this subsequence to ∞, but *only for the most recently attempted template* to prevent the same subsequence from matching to that template in the DTW executions that follow (line 14). Otherwise, exactly the same subsequence would be matched to the same template in all of the following iterations because SDTW always finds the best matching subsequence and does not contain any randomness. Thus, the available sample range of $\mathbf{y}$ is different for each template and keeps changing as we execute the algorithm. If a subsequence does not satisfy the length criterion, there are two possibilities in the following DTW execution: (1) another subsequence matching the same template at a different location in $\mathbf{y}$ (satisfying or not satisfying the length criterion) may be found; or (2) a subsequence matching another template (at the same or different location) in $\mathbf{y}$ may be found. Therefore, the subsequence that did not satisfy the length criterion for a particular template must be invisible to that template so that the template may match with subsequences at different positions [case (1)], but at the same time visible to other templates to allow finding another match at the same position (because the previous match is ignored) [case (2)]. This procedure is repeated until, for each template, the maximum number of successively available (finite-valued) samples in $\mathbf{y}$ becomes smaller than some fraction α of the length of the template. After this point, it is not possible for any remaining subsequence to satisfy the length criterion even if it matches.

In the MTMM-DTW algorithm, the matched subsequences (associated either with the same or different templates) in $\mathbf{y}$ are not normally allowed to overlap. This restriction may be relaxed to allow some overlap by replacing the sample range $(m_1^{(k^*)} : m_2^{(k^*)})$ in lines 12 and 14 by $(\tilde{m}_1^{(k^*)} : \tilde{m}_2^{(k^*)})$, where $\tilde{m}_1^{(k^*)} = (1-\beta)m_1^{(k^*)} + \beta m_2^{(k^*)}$ and $\tilde{m}_2^{(k^*)} = \beta m_1^{(k^*)} + (1-\beta)m_2^{(k^*)}$, with $\beta \in (0, 1]$ being the ratio of the matched subsequences that are allowed to overlap in the beginning and at the end with other subsequences. If $\beta = 0$, overlapping is not allowed.

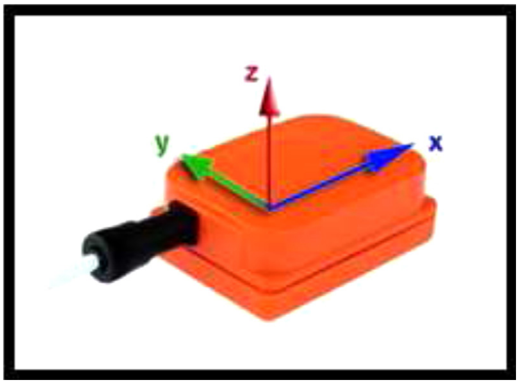


Fig. 3 – Xsens MTx unit [36].

The advantages of the MTMM-DTW algorithm are that: (1) the number of templates, the number of subsequences, their positions on the sample axis, and the length of the test sequence $\mathbf{y}$ need not be known; (2) the templates and the test sequence $\mathbf{x}$ and $\mathbf{y}$ may be multi-dimensional (see Appendix A.1); (3) trivial false matches with a subsequence of length much shorter than the matching template may be eliminated by imposing a condition on length; (4) the amount of overlap between the matched subsequences can be adjusted as desired; and (5) all occurrences of the templates in the test sequence can be detected and classified. The algorithm can be used for several different purposes: classifying a sequence given multiple exercise templates (exercise pattern recognition), detecting the occurrences of single or multiple templates in the test sequence with their time instances and durations, estimating the number of repetitions of all templates in the test sequence (counting the exercises), or all three. Values of the local weights w_h, w_v, and w_d (see Appendix) and the parameters α and β of the MTMM-DTW algorithm can be tuned to the application.

A related algorithm is proposed in [35] to recognize multiple templates in sequential recordings of dance motions. The data set is acquired using a marker-based optical motion capture system. It is assumed that the recordings consist of dance figures that follow one another, where there are neither idle periods nor unmatched intervals. The recording is segmented using the open-end DTW that finds the best-matching template and terminates the segment. Thus, that algorithm is not as flexible as the MTMM-DTW algorithm proposed here.

3. Experimental verification

3.1. *Experimental setup*

We employed wearable motion sensor units to capture the exercises performed by five subjects who were free from movement disorders. This study was approved by Bilkent University Ethics Committee for Research Involving Human Participants and the participants gave informed consent to the work. We used five MTx units (Fig. 3) manufactured by Xsens Technologies [36], each containing three tri-axial devices: an accelerometer, a gyroscope, and a magnetometer. We calibrated the sensors using the system's default calibration

[1] The normalization is approximate because it depends on the step pattern and the values of the local weights, w_h, w_v, and w_d, used in DTW [34], see Eq. (9) in Appendix.

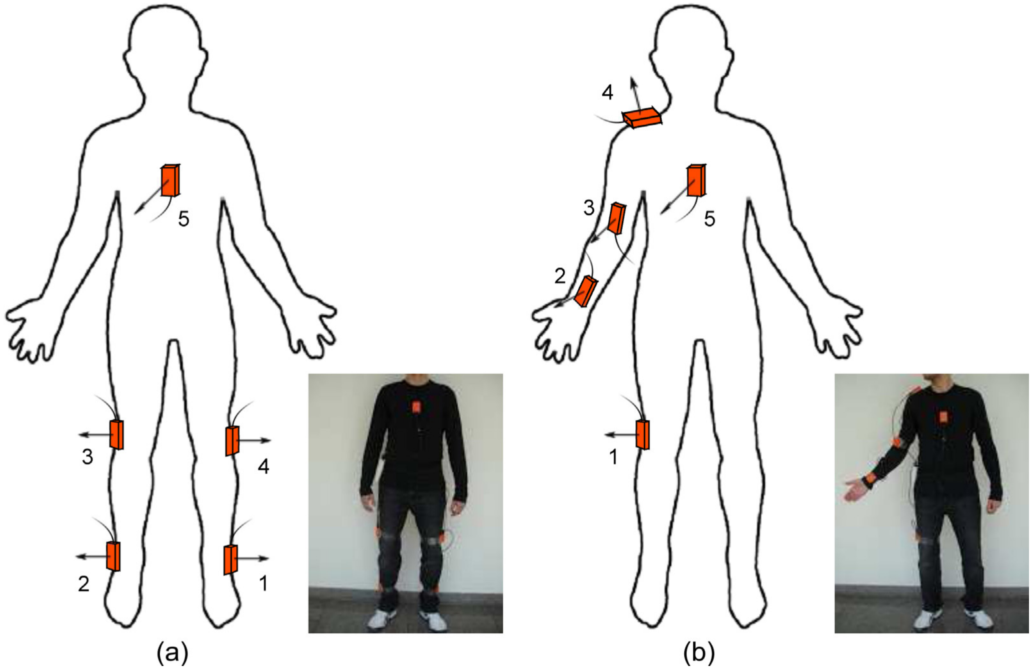


Fig. 4 – Sensor placement on the human body. (a) The first and (b) second configurations are designed for leg and arm exercises, respectively. The arrow perpendicular to the sensor unit indicates the z direction. The cable goes into the sensor unit in the direction of the *x* axis. The y axis can be easily determined given that right-handed coordinate systems are used.

procedure and selected a sampling frequency of 25 Hz. The operating ranges of three of the accelerometers (units 1–3 in Fig. 4) were ±18 g, and two of them (units 4 and 5) were ±5 g, where $g = 9.807\ \mathrm{m/s^2}$ is the gravitational constant. The operating ranges of the gyroscopes and magnetometers were ±1200°/s and ±75 μT, respectively.

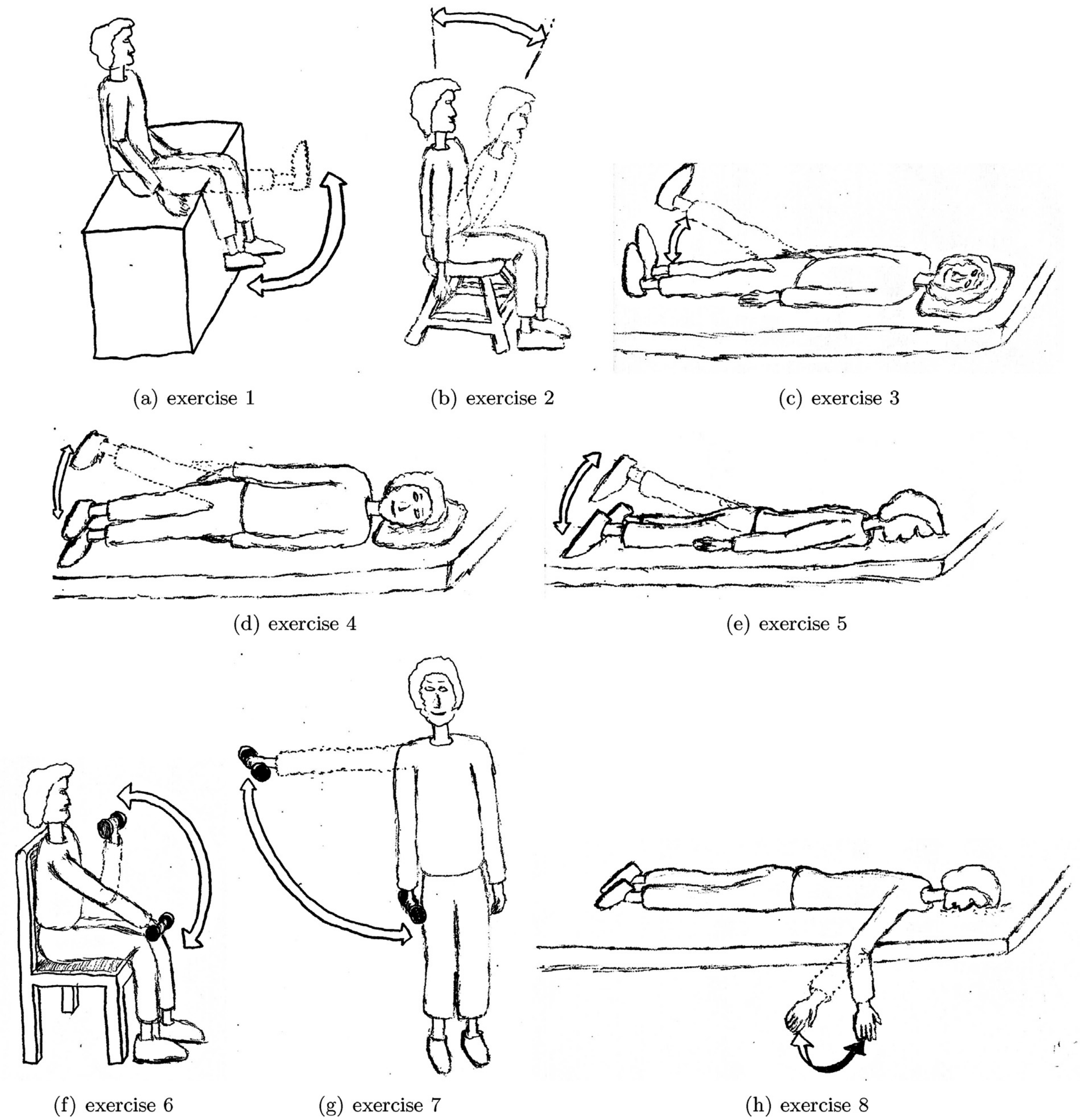
Because the exercises we consider in this study involve only arm or only leg movements, we used two suitably designed sensor configurations to capture these motions (see Fig. 4 for details). The sensor units can be freely configured to properly capture the exercise movements and units can be easily added or removed when needed. The system does not require prior knowledge of the sensor configuration; however, the same configuration should be used while recording the templates and while exercising. The system is highly flexible and modular in these respects, especially when compared with rule-based systems that model the human body to evaluate the exercises.

Because each unit contains three tri-axial devices, 45 (= 9 axes × 5 units) discrete-time sequences were recorded during each experiment. We normalized the acquired sequences separately for each sensor type so that, for example, accelerometer sequences in the whole data set have unit variance. The same holds for the gyroscope and magnetometer sequences.

3.2. *Physical therapy exercises*

The eight exercises we consider in this study were suggested and approved by a physical therapy specialist [37]. They are the most commonly assigned exercises to patients for orthopedic rehabilitation (see Fig. 5). A brief description of each is provided below:

1. Sitting on a high flat surface, raise the right leg, hold for 5 s, keeping the right knee straight, and return to the initial position.
2. Sitting upright on a stool with arms hanging downwards, bend the upper body 30° forward, hold for 5 s, and return to the initial position.
3. Lying flat on the back on a flat surface, raise the right leg from the hip joint, keeping the right knee and left leg straight, hold for 5 s, and return to the initial position.
4. Lying flat on the left side on a flat surface, raise the right leg from the hip joint, keeping the right knee and left leg straight, hold for 5 s, and return to the initial position.
5. Lying facedown on a flat surface, raise the right leg from the hip joint, keeping the right knee and the left leg straight, hold for 5 s, and return to the initial position.
6. Sitting on a chair, holding a 1 kg weight in the right hand, extend the right arm in front of the body to just above the right knee with the palm facing upwards. Bending the elbow joint, raise the weight until the forearm is perpendicular with the thigh, hold for 5 s, and return to the initial position.
7. Standing upright with the right arm holding a 1 kg weight and hanging straight down, raise the weight to the right side from the shoulder joint to a horizontal position while keeping the elbow joint straight, hold for 5 s, then return to the initial position.
8. Lying facedown on a raised flat surface, hang the right arm over the side at the elbow. Raise the right forearm to



(a) exercise 1

(b) exercise 2

(c) exercise 3

(d) exercise 4

(e) exercise 5

(f) exercise 6

(g) exercise 7

(h) exercise 8

Fig. 5 – (a–h) The eight physical therapy exercises, labeled 1–8, considered in this study. In each exercise, the subject moves his leg/arm from the initial position (in solid lines) to the position in dotted lines, holds the position for 5 s, and returns to the initial position.

straighten the elbow, hold for 5 s, then return to the initial position.

3.3. *Experimental procedure*

We designed the experiments to test whether the proposed MTMM-DTW algorithm is able to detect the exercises within a long recording of a typical exercise session while identifying idle time periods, classify each exercise type, estimate the number of repetitions per exercise, determine whether an exercise is performed correctly, and identify the error type, if any.

Two common types of errors that patients make during exercise sessions are:

- Performing the movements too fast; patients do not hold the position for the necessary amount of time because they want to quickly complete the number of repetitions required.
- Performing the exercises at a low amplitude without completely executing the movement. This error may be caused by physical incapability (such as after a stroke) or by negligence, sloppiness, carelessness, etc.

Table 1 – Physical properties of the subjects who participated in the experiments.

Subject	Gender	Age	Weight (kg)	Height (cm)
1	Female	55	73	169
2	Male	61	85	180
3	Male	23	95	180
4	Female	48	55	158
5	Male	53	98	175
Average	60% male	48	81.2	172.4

These errors will be referred to as *type-1* and *type-2 errors*, respectively. Therefore, three execution types are considered in each experiment: (1) correct, (2) with type-1 error, and (3) with type-2 error.

First, all three execution types of each exercise were recorded for each subject while under supervision. These recordings constitute our *reference data set*. Altogether, $K = 24$ (= 3 execution types × 8 exercise types) templates were recorded, denoted by $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$, for each subject. The exercises that a patient performs not under supervision will be compared with these templates. If a patient is unable to perform the exercises properly because his muscles have not sufficiently developed or if he feels too much pain, the specialist can apply external forces to help him execute the exercises [37].

Systematic experiments conducted in the *test phase* simulate a typical real-world exercise session. Each of the five subjects (see Table 1 for their physical properties), performed eight experiments, each corresponding to one of the eight exercise types. A test sequence $\mathbf{y}$ is a recording of a subject performing one of the assigned exercises. The subject repeats the exercise *correctly* 10 times, and then waits until the 100th second of the experiment. While waiting, the subject neither performs an exercise nor moves too much, hence, is considered to be idle. At the 100th second, he starts repeating the same exercise 10 times with a type-1 error, and again waits idly, this time until the 160th second. Then, the subject executes the exercise 10 times with a type-2 error, and the experiment terminates with no more idle time periods. Therefore, an experiment in the test set consists of 10 executions of an exercise for each execution type (30 executions in total), and two idle time intervals of variable duration between the three groups of 10 exercises. Because there are $5 \times 8 = 40$ experiments, each containing 30 executions, the test set contains $5 \times 8 \times 30 = 1200$ exercise executions together with 80 (= 2×40) idle time periods.

Fig. 6 illustrates typical reference and test recordings. Part (a) of the figure shows the outputs of the sensors in unit 2 belonging to the templates for the three execution types. Manually selected templates in a separate recording are highlighted. Part (b) depicts the experiment for exercise 1, performed by the third subject. The two idle time periods are observed between the active periods.

Some executions in the recorded data set may not exactly belong to one of the three execution types. For example, an execution of an exercise with a type-1 error may not be sufficiently fast to be classified into this category, but at the same time it may not be executed sufficiently accurately to be classified as correct; that is, some samples in the data set may not belong to any class. Furthermore, the subjects may not always perform the complete set of exercises properly due to fatigue, lack of concentration or interest, etc. We did our best to reduce these problems by repeating the whole experiment if such an execution error was clearly noticeable.

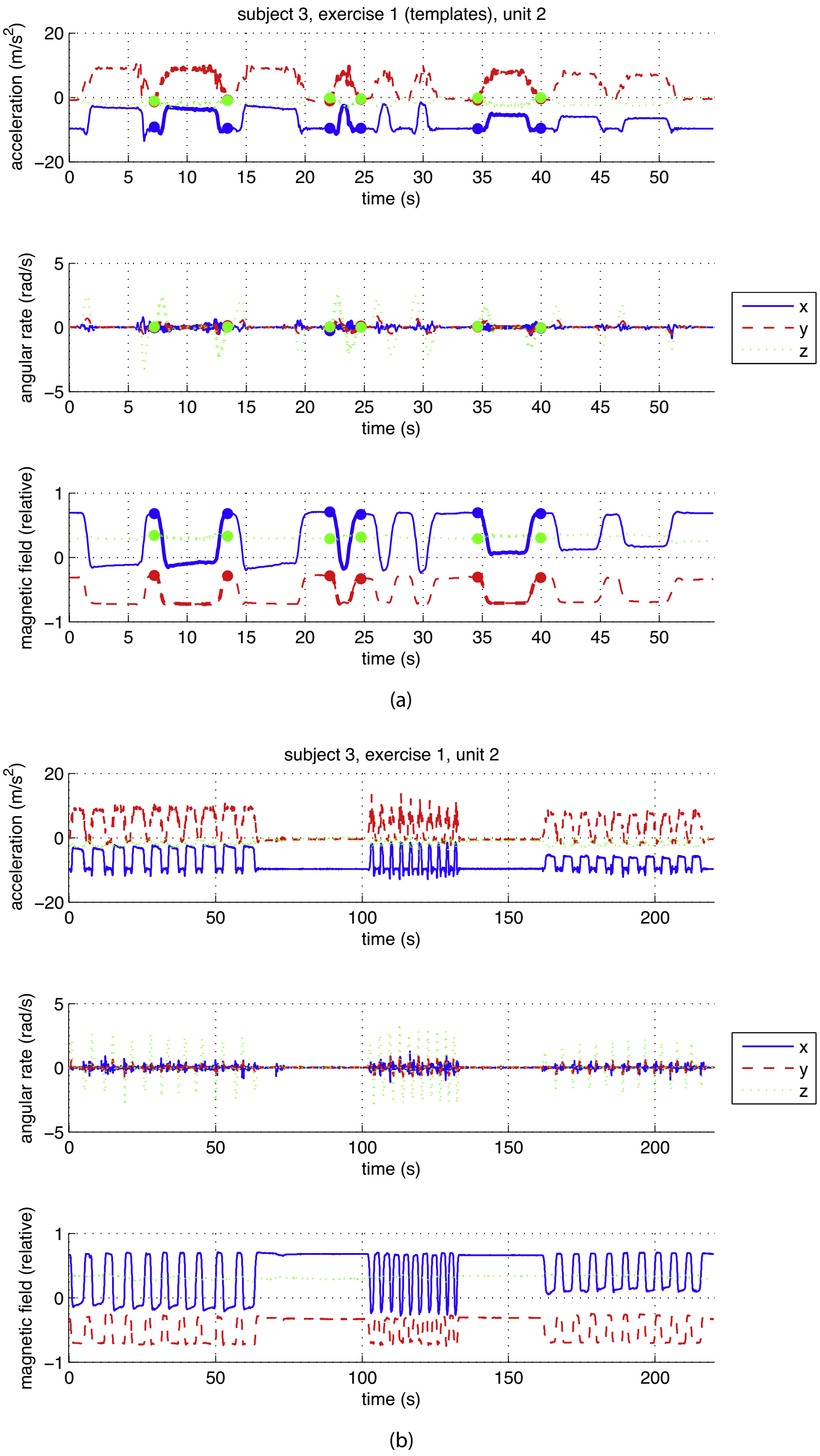
3.4. *Movement detection and classification using MTMM-DTW*

We applied the MTMM-DTW algorithm described in Section 2 to the acquired data set to detect, classify, and evaluate physical therapy exercises in the recorded test sequences. To perform the tests for a given subject, from whom eight test sequences were recorded, we consider the 24 templates recorded from that subject as reference. We used uniform local weights (i.e., $w_h = w_v = w_d = 1$; see Appendix) and selected the parameter $\alpha = 0.5$ to impose the restriction that each matched subsequence is at least half the length of the matching template. We set the β parameter (described in Section 2) to 0.05 to allow the matched subsequences to overlap to up to 5% of their durations in the beginning and at the end. In addition, we discarded matching subsequences with a normalized DTW distance larger than 10 (per sample of the matching template) because they were not sufficiently similar to the matching template.

3.5. *Experimental results*

We evaluated the performance of the proposed MTMM-DTW algorithm in different ways: the percentage of executions the algorithm detected, the algorithm's accuracy in classifying the exercise and/or the execution types, the sensitivity and specificity values, the false alarm (FA) and missed detection (MD) rates. The number of true and false positives and negatives are needed to calculate these quantities [32]. The number of true positives and false negatives can be directly obtained because for these cases, the correct class is true; i.e., there exists an exercise execution. The number of false positives (incorrectly recognizing exercise execution(s) during an idle time interval) is simply the same as the number of FAs. However, the number of true negatives (idle time intervals that are not recognized as an exercise execution) is not directly available because idle time intervals are not countable. The number of samples in the idle intervals can be estimated by dividing the interval's duration by the duration of the correctly executed template of the exercise in each experiment, obtaining the number of negative (idle) samples. Then, the number of true negatives is calculated by subtracting the number of FAs from the total number of negatives.

In Fig. 7, the detected executions in the eight experiments (each actually containing 30 executions of an exercise as well as two idle time intervals), are shown as bars along the time axis for one of the subjects. The widths of the bars indicate the durations of the executions, and the heights show the normalized DTW distance between the executions and the matching template. That is, the shorter the bar, the more similar the matched subsequence and the matching template. Four FAs occur in the idle time intervals: two in the first, one in the fifth, and one in the seventh experiment.



(a)

(b)

Fig. 6 – Recording of the templates and the experiment for exercise 1 performed by subject 3. (a) The three templates (highlighted with thick lines) for correct, type-1 error, and type-2 error execution types of exercise 1, (b) the experiment consisting of 10 repetitions of exercise 1 for the three execution types and two idle time periods in between. Only the sensor outputs of unit 2, the most important one for this exercise, are shown.

Table 2 – Cumulative confusion matrix, which contains the three execution types (A: correct, B: type-1 error, C: type-2 error) of all eight exercises (1–8) summed up for all five subjects. The number of MDs and FAs are shown in an additional column and row, respectively. The elements in the 3 × 3 blocks on the diagonal correspond to correct exercise type classifications.

		Estimated classes																								
		1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C	5A	5B	5C	6A	6B	6C	7A	7B	7C	8A	8B	8C	MD	total
True classes	1A	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	1B	2	39	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	1C	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	2A	0	0	0	44	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	50
	2B	0	0	0	3	45	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	50
	2C	0	0	0	7	4	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	50
	3A	0	0	0	0	0	0	43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	50
	3B	0	0	0	0	0	0	3	43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50
	3C	0	0	0	0	0	0	0	6	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	50
	4A	0	0	0	0	0	0	0	0	0	41	7	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50
	4B	0	0	0	0	0	0	0	0	0	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	4C	0	0	0	0	0	0	0	0	0	5	6	17	0	0	0	0	0	0	0	0	0	0	0	0	22	50
	5A	0	0	0	0	0	0	0	0	0	0	0	0	42	4	1	0	0	0	0	0	0	0	0	0	3	50
	5B	0	0	0	0	0	0	0	0	0	0	0	0	5	21	0	0	0	0	0	0	0	0	0	0	24	50
	5C	0	0	0	0	0	0	0	0	0	0	0	0	5	7	25	0	0	0	0	0	0	0	0	0	13	50
	6A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	2	0	0	0	0	0	0	0	0	50
	6B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	45	0	0	0	0	0	0	0	2	50
	6C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	40	0	0	0	0	0	0	2	50
	7A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	1	0	0	0	0	0	50
	7B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	47	0	0	0	0	0	50
	7C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	1	39	0	0	0	0	50
	8A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	50
	8B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	37	0	7	50
	8C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	2	39	0	50
	FA	4	6	0	0	16	3	1	4	1	1	2	0	9	2	4	5	4	0	1	2	0	6	0	3		74
	Total	56	45	59	54	70	34	47	55	37	48	64	17	61	34	30	63	52	40	63	51	39	71	39	42	103	1274

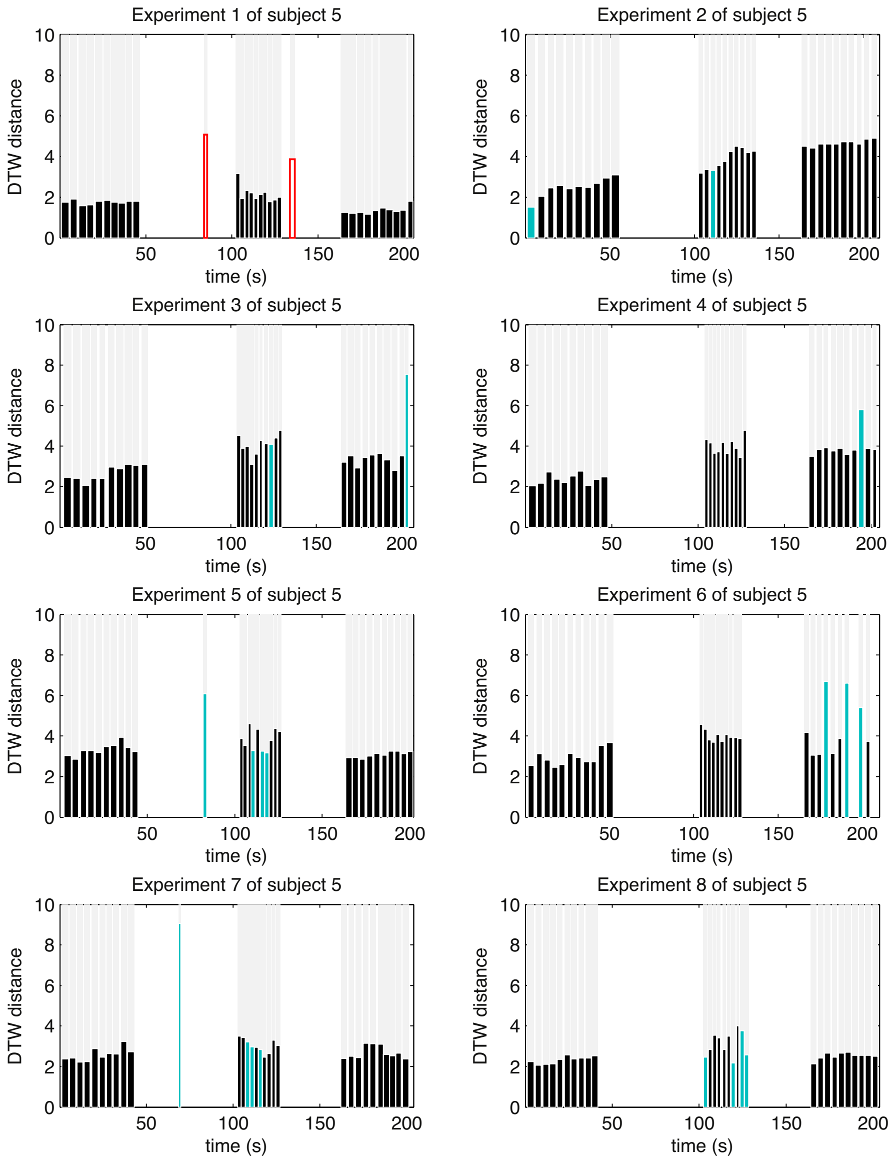


Fig. 7 – Detection and classification of exercise executions performed by subject 5 in the eight experiments corresponding to the eight exercises. Each detected execution is shown as a bar whose width is the execution's duration and whose height is the normalized DTW distance between the detected subsequence and the matching template. Black/gray-filled (black/blue online) bars indicate correctly/incorrectly detected executions of the correct exercise type. Unfilled (red online) bars indicate incorrectly detected exercise types. (The true executions, MDs, and whether the matched subsequences are FAs are not illustrated.)

Table 2 shows the cumulative confusion matrix for the eight exercise types (1–8), each with three execution types (A, B, and C). This matrix is obtained by summing up the confusion matrices of the five subjects. The last column and the last row indicate the number of MDs and FAs in each class, respectively. There are 103 MDs and 74 FAs in total.

Although the proposed system simultaneously classifies the exercise and execution types into 24 classes (1A, 1B, 1C, 2A, . . ., 8C), exercise and execution type classification can be considered separately. The matrix elements corresponding to the correct *exercise type* classifications are emboxed in the confusion matrix. These elements contain correct and incorrect execution type classifications. Combining the three execution types (A, B, and C) of each exercise, the 8 × 8 confusion matrix of the eight exercises is obtained and presented in Table 3. We observe that the system never incorrectly classifies the exercise type that the execution belongs to, but it may miss some executions or detect some additional ones, resulting in MDs and FAs. The true class of the FAs is the negative class, meaning that they do not belong to any of the eight exercise types. However, they have been classified positively as if they do because while the patient remains idle, not performing any exercises, the system may recognize a number of exercises because of the noise on the sequences.

The results are summarized for the five subjects in Table 4 and for the eight exercises in Table 5. The system correctly

Table 3 – Cumulative confusion matrix of all eight exercise types (1–8) summed up for all five subjects. The number of MDs and FAs are shown in an additional column and row, respectively.

		Estimated classes									Total
		1	2	3	4	5	6	7	8	MD	
True classes	1	150	0	0	0	0	0	0	0	0	150
	2	0	139	0	0	0	0	0	0	11	150
	3	0	0	133	0	0	0	0	0	17	150
	4	0	0	0	126	0	0	0	0	24	150
	5	0	0	0	0	110	0	0	0	40	150
	6	0	0	0	0	0	146	0	0	4	150
	7	0	0	0	0	0	0	150	0	0	150
	8	0	0	0	0	0	0	0	143	7	150
	FA	10	19	6	3	15	9	3	9		74
Total		160	158	139	129	125	155	153	152	103	1274

Table 4 – Experimental results summarized per subject.

	Subject					Total
	1	2	3	4	5	
Number of detected executions	194	255	216	216	244	1125
Number of actual executions	240	240	240	240	240	1200
			Percentage (%)			Overall (%)
Accuracy of exercise classification	88.42	93.14	93.17	93.86	97.64	93.46
Accuracy of exercise and execution type classification	80.69	89.29	88.42	88.48	94.62	88.65
Sensitivity	80.00	99.58	88.75	89.17	99.58	91.42
Specificity	97.35	88.81	96.86	97.86	96.33	95.09
MD rate	20.00	0.42	11.25	10.83	0.42	8.58
FA rate	2.65	11.19	3.14	2.14	3.67	4.91

Table 5 – Experimental results summarized per exercise type.

	Exercise type								Total
	1	2	3	4	5	6	7	8	
Number of detected executions	160	142	135	127	115	149	151	146	1125
Number of actual executions	150	150	150	150	150	150	150	150	1200
				Percentage (%)					Overall (%)
Accuracy of exercise classification	97.01	90.71	91.92	91.35	83.07	96.36	99.26	94.84	93.46
Accuracy of exercise and execution type classification	93.54	84.29	87.93	85.67	77.01	92.74	94.73	89.57	88.65
Sensitivity	100.00	92.67	88.67	84.00	73.33	97.33	100.00	95.33	91.42
Specificity	89.15	95.65	96.56	98.29	94.44	95.65	98.83	94.97	95.09
MD rate	0.00	7.33	11.33	16.00	26.67	2.67	0.00	4.67	8.58
FA rate	10.85	4.35	3.44	1.71	5.56	4.35	1.17	5.03	4.91

detects 1125 of the 1200 executions in the data set, undercounting the exercises by −6.25%. As observed in Table 4, the number of detected executions for each subject varies between 194 and 255, where the correct number is 240. Table 5 shows that the number of detected executions ranges between 115 and 160 for each of the eight exercises, where the correct number is 150. The variation in the exercises is caused by the fact that some exercises inherently contain movements of lower amplitude compared to others. For example, in exercises 4 and 5, the leg movements are small because of the difficulty of the exercise, and the system can only recognize 85% and 77% of the exercises, respectively. Considering that the algorithm tries to detect the correct and erroneous movements of two types (executed quickly or at a low amplitude), it is more difficult for it to recognize executions performed at a low amplitude. For this reason, not only does the number of MDs increase, but also the number of FAs, because the templates that belong to the low-amplitude executions are more similar to the sequences in the idle time intervals.

The overall accuracy of the system in exercise classification only is 93.46%, whereas in simultaneous exercise and execution type classification it is 88.65%.[2] For the former, FAs and MDs decrease the accuracy, whereas for the latter, in addition to FAs and MDs, incorrect execution type classifications also decrease the accuracy. In Tables 4 and 5, we observe that the performance of the system varies considerably among the subjects and the exercises. The accuracy in exercise and execution type classification varies between 80.69% (subject 1) and 94.62% (subject 5) for the five subjects, and 77.01% (exercise 5) and 94.73% (exercise 6) for the eight exercises. The

[2] In calculating the accuracy values, we consider both the exercise executions whose true class is positive and the idle time intervals whose true class is negative.

accuracy of the system can be further improved by tuning the parameters (such as α and β in the MTMM-DTW algorithm and the local weights in the DTW algorithm) for specific exercise types.

The overall sensitivity and specificity values are 91.42% and 95.09%, respectively, and the overall MD and FA rates are 8.58% and 4.91%, respectively, for the whole data set (Tables 4 and 5).

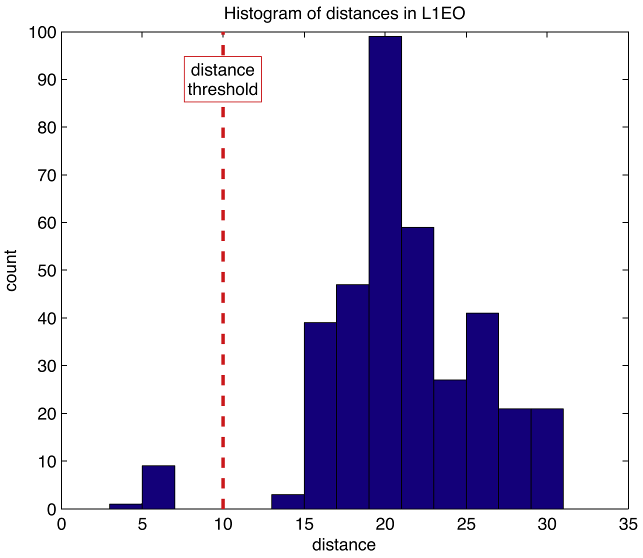
3.6. *Testing the system with unknown exercise movements*

In the proposed system, we consider three execution types of each exercise, two of them being the most common types of incorrect executions [28,37]. However, the patient may perform the exercise in an incorrect way that may not belong to one of these categories, or he may perform a completely different activity between exercises, such as repositioning his body or stretching. When an unknown movement is performed, the system is ideally expected not to detect any exercise execution or detect one with a relatively large DTW distance that can be easily discarded by simple thresholding. Since there is a wide variety of movements to consider and the output depends on how different the unknown movements are from the recorded templates, it is rather difficult to test this aspect of the system.

To validate the robustness of the proposed system to unknown movements or execution types, we use the *leave-one-exercise-out* (L1EO) cross-validation approach: We execute the MTMM-DTW algorithm by leaving out the three templates (correct, type-1, and type-2 errors) of one type of exercise at a time for each subject, and comparing the recorded test sequence of that exercise (with 30 executions) with the templates of the remaining seven exercise types of that subject (21 templates). As mentioned above, ideally, the system should not detect any executions. Note that in the proposed system (see Section 3.4), we discard matching intervals with normalized DTW distances larger than 10. Using the same threshold value of 10 in L1EO and considering each detection with a DTW distance below this level as an FA results in 10 false alarms out of the 1200 executions in the data set. The very low FA rate (0.83%) demonstrates the robustness of the proposed system to unknown movements. The number of FAs is tabulated in Fig. 8(a) for each exercise of each subject. The histogram of the normalized DTW distances of the detections is provided in Fig. 8(b). The threshold level of 10 separates the histogram into two distinct parts and the distance values nearest to the threshold (5.44 and 14.88) are relatively far from each other and from the threshold, indicating that the number of FAs is not very sensitive to the threshold level.

		exercise								
	FAs	1	2	3	4	5	6	7	8	total
subject	1	0	0	0	0	0	0	0	0	0
	2	8	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0
	5	2	0	0	0	0	0	0	0	0
	total	10	0	0	0	0	0	0	0	10

(a)



(b)

Fig. 8 – (a) Number of FAs for each exercise of each subject in L1EO in tabular form. (b) Histogram of the normalized DTW distances of all the detections in L1EO. Detections with DTW distances below the threshold are considered FAs.

3.7. *Computational complexity*

The computational complexity of the DTW and the SDTW algorithms is directly proportional to the product of the lengths of the two sequences [42]. This is also true for the STMM-DTW and MTMM-DTW algorithms because both are based on the SDTW algorithm. The computational complexity of the MTMM-DTW algorithm is also directly proportional to the number of templates. However, the algorithm repeats a particular process until a condition is satisfied; thus, its computational complexity differs for different sequences of the same length. We implemented the MTMM-DTW algorithm on a laptop with a quad-core processor at 2 GHz (Intel Core i7 2630QM) and with 8 GB of RAM, running 32-bit MATLAB. Although the calculations are completed with some delay because the computational complexity of the proposed algorithm is somewhat high, the algorithm can be modified to run in near real time. The computational efficiency would increase by about 200 times if the SDTW algorithm were programmed in C. It is demonstrated in [38] that if a graphical processing unit (GPU) is used in addition to a central processing unit (CPU), the running time further decreases, up to 29 times. When a field programmable gate array (FPGA) is used instead of a PC, the SDTW algorithm would run up to 4500 times faster than its version programmed in C, which makes real-time implementation possible even on a low-cost PC or a portable device [38]. A real-time implementation of a classification scheme, based on the *ordinary* DTW, is presented in [39], where the classifier performance is evaluated based on experimental data.

Since the computational complexity of the MTMM-DTW algorithm is directly proportional to the number of templates used, to further improve the computational efficiency, the number of reference templates can be reduced by removing the two incorrect execution types of the exercises.

4. Discussion

The methodology proposed here does not require previously defined rules and is flexible and adaptable because (1) there are no restrictions on the number of exercise executions, number of templates, or durations of the sequences; (2) the executions may be allowed to overlap with each other at a desired level; (3) each of the template sequences may be of any length because the DTW distance is normalized by the template duration; and (4) the sequences may be one- or multi-dimensional.

The main advantages of the proposed system are the following: (1) The patient does not need to press a button to indicate the beginning or the end of the exercise executions, wait for a signal to start an execution, or select the exercise type he intends to perform. This feature makes the system usable in a physical therapy session of any duration, theoretically consisting of an unlimited number of executions. (2) There is no need for the system to be configured for different exercise types or patients; it does not require information on the physical properties of the subjects nor a set of rules to define the exercise types, but only needs a recording of the template executions of each exercise performed by each patient. (3) The system works independently of the number, type, and configuration of the wearable sensors; these can be chosen as needed. Hence, any sensor configuration that can properly capture the movements can be used. The only requirement is that the configuration must remain the same when recording the template sequences and when using the system. It is easy to add a new exercise type, a new sensor, or for a non-expert to change some or all of the sensors; i.e., the whole system does not have to be re-programmed for such modifications.

The last feature represents a significant advantage over the many studies that employ 3D human body models to evaluate exercises [5,18,24], where the number, types, positions, and orientations of the sensors on the body are pre-determined. When this is the case, the system must be modified considerably if a change in sensor configuration occurs. In addition, these types of systems often require rule-based algorithms (or decision trees) to evaluate the correctness of the exercise executions, which makes them extremely difficult to reconfigure for newly added exercise types. However, in our system, if a new exercise needs to be added, the only requirement is to record the templates of the patient performing the different execution types of that exercise, which the physician/therapist can easily do instead of having to rely on the engineer who developed the system. Considering the wide variety of physical therapy exercises assigned for different disorders, the unsupervised nature of our algorithm allows flexibility in exercise type and in hardware type and configuration. In the experiments, we observed that this autonomous methodology causes rare MDs and FAs, which are tolerable considering that the system is much easier to use compared to the systems developed in studies such as [10].

Ours is a very early proof-of-principle study in the pre-clinical stage and the methodology has not yet been demonstrated in clinical settings or to be clinically efficacious. The primary applicability of the system is for patients undergoing orthopedic rehabilitation who periodically repeat the prescribed exercises in the same way during the rehabilitation process. Neurological rehabilitation usually relies on one-to-one interaction between the patient and the physician/therapist. The assigned exercises are more directly focused on improving functional activities (e.g., grasping or squeezing an object, holding a cup) and more often involve application of forces to the patient's body manually or using robotic devices. However, such patients may also benefit from the system proposed here. While they may not be able to completely execute the exercise movements at first, they will likely progress, and as they do so, the normalized DTW distance between the motion performed and its correct form may be used as a measure of discrepancy or improvement. For patients in this second group, the system can be modified as follows to keep track of their progress during rehabilitation: Instead of recording only one correct template for each exercise that represents the final goal (and possibly some erroneous templates), templates that correspond to intermediate goal points (such as partial, low-amplitude, or slow executions of an exercise) may be recorded. The templates may be updated as needed according to the patient's progress; for instance, some intermediate templates may be removed over time. This way, the progress of the patient can be monitored quantitatively to provide feedback to the patient and the physician/therapist. Not having to press a button can also be advantageous for patients undergoing neurological rehabilitation, especially those with upper extremity problems. Even healthy patients may find it difficult to press the button in a timely manner which may degrade the system performance.

The system can provide feedback at any desired level of detail in the form of a progress report (e.g., the total number of executions the patient performs, the percentage of correct executions, the accuracy of the executions, whether the patient reached the targeted number of exercises, and the active/idle time intervals in the physical therapy session). To save time, feedback can also be in the form of alerts that inform physicians/therapists only when needed, for example, when the activity level of the patient is too low, the majority of the executions are incorrect or too fast, etc. Providing feedback in this form will take much less time of the physician than having to monitor each patient on an individual basis.

If an exercise is performed too quickly or too slowly, the detected execution will be shorter/longer compared to the reference template, respectively, which will naturally affect the resulting cost (the DTW distance) in the algorithm. If the exercise is performed too slowly, the DTW distance is calculated based on comparing a larger number of samples of the template and test sequences, and the resulting DTW distance will be slightly larger due to differences in the movement and noise. Since we choose the values of the horizontal and diagonal local weights (w_h and w_d) as the same, we do not additionally penalize long matches and do not impose any limit on the length of the matching subsequence. Therefore, slower executions can also be detected using this scheme.

If the exercise is performed too quickly, the matching subsequence will be short, and will be ignored if shorter than $\alpha N^{(k)}$, where $N^{(k)}$ is the length of the kth template, $k = 1, \ldots, K$.

By setting $\alpha = 0.5$, we discard matches shorter than $N^{(k)}/2$, as stated in Section 2. The value of α can be decreased further to allow even shorter matches. (Obviously, α should be set between $1/N^{(k)}$ and 1: when it is $1/N^{(k)}$, we allow matching subsequences with a minimum length of 1; that is, we allow all matches. When it is 1, we discard matching subsequences shorter than the length of the template.) However, if α is selected too small (say, less than 0.2), then the template can be matched to subsequences much shorter than itself, which are likely not to contain any exercise execution. This situation may increase the FA rate because the sequences contain noise, and some subsequences of a recording may resemble one of the templates as a result of warping. We observed that $\alpha = 0.5$ is a suitable choice for our data set because exercises should not be executed too quickly and because we have templates recorded for exercises performed at a faster pace. Furthermore, to accomodate shorter matches, the local weight, w_v, for the vertical direction can be decreased. Thus, the DTW algorithm is sufficiently flexible in the sense that the amount and direction of warping of the sequences can be adjusted as needed.

Note also that the inertial sensor recordings of an exercise execution at a faster pace may not simply be the time-contracted versions of the sequences of an execution at regular pace. For instance, a gyroscope signal provides the angular rate of change of some part of the body, and its amplitude may also increase when the exercise is performed faster. Therefore, it may not be trivial to match very quick or slow executions to the template of a normal execution. When the purpose is to detect exercises performed at an unusual pace, it may be better to include templates for such executions in the system, as is done in this work for faster executions.

When the patient needs to use the system while not under supervision, the sensors must be fixed to the same positions on his body as when the template sequences were recorded. As part of their training, patients must be taught to wear the sensor set properly in all environments. The positions and orientations could be marked on the body using permanent markers. Illustrative/descriptive brochures may be also provided to guide sensor placement. Another possibility is to fix the sensors to a garment that fits the patient's body tightly to make sure that they will always be worn in the same way.

A possible modification to the proposed methodology would be to omit recording and including the templates of incorrect executions. In this case, one would need to record only a single template for each exercise type, corresponding to the correct execution of that exercise. The amount of DTW error can be used as a measure of the discrepancy between the patient's execution and the reference template of the correct execution, and provide alerts when necessary. An execution can be classified as the exercise type whose template matches with the smallest DTW distance. Incorrect executions would be matched to the most-similar correctly executed template with some error. In this case, the degree of correctness of the matched executions would be determined by thresholding the corresponding normalized DTW distance. If the distance is below the threshold, the execution is sufficiently similar to the template, and the execution would be considered correct. If this is not the case, then the execution is not sufficiently similar to the template, and it would be classified as incorrect. This approach would also have a lower computational cost, as mentioned in Section 3.7.

Other techniques such as the Wii [40,41] can be used in physical therapy. However, the aim of studies utilizing that technology is quite different than the purpose of our system. Wii systems are mostly based on virtual reality, where the patient executes some movements while holding the Wii controller itself or a tool that contains it, such as a tennis racket, to attain certain goals in a game-like environment by controlling a character or an object, which is displayed on the screen in real time. The objectives are selected specific to the therapy. Hence, in Wii systems, the patient is not guided to perform well-defined movements; rather, he performs some movements himself to attain the described goal. In addition, a Wii system has only a single controller containing motion sensors; thus, it cannot detect body posture or multiple parts of the body as our system does. For this reason, the patient may complete the process in an undesired manner; for instance, he may satisfy the goals while sitting instead of standing, or (if the purpose was to strengthen an injured hand) use his healthy hand. Such exploitations are not allowed in our system that uses multiple sensors. Wii systems are inexpensive and easily accessible to anyone with standard TV connections, and our methodology can be integrated with a Wii system or another game console if necessary. Modern game consoles are readily connected to the internet and TV with audio and our MTMM-DTW algorithm integrated with a game console may be used to provide visual or voice feedback to the patient. The results can also be sent to the specialist. Considering that GPU implementation of DTW is much faster than the CPU implementation [38], and that modern game consoles have fast GPUs, the hardware of most game consoles has sufficient computational power to run the MTMM-DTW algorithm.

5. Summary and conclusions

In this study, we address an important problem in physical therapy: automatically detecting the individual executions of assigned exercises during a physical therapy session, classifying them, and evaluating their correctness/accuracy to provide feedback to the patient and the specialist. We propose the MTMM-DTW algorithm for this purpose, based on the DTW dissimilarity measure, to detect the occurrences of multiple templates in the recording of a physical therapy session.

To evaluate the algorithm's performance, we conduct experiments involving eight types of exercises performed by five subjects and acquire a data set with a total of 120 and 1200 exercise executions in the reference and test sets, respectively. The accuracy is 93.46% for exercise classification only and 88.65% for simultaneous exercise and execution type classification. The algorithm misses 8.58% of the performed executions and detects 4.91% executions in excess, caused by idle time intervals. To test the robustness of the system to unknown exercises, we employ a leave-one-exercise-out cross-validation strategy. This results in a false alarm rate of 0.83%, demonstrating the robustness of the system to unknown movements.

Considering these outcomes, the performance of the proposed system is acceptable, especially in counting and classifying the exercises. This is valuable because manual counting of a patient's executions can be a tedious process. Furthermore, automatic and objective evaluation of the exercise session has merit because direct observation may provide subjective feedback [7,8]. Our system can provide feedback at any desired level of detail to the patient and the physician/therapist.

This algorithm is quite flexible and highly adaptable for different schemes because the flexibility allowed in the sequences, the sensitivity in detecting the occurrences, the amount of overlap allowed between the occurrences, and the length of the matched subsequences are all adjustable. The method may be potentially applied to other areas, such as recognizing quasi-periodic daily or sports activities.

In future work, we plan to extend our study to patients undergoing orthopedic rehabilitation to demonstrate our methodology in clinical settings. The algorithm may be modified to work in near real time so that the patient receives feedback after each exercise execution within a reasonable time frame. The system parameters can be optimized individually for each exercise type. Other sensor modalities may be used or combined with motion sensors, such as radio-frequency (RF) localization, to directly acquire position information instead of using inertial sensors that provide rate information.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank physical therapy specialists Assoc. Prof. Dr. İlknur Tuğcu and Dr. Özlem Yoleri for providing the descriptions of the exercises and for providing background information on the needs of patients and specialists in physical therapy. The authors are also grateful to the five volunteers who participated in our experiments for their efforts, dedication, and time.

Appendix A. Dynamic time warping

The ordinary DTW algorithm matches two discrete-time sequences (represented as vectors $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_M]^T$) by elastically transforming their time (or sample) axes such that they are most similar to each other. To quantize similarity, a *local distance (cost) measure* can be defined between two samples x_n and y_m as

$$c(x_n, y_m) : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}, \tag{1}$$

where $\mathcal{F}$ is the feature space such that $x_n, y_m \in \mathcal{F}\ \forall n, m$ [42]. The more similar are x_n and y_m, the smaller is the cost function $c(x_n, y_m)$. In this paper, we select the local cost as the square of the distance between x_n and $y_m, c(x_n, y_m) = (x_n - y_m)^2$, as is usually done.

To find the optimal match between $\mathbf{x}$ and $\mathbf{y}$, one can calculate the *cost matrix* $\mathbf{C}$ of size $N \times M$ between each pair of elements of x_n and y_m as

$$\mathbf{C} = [C_{n,m}] = [c(x_n, y_m)] \tag{2}$$

and find the *optimal warping path* in the cost matrix $\mathbf{C}$ with the smallest cumulative cost.

A warping path can be represented with the sequence $\boldsymbol{p} = (p_1, p_2, \ldots, p_L)$, where

$$p_l = (n_l, m_l) \in [1:N] \times [1:M], \quad 1 \leq l \leq L. \tag{3}$$

There are three basic conditions for the warping path [42]:

1. *Boundary condition*: The path starts from the very first element of the cost matrix and ends at the very last element; i.e., $p_1 = (1, 1)$ and $p_L = (N, M)$.
2. *Monotonicity condition*: The path can proceed to the right, to the bottom, or to the direction in between (bottom-right), but it cannot return back; i.e., $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$.[3]
3. *Step-size condition (continuity)*:[4] The path can proceed to the neighbor element at the right, at the bottom, or at the bottom-right; i.e.,

$$p_l \in \{p_{l-1} + (0, 1), p_{l-1} + (1, 0), p_{l-1} + (1, 1)\}, \quad 2 \leq l \leq L. \tag{4}$$

The *total (cumulative) cost* of a warping path $\boldsymbol{p}$ between the sequences $\mathbf{x}$ and $\mathbf{y}$ is defined simply as the sum of the local costs of the matched elements of $\mathbf{x}$ and $\mathbf{y}$:

$$C_{\boldsymbol{p}}(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{L} c(x_{n_l}, y_{m_l}). \tag{5}$$

Then, the *optimal warping path* $\boldsymbol{p}^*$ is the path having the minimum total cost among all warping paths between $\mathbf{x}$ and $\mathbf{y}$ satisfying the path conditions:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}} C_{\boldsymbol{p}}(\mathbf{x}, \mathbf{y}). \tag{6}$$

The *DTW distance* between $\mathbf{x}$ and $\mathbf{y}$ is then defined as the total distance of the optimal warping path:

$$\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = C_{\boldsymbol{p}^*}(\mathbf{x}, \mathbf{y}) = \min_{\boldsymbol{p}} C_{\boldsymbol{p}}(\mathbf{x}, \mathbf{y}). \tag{7}$$

Unlike its name, the DTW distance does *not* satisfy the triangle inequality even if the cost function is a metric, and hence the DTW distance is not a metric [42].

Instead of an exhaustive search for all possible warping paths between $\mathbf{x}$ and $\mathbf{y}$, which would be extremely inefficient, one can use an algorithm with computational complexity

[3] The indices n and m of the $\mathbf{C}$ matrix increase downwards and from left to right, respectively.

[4] See Ref. [42] for the different choices for the step-size condition.

$\mathcal{O}(NM)$ based on dynamic programming. For this purpose, the *accumulated cost matrix* of size $N \times M$ is defined as

$$\mathbf{D} = [D_{n,m}] = [\mathrm{DTW}(\mathbf{x}_{1:n}, \mathbf{y}_{1:m})], \tag{8}$$

where $\mathbf{x}_{1:n} = [x_1, x_2, \ldots, x_n]$ and $\mathbf{y}_{1:m} = [y_1, y_2, \ldots, y_m]$ are the prefixes of the sequences $\mathbf{x}$ and $\mathbf{y}$ with lengths n and m, respectively, for $1 \le n \le N$ and $1 \le m \le M$. Obviously, $D_{N,M}$ is the desired DTW distance; i.e., $\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = D_{N,M}$. The accumulated cost matrix $\mathbf{D}$ can be computed relatively efficiently with the following equations [42]:

$$\begin{aligned} D_{n,1} &= \sum_{i=1}^{n} c(x_i, y_1) \\ D_{1,m} &= \sum_{i=1}^{m} c(x_1, y_i) \qquad 1 \le n \le N, \quad 1 \le m \le M \\ D_{n,m} &= \min \begin{cases} D_{n-1,m} + w_h c(x_n, y_m) \\ D_{n,m-1} + w_v c(x_n, y_m) \\ D_{n-1,m-1} + w_d c(x_n, y_m). \end{cases} \end{aligned} \tag{9}$$

Here, w_h, w_v, and w_d are the local weights for the horizontal, vertical, and diagonal directions, respectively, with default values of one. If one needs to discourage advancing in a certain direction in warping the sequences, the local weight for that direction can be chosen to be larger than one.

Note that all elements of $\mathbf{D}$ must be calculated to obtain the very last element $D_{N,M}$, which is the DTW distance. By using this method, the DTW distance is calculated *without* explicitly finding the optimal warping path $\boldsymbol{p}^*$. Using $\mathbf{D}$, $\boldsymbol{p}^*$ can be calculated by initializing $p_L^* = (N, M)$ and progressing in the reverse order: If p_l^* is computed, p_{l-1}^* is calculated as

$$p_{l-1}^* = \begin{cases} (1, m-1) & \text{if } n = 1 \\ (n-1, 1) & \text{if } m = 1 \\ \arg\min\{D_{n-1,m}, D_{n,m-1}, D_{n-1,m-1}\} & \text{otherwise.} \end{cases} \tag{10}$$

The optimal path $\boldsymbol{p}^*$ terminates with $p_1^* = (1, 1)$. In other words, starting at the bottom-right element $D_{N,M}$, the optimal warping path steps into the next-smallest neighbor element in $\mathbf{D}$ (proceeding only to the left, top, or top-left) and finally ends up with the top-left element $D_{1,1}$.

A.1. *Multi-dimensional sequences*

Note that one can compute the DTW distance and the optimal warping path of two sequences $\mathbf{x}$ and $\mathbf{y}$ by knowing only the cost matrix $\mathbf{C}$, which can be computed by using the local cost function $c(x_n, y_m)$ defined on every sample x_n and y_m of $\mathbf{x}$ and $\mathbf{y}$. In the case of *multi-dimensional sequences* $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$, the local cost $c(\underline{x}_n, \underline{y}_m)$ can be defined to handle this such that its range is scalar, as before. Then, the DTW and the optimal warping path can be calculated in exactly the same way as before for one-dimensional sequences. In this case, the same warping is applied to all dimensions of $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$, considering the overall similarity between $\underline{\mathbf{x}}$ and $\underline{\mathbf{y}}$.

A.2. *Free endpoints*

For some applications, the sequences to be matched are cropped manually and hence may naturally contain a prefix or a suffix that contains no valuable or meaningful information. Then, the sequences should be matched to each other with some unmatched parts at the beginning or at the end of *one* of the sequences. From now on, the sequences $\mathbf{x}$ and $\mathbf{y}$ are called the *template* and the *test* sequences, respectively, assuming the test sequence may contain prefix and/or suffix parts, whereas the template sequence does not. This assumption is valid in pattern classification and pattern search problems, where there are typically several template sequences obtained for this purpose that do not contain any undesired parts, but there are many test sequences that may contain additional parts at the beginning and/or at the end [42].

Using the ordinary DTW algorithm, the prefixes and/or suffixes of the sequences will cause an additional undesired cost, increasing the DTW distance. A better approach would be to modify the ordinary DTW algorithm to allow *free endpoints* so that the prefix and/or suffix parts of the test sequence $\mathbf{y}$ can be ignored [43]. The length of the ignored parts are selected optimally in the sense that the DTW distance between $\mathbf{x}$ and the matched subsequence of $\mathbf{y}$ is minimized [4,42]. If both endpoints are free, the algorithm is called *subsequence DTW* (SDTW) [42] or *open-begin open-end DTW* (OBE-DTW) [4], and the resulting distance is

$$\mathrm{SDTW}(\mathbf{x}, \mathbf{y}) = \min_{m_1, m_2} \mathrm{DTW}[\mathbf{x}, \mathbf{y}(m_1 : m_2)], \tag{11}$$

where $\mathbf{y}(m_1 : m_2) = [y_{m_1}, y_{m_1+1}, \ldots, y_{m_2}]$ is the subsequence of $\mathbf{y}$, with $1 \le m_1 \le m_2 \le M$. The m_1 and m_2 values minimizing the DTW distance, namely m_1^* and m_2^*, determine the (optimal) matched subsequence of $\mathbf{y}$ to $\mathbf{x}$. If m_1 is set to 1, only the suffix of $\mathbf{y}$ is excluded. Similarly, if $m_2 = M$, only the prefix of $\mathbf{y}$ is excluded. Obviously, using $m_1 = 1$ and $m_2 = M$ results in the ordinary DTW algorithm.

To allow free endpoints, the ordinary DTW algorithm needs to be modified as follows:

- *open-begin DTW*: To exclude the prefix of $\mathbf{y}$, the first boundary condition of the warping path is extended so that $p_1 = (1, m_1)$ with $1 \le m_1 \le M$, allowing the beginning point of the warping path to reside anywhere in the first row of the accumulated cost matrix $\mathbf{D}$. Thus, the first $m_1 - 1$ samples of $\mathbf{y}$ are ignored. To this end, the first modification in the ordinary DTW algorithm is in the calculation of the first row of the accumulated cost matrix $\mathbf{D}$:

$$D_{1,m} = c(x_1, y_m). \tag{12}$$

  In this way, the first row of $\mathbf{D}$ consists of the costs of matching x_1 to each element of $\mathbf{y}$ instead of to the accumulated costs. The open-begin DTW distance is $D_{N,M}$, as before. The second modification is in the calculation of the optimal warping path. The optimal path $\boldsymbol{p}^*$ is calculated in the reverse order, as before; however, this time the process terminates when the first row of $\mathbf{D}$ is reached; i.e., when $p_1^* = (1, m')$.

Here, the column where the process ends is the first sample of the matched part of **y**: $m_1^* = m'$.

- *open-end* DTW: To exclude the suffix of **y**, the second boundary condition of the warping path is extended so that $p_L = (N, m_2)$, with $1 \le m_2 \le M$, allowing the end point of the warping path to reside anywhere in the last row of the accumulated cost matrix **D**. Thus, the last $M - m_2$ samples of **y** are ignored. For this purpose, the ordinary DTW algorithm is executed, but this time the DTW distance (allowing exclusion of the suffix of **y**) is the minimum element in the last row of the accumulated cost matrix **D** instead of the last element $D_{N,M}$:

$$\mathrm{DTW}_{\text{open end}}(\mathbf{x}, \mathbf{y}) = \min_{m_2} D_{N,m_2}. \tag{13}$$

Here, the optimal value m_2^* of m_2 determines the exclusion of the last $M - m_2$ elements of **y**, simply ignoring the rest of **D** on the right side. Obviously, the optimal warping path is now initialized as $p_L^* = (N, m_2^*)$ instead of $p_L^* = (N, M)$ from the end. The suffix of **y** is likely to be ignored with these two modifications in the ordinary DTW algorithm.

- *subsequence DTW*: To allow the exclusion of both the prefix and the suffix of **y**, the two modifications to exclude the prefix (in calculating the first row of **D** and the new ending criterion in calculating $\boldsymbol{p}^*$) and the two modifications to exclude the suffix (in obtaining the DTW distance from **D** and in the starting point of $\boldsymbol{p}^*$) explained above need to be made in the ordinary DTW algorithm. Then, the SDTW algorithm is obtained.

Note that, as in the ordinary DTW, the SDTW algorithm is able to handle multi-dimensional sequences and local weights. Surprisingly, the computational complexity of the SDTW algorithm is $\mathcal{O}(NM)$, which is exactly the same as the ordinary DTW, although there are two additional parameters (m_1 and m_2) that are jointly optimized to minimize the DTW distance.

REFERENCES

[1] N.U. Ahamed, K. Sundaraj, R.B. Ahmad, SAM M. Rahman, Biosensors assisted automated rehabilitation systems: a systematic review, Int. J. Phys. Sci. 7 (January (1)) (2012) 5–17.

[2] B.G. Steele, B. Belza, K. Cain, C. Warms, J. Coppersmith, J. Howard, Bodies in motion: monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease, J. Rehabil. Res. Dev. 40 (September–October (5 Suppl. 2)) (2003) 45–58.

[3] M. Milenkovic, E. Jovanov, J. Chapman, D. Raskovic, J. Price, An accelerometer-based physical rehabilitation system, in: Proc. 34th IEEE Southeastern Symposium on System Theory, Huntsville, AL, U.S.A., 18–19 March, 2002, pp. 57–60.

[4] P. Tormene, T. Giorgino, S. Quaglini, M. Stefanelli, Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation, Artif. Intell. Med. 45 (January (1)) (2009) 11–34.

[5] P. Fergus, K. Kifayat, M. Merabti, A. Taleb-bendiab, A. El Rhalibi, Remote physiotherapy treatments using wireless body sensor networks, in: Proc. Int. Conf. Wireless Communications and Mobile Computing: Connecting the World Wirelessly, New York, U.S.A., 21–24 June, 2009, pp. 1191–1197.

[6] J.M. Winters, Y. Wang, J.M. Winters, Wearable sensors and telerehabilitation, IEEE Eng. Med. Biol. 22 (May–June (3)) (2003) 56–65.

[7] F. Pitta, T. Troosters, V.S. Probst, M.A. Spruit, M. Decramer, R. Gosselink, Quantifying physical activity in daily life with questionnaires and motion sensors in COPD, Eur. Respir. J. 27 (May (5)) (2006) 1040–1055.

[8] C. Stéphane, H. Mathieu, B. Patrick, Accelerometer-based wireless body area network to estimate intensity of therapy in post-acute rehabilitation, J. NeuroEng. and Rehabil. 5 (September (1)) (2008) 20–31.

[9] J. Brutovsky, D. Novak, Low-cost motivated rehabilitation system for post-operation exercises, in: Proc. 28th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, New York, U.S.A., 30 August–3 September, 2006, pp. 6663–6666.

[10] I. Raso, R. Hervás, J. Bravo, m-Physio: personalized accelerometer-based physical rehabilitation platform, in: Proc. Fourth Int. Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies, Florence, Italy, 25–30 October, 2010, pp. 416–421.

[11] P.E. Taylor, G.J.M. Almeida, T. Kanade, J.K. Hodgins, Classifying human motion quality for knee osteoarthritis using accelerometers, in: Proc. 32nd Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina, 31 August–4 September, 2010, pp. 339–343.

[12] M. Kirilly, Sensor-Based Skill Assessment for Health and Fitness Applications, M.Sc. Thesis, Dept. of Media Technology, Technische Universität München, Munich, March 2011.

[13] D. González-Ortega, F.J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, A Kinect-based system for cognitive rehabilitation exercises monitoring, Comput. Meth. Prog. Bio. 113 (November (2)) (2013) 620–631.

[14] D.A. Brooks, A.M. Howard, Quantifying upper-arm rehabilitation metrics for children through interaction with a humanoid robot, Appl. Bionics Biomech. 9 (April (2)) (2012) 157–172.

[15] M. Hamel, R. Fontaine, P. Boissy, In-home telerehabilitation for geriatric patients, IEEE Eng. Med. Biol. 27 (July–August (4)) (2008) 29–37.

[16] T. Giorgino, P. Tormene, G. Maggioni, C. Pistarini, S. Quaglini, Wireless support to poststroke rehabilitation: MyHeart's neurological rehabilitation concept, IEEE Trans. Inform. Technol. B 13 (November (6)) (2009) 1012–1018.

[17] E. Jovanov, A. Milenkovic, C. Otto, P.C. de Groen, A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation, J. NeuroEng. and Rehabil. 2 (March (1)) (2005).

[18] Y. Tao, H. Hu, H. Zhou, Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation, Int. J. Robot. Res. 26 (June (6)) (2007) 607–624.

[19] Y. Tao, H. Hu, A novel sensing and data fusion system for 3-D arm motion tracking in telerehabilitation, IEEE Trans. Instrum. Meas. 57 (May (5)) (2008) 1029–1040.

[20] G. Uswatte, C. Giuliani, C. Winstein, A. Zeringue, L. Hobbs, S.L. Wolf, Validity of accelerometry for monitoring real-world arm activity in patients with subacute stroke: evidence from the extremity constraint-induced therapy evaluation trial, Arch. Phys. Med. Rehabil. 87 (October (10)) (2006) 1340–1345.

[21] S. Wilson, R.J. Davies, T. Stone, J. Hammerton, P. Ware, S. Mawson, N. Harris, C. Eccleston, H. Zheng, N.D. Black, G. Mountain, Developing a telemonitoring system for stroke rehabilitation, Contemp. Ergon. (2007) 505–512.

[22] R.D. Willmann, G. Lanfermann, P. Saini, A. Timmermans, J. te Vrugt, S. Winter, Home stroke rehabilitation for the upper

limbs, in: Proc. 29th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, Lyon, France, 22–26 August, 2007, pp. 4015–4018.

[23] H. Zheng, R.J. Davies, N.D. Black, Web-based monitoring system for home-based rehabilitation with stroke patients, in: Proc. 18th IEEE Symposium on Computer-Based Medical Systems, Dublin, Ireland, 23–24 June, 2005, pp. 419–424.

[24] K. Kifayat, P. Fergus, S. Cooper, M. Merabti, Body area networks for movement analysis in physiotherapy treatments, in: Proc. 24th IEEE Int. Conf. Advanced Information Networking and Applications Workshops, Perth, Australia, 20–23 April, 2010, pp. 866–872.

[25] H. Zhou, H. Hu, Inertial motion tracking of human arm movements in stroke rehabilitation, in: Proc. 2005 IEEE Conf. Mechatronics Autom., vol. 3, Ontario, Canada, 29 July–1 August, 2005, pp. 1306–1311.

[26] N.W. Tierney, J. Crouch, H. Garcia, M. Walker, B.V. Lunen, G. DeLeo, G. Maihafer, S. Ringleb, Virtual reality in gait rehabilitation, in: Proc. MODSIM World, Richmond, VA, U.S.A., 10–13 September, 2007.

[27] J.M.I. Zannatha, A.J.M. Tamayo, A.D.G. Sánchez, J.E.L. Delgado, L.E.R. Cheu, W.A.S. Arévalo, Development of a system based on 3D vision, interactive virtual environments, ergonometric signals and a humanoid for stroke rehabilitation, Comput. Method Prog. Biomed. 112 (September (2)) (2013) 239–249.

[28] Ö. Yoleri, Physical Therapy Specialist, Physical Medicine and Rehabilitation Dept., İzmir Katip Çelebi Univ., Atatürk Training and Research Hospital, personal communication, June 2014.

[29] D. Moss, Psychophysiological psychotherapy: the use of biofeedback, biological monitoring, and stress management principles in psychotherapy, Psychophysiol. Today: Mag. Mind-Body Med. 2 (1) (2005) 14–18.

[30] S. Anwer, N. Quddus, M. Miraj, A. Equebal, Effectiveness of electromyographic biofeedback training on quadriceps muscle strength in osteoarthritis of knee, Hong Kong Physiother. J. 29 (December (2)) (2011) 86–93.

[31] S.L. Wolf, Electromyographic biofeedback applications to stroke patients, Phys. Ther. 63 (September (9)) (1983) 1448–1459.

[32] A. Yurtman, B. Barshan, Detection and evaluation of physical therapy exercises by dynamic time warping using wearable motion sensor units, in: E. Gelenbe, R. Lent (Eds.), Information Sciences and Systems 2013–Proc. 28th Int. Symposium on Computer and Information Sciences, 28–29 October 2013, Paris, France, Lect. Notes Electrical Eng., vol. 264, Springer International Publishing, Cham, Switzerland, 2013, pp. 305–314.

[33] A. Yurtman, Detection and Classification of Human Activities Using Wearable Sensors, M.Sc. Thesis, Bilkent Univ., Dept. of Electrical and Electronics Eng., Ankara, Turkey, September 2012.

[34] T. Giorgino, Computing and visualizing dynamic time warping alignments in R: the DTW package, J. Stat. Software 31 (August (7)) (2009) http://www.jstatsoft.org/v31/i07/.

[35] L. Deng, H. Leung, N. Gu, Y. Yang, Automated recognition of sequential patterns in captured motion streams, in: L. Chen, C. Tang, J. Yang, Y. Gao (Eds.), Web-Age Information Management, Lect. Notes Comput. Sci., vol. 6184, Springer, Berlin, Heidelberg, Germany, 2010, pp. 250–261.

[36] Xsens Technologies B.V., Enschede, The Netherlands, MTi and MTx User Manual and Technical Documentation, 2014, http://www.xsens.com.

[37] İ. Tuğcu, Physical Therapy Specialist, Department of Physical Therapy and Rehabilitation, Gülhane Military Medical Academy, Turkish Armed Forces Rehabilitation Centre, personal communication, February 2014.

[38] D. Sart, A. Mueen, W. Najjar, E. Keogh, V. Niennattrakul, Accelerating dynamic time warping subsequence search with GPUs and FPGAs, in: Proc. 10th IEEE Int. Conf. on Data Mining, Sydney, Australia, 13–17 December, 2010, pp. 1001–1006.

[39] R. Muscillo, M. Schmid, S. Conforto, T. D'Alessio, Early recognition of upper limb motor tasks through accelerometers: real-time implementation of a DTW-based algorithm, Comput. Biol. Med. 41 (March (3)) (2011) 164–172.

[40] P. Rego, P.M. Moreira, L.P. Reis, Serious games for rehabilitation: a survey and a classification towards a taxonomy, in: 5th Iberian Conf. on Information Systems and Technologies, Santiago de Compostela, Spain, 16–19 June, 2010, pp. 1–6.

[41] F. Anderson, M. Annett, W.F. Bischof, Lean on Wii: physical rehabilitation with virtual reality Wii peripherals, Stud. Health Technol. Inform. 154 (2010) 229–234.

[42] M. Müller, Information Retrieval for Music and Motion, vol.6, Springer, Berlin, Heidelberg, Germany, 2007.

[43] L. Rabiner, C. Schmidt, Application of dynamic time warping to connected digit recognition, IEEE Trans. Acoust. Speech ASSP-28 (August (4)) (1980) 377–388.