

# Hidden Markov Model Approach for the Assessment of Tele-Rehabilitation Exercises

Jan Kleine Deters<sup>1,2</sup> and Yves Rybarczyk<sup>2,3</sup>

<sup>1</sup>University of Twente,  
Enschede, The Netherlands  
Email: j.kleinedeters@utwente.nl

<sup>2</sup>Intelligent & Interactive Systems Lab (SI<sup>2</sup> Lab), Universidad de Las Américas,  
Quito, Ecuador

<sup>3</sup>CTS/UNINOVA, Department of Electrical Engineering, Nova University of Lisbon  
Monte de Caparica, Portugal  
Email: yr@uninova.pt

## ABSTRACT

*Two mandatory conditions in the development of tele-rehabilitation platforms are: (i) being based on affordable technologies and (ii) ensuring the patient is performing the exercises correctly. To do so, the present study proposes a cognitive algorithm based on a Hidden Markov Model (HMM) approach to assess in real-time the quality of a human movement recorded through a low-cost motion capture device. The assessment of the correctness of the exercises, which includes the detection of multiple undesirable compensatory movements, shows a very high accuracy (the average performance = 97%). In addition, the proposed model shows a potential for providing the patients with real-time feedback on their performance (up to five times a second). A certain limitation of the model occurs for the compensatory movements characterized by an absence of translational motion of the centre of mass (17% of misclassifications). In this situation, additional features are required to properly assess the quality of the therapeutic exercise.*

**Keywords:** Real-time motion assessment; Hidden Markov Models; rehabilitation exercises.

**Mathematics Subject Classification:** 68T05

**Computing Classification System:** G.3, D.4.8, I.2.1, I.6, H.1.2, J.3

## 1. INTRODUCTION

A current trend in medicine is home therapy systems. This concept consists of enabling patients to carry out part of the rehabilitation at home and to communicate through the Internet the evolution of the recovery process. Thus, health professionals can proceed with a remote monitoring of the patient's performance and an adaptation of the therapeutic program, accordingly. This technology could bring several advantages for the individual and the society in terms of healthcare (improvement of the recovery process by the possibility of performing rehabilitation exercises more frequently), economy (reduction of the number of medical appointments and time spent at the hospital), mobility (diminution of the transportations to and from the hospital) and ethics (healthcare democratization and increased empowerment of the patient) (Rybarczyk and Vernay, 2016).

The present study is part of a project of a Web-based platform for home motor rehabilitation (Rybarczyk et al., 2017a). The tool is developed for patients after hip arthroplasty surgery. This orthopaedic procedure is an excellent case study, because it involves people who need a postoperative functional rehabilitation program to recover strength and joint mobility. However, the condition of these patients makes difficult their transportation to and from the physiotherapist's office. Here, two main issues are tackled. First, the system must make use of a low-cost motion capture device, in order to be economically suitable. Second, the platform should automatically detect the correctness of the executed movement to provide the patient with real-time feedback. The intention is to start with a pre-recorded set of data to build preliminary detection models and, gradually, increase the database with patient's data, in order to enhance the accuracy and generalization of the classification models.

The paper focuses on smart algorithms to assess the correctness of the movements by using a signal provided by a low-cost vision-based motion capture (i.e., Kinect camera). The manuscript is divided into four sections. The first part is a survey of the main approaches to recognize and assess the human movement. The second is a description of the methodology chosen to evaluate a therapeutic movement, which is part of the rehabilitation protocol. Section 3 consists of a presentation of the results. Finally, some conclusions and perspectives are drawn up regarding the cognitive algorithms that can be used for tele-rehabilitation purposes.

## **2. MOTION ASSESSMENT METHOD**

### **2.1. Related work**

Human motion analysis and classification are particularly challenging due to the intra and inter-individual variabilities in the execution of a same movement. Time duration, which is the main variance within human movement, can be coped through several time-series algorithms. A widely used is Dynamic Time Warping (DTW), which has been successfully applied in assessing rehabilitation movements (Antón, Goñi and Illarramendi, 2015; Rybarczyk et al., 2017b).

DTW finds the optimal alignment between two temporal sequences and calculates a 'local distance measure' (or local cost value) between the signals. A low Accumulated Cost Value (ACV) over the whole aligned signals indicates high similarity. Then, classification algorithms can provide the threshold of ACV for which a signal is classified as similar or different from a reference. Other algorithms that can be applied in human motion classification are Hidden Markov Models (HMM) (Papadopoulos, Axenopoulos and Daras, 2014). It is a statistical approach that models states, which are not directly visible (or hidden), from observed variables.

HMM is described in terms of probabilities. These are 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, which is described in Section 2.2.

Calin (2016) describes a comparative analysis between DTW and HMM for gesture recognition using both Kinect V1 and V2. Although obtaining a high overall accuracy with DTW, the study points out the fact that the performance of the algorithm is very sensitive to the database size. In addition, the authors claim that it is preferable to use HMM than DTW for gesture recognition, because it enables the system to be dynamically created and adjusted. Unlike HMM, DTW cannot model the stochastic nature of the signals. As it is a deterministic method, there is no knowledge about the variance within a specific movement. A hard boundary decides if a movement belongs to a category or another. Some authors attempted to implement a probability based on a DTW approach (Hernández-Vela et al., 2014), but this is not yet applied successfully in practice (Riccadonna et al., 2016). Another disadvantage is its limitation for a real-time implementation. If a signal is classified on the fly, a temporal segment needs to be matched to a part of the reference signal. This is possible, as shown in Müller (2007), but it involves an additional matching threshold, which makes it prone to errors in classifying stochastic signals, especially on small datasets.

Due to the facts that a real-time assessment is crucial and the movement variations play an important role in the evaluation of the exercises, the HMM is chosen to classify motions. HMMs can be used in real-time without the limitations mentioned for DTW (Lin and Kulic, 2014). Considering that HMMs are generative models, it is possible to find out how the categories differ from each other, based on the distribution differences.

## **2.2. Classification**

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, need 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 relate 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 (Yamato, Ohya and Ishii, 1992; Fiosina and Fiosins, 2014). 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. Also, 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 1 (where  $\lambda_i$  represents a determined model and  $O$  is a sequence of observations):

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

### 2.3. HMM state assignment

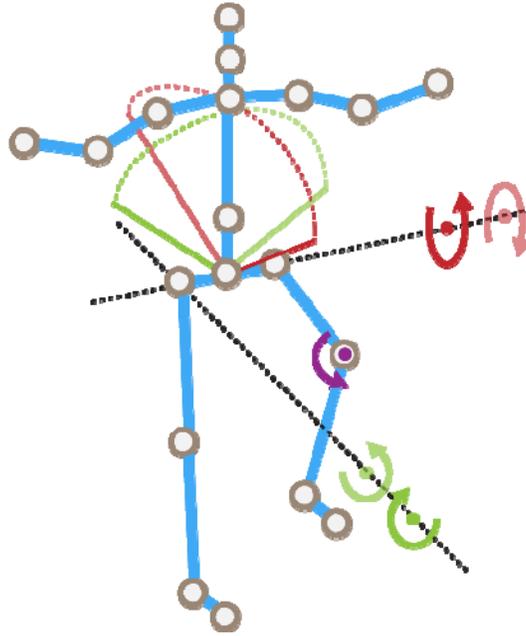
The amount of states in a HMM is a free parameter. The Bayesian Information Criteria (BIC) is a technique that aids to define a determined number of parameters by taking into account the possibility of overfitting the data when the number of states increases. BIC penalizes HMMs that have a high number of states, as described in Equation 2 (where  $n$  is the data size and  $s$  the amount of states):

$$BIC = \ln(n)s - 2\ln(MLE) \quad (2)$$

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.

### 2.4. Gesture representation

A skeletonized 3D image from a Kinect camera provides Cartesian x, y and z coordinates of twenty joints. The gesture representation is chosen to be a skeletonized image as this has been shown to improve the model accuracy (Yao et al., 2011). 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 1 shows a graphical representation of the features in relationship to the skeleton image. Table 1 describes the feature vector of the joint movements according to the anatomical terminology.



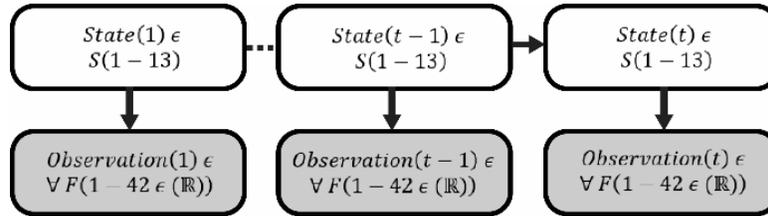
**Figure 1.** Graphical representation of the features used in the study. The movements in the egocentric frontal plane and sagittal plane are represented in green and red, respectively. The purple arrow represents the angle of the knee (independent from any plane).

*Table 1:* Feature vector describing the joint movements. For the purpose of the assessment these features are also transformed into speed and acceleration.

Right hip Frontal plane rotation (abduction)	Left hip Frontal plane rotation (abduction)	Right hip Sagittal plane rotation (flexion)	Left hip Sagittal plane rotation (flexion)
Right hip Frontal plane rotation (adduction)	Left hip Frontal plane rotation (adduction)	Right hip Sagittal plane rotation (extension)	Left hip Sagittal plane rotation (extension)
Spine centre Frontal plane rotation (lateral left)	Spine centre Frontal plane rotation (lateral right)	Spine centre Sagittal plane rotation (flexion)	Spine centre Sagittal plane rotation (extension)
Right knee (flexion)		Left knee (flexion)	

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 centre 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 2 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.  $\in$  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 4.1. Each state is dependent on its previous state and observations are samples of the associated current state.



**Figure 2.** Graphical representation of the HMM for the exercise assessment.

### 3. EXPERIMENT

#### 3.1. Protocol

Four subjects participated in the experiment. They were asked to take place at approximately two meters distance from 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 in the following order: (I) correct execution, (II) steps too short, (III) execution without moving the centre of mass, (IV) steps too large, (V) steps with bended knee, (VI) steps with bended knee and flexed torso. The last ten trials (VII) are partially wrong executions of the exercise, in which the faults II to VI are only occurring in the beginning, middle or end of the sequence. These last executions are used to evaluate the real-time applicability of the HMM technique.

#### 3.2. Materials

An application is created to capture the skeletonized image of the subjects performing an exercise. Python 2.7 is used to create a graphical user interface with the option to name, start and stop a recording. In addition, Python is used for the later processing steps, which are feature transformations and classifications. The application communicates with the Kinect SDK and whiles in recording mode writes the data into a CSV file with a frequency of 60Hz. 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 is used for the training and application of the HMMs (<https://github.com/hmmlearn/hmmlearn>).

### 3.3. Evaluation methods

Using the BIC score to select the appropriate amount of states is done for each type of trained HMMs (I-VI). It provides 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 is something missing in the execution', whereas the detection of extra states implies that 'there is something added to the movement'.

For each type of execution (I-VI) an HMM is trained, leading to a total of six distinct HMMs. In order to build a general model that can assess the movements of any subject, models that classify the executions of a subject are exclusively trained on the recordings of the other remaining subjects. This leads to a total of 24 trained models (6 per subject). The initial model parameters are set with a Gaussian density function and a full covariance matrix type (the initializations of the model parameters are done randomly, HmmLearn standard initialization is used, and EM iterates a 1000 times unless log-likelihood gain is less than 0.01). The HMM topological structure is fully connected, because priory knowledge about the expected state sequence cannot be estimated with sufficient certainty. In addition, as the outcome of six classifiers determines the most likely model that is 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 is calculated by means of the forward probabilities. Then, these probabilities are ranked from 1 to 6, where 1 and 6 are assigned to the highest and the lowest probability, respectively. A confusion matrix is 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 is provided. Finally, a range of sliding temporal windows is used to evaluate the real-time suitability of the approach. It is applied to assess the correct detection of the present types of faults in executions VII. These windows classify a subsequence in a fixed number of samples, which partially overlap over time.

### 3.4. Validation

To get a reliable result that validates the models, it is important that the test data are different from the training data. Both training and test sets must be produced by independent sampling from an infinite population, which avoid a misleading result that would not reflect what it is expected when the classifier will be deployed. This section describes the used method that enables us to apply this mandatory rule in machine learning. A 10 times repeated random sub-sampling, Monte Carlo cross-validation (MCCV) is used to evaluate the performance of each model (6 per subject and 24 in total). Results with Gaussian mixtures on real and simulated data suggest that MCCV provides genuine insight into cluster structure (Smyth, 1996). This is a method to select the most appropriate model for classification. To assure the ability of the model to generalize well, the validation is executed by applying, for each subject, the other three subject's recordings. This means that each trained HMM is used as classifier of the data of an unrelated subject. Each fold contains a trial of the three subjects. The split (80% train, 20% validation) is newly created during every validation (10 times) with a random assignment of the trials in the training and test sets. This leads 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 is used. During each validation, 6 HMMs (models I-VI) are trained such as the best performing (based on the validation score) set of HMMs is selected as models set for classification. Forward probabilities are calculated for each HMM. When the correct HMM outputs the highest probability the classification value becomes 1 and contrary 0. Per fold, each HMM classifies the remaining 6 exercises where the performance per fold is the fraction correctly classified exercises (sum of classification results) of the total classifications (36) of the 6 HMMs combined. The model's parameters differ slightly between the sets as the random data selection alters the learned state Probability Density Functions (PDFs) per fold. The best performing model set out of the 10 validations is then selected to perform the classification for the test subject.

### 3.5. Optimization methods

Optimization takes place in case the classification does not result in high classification performs. It needs to overcome the model's incapability by uncovering HMM specific states/features space that are associated to non-ambiguous characterization of the HMM. Each HMM learns a correct representation of a movement, but does not provide class distinguishing information. This information can be revealed by means of inspection of the distribution overlap. First, the predicted state occurrence in the classified sequences is characterized by the Viterbi algorithm (this algorithm predicts a state sequence given an observation sequence). This leads to a percentage value of each state occurrence per HMM (Pseudo code 1). Second, the Monte Carlo method is used to approximate the overall distribution by means of generated data draws from all HMM states. From each state, its percentage times 10000 from the PDFs are sampled.

```

Def Sample_from_models(models):
    S_numbers = 10000
    Sample_data = zeros(S_numbers,Features,States,size(models))
    For mod_num, model in enumerate(models):
        Predicted = [statesequence_O1, ..., statesequence_On]
        Percentage_Values = Histogram(Predicted,Amount_of_States)/size(Predicted)
        For n in States:
            New_Samples = Model[n].sample(Percentage_Values[n]*S_numbers)
            Sample_data[:, :, n, mod_num].append(New_Samples)
    Return Sample_data

```

**Pseudo code 1.** Finding the percentage of each states general contribution to create a sampled data distribution of a given HMM.

Then, each dimension (feature) can be inspected on sample overlap in a histogrammed fashion (30 bins with range min/max sampled values of evaluated HMMs). The overlap value per feature is calculated between two different HMM samplings expressed as a ratio (Pseudo code 2, i.e. number of samples of one HMM compared to samples of another one where the denominator is always the greatest value). The ratio only counts for those bins that contain at least 1% of the sampled data, since probabilistic outliers do not always occur in an exercise and, therefore, could not be a class separator. Finally, the features with the lowest average ratio are used to determine the sample area of

interest for class separation. This area of interest is set to be the area of the 50% most distinguishing bins. Samples falling outside of this sampling area are not considered when calculating the forward probabilities. Applying more feature value restrictions leads to less data usage in the classification.

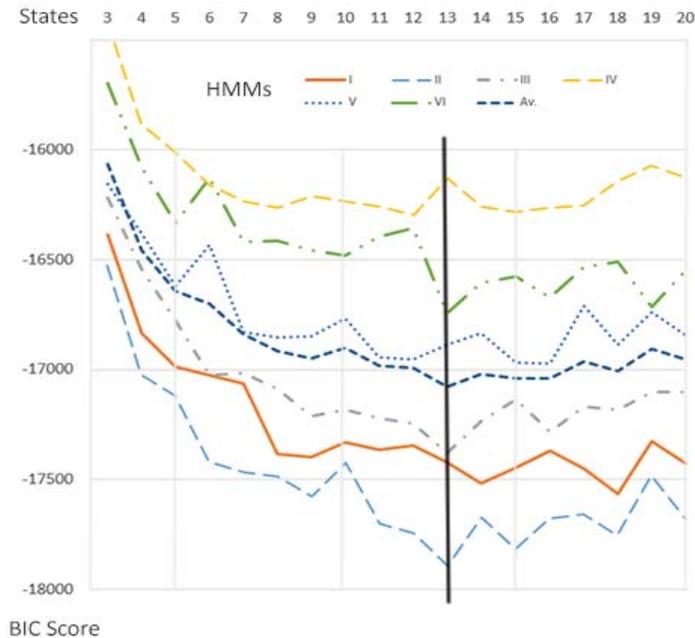
```
# Optimal Feature comparing two models
Def Optimal_Feature(models):
    sample_Hist = zeros(Features,30,models)
    For F in Features:
        sample_Hist(F, :, :) = Histogram(Sample_data(:,F, :, [models]), 30)
    # keep only the bins with more than 1%
    sample_Histogram[sample_Histogram < S_numbers/100] = 1
    ratio_Matrix = Devide(sample_Hist(:, :, models[0]), sample_Hist(:, :, models[1]))
    Invert_index = where(ratio_Matrix > 1)
    ratio_matrix[Invert_index] = 1/ratio_Matrix[Invert_index]
    ratio_matrix.sum(axis=1)
    Return Optimal_Feature = Argmin(ratio_Matrix)
```

**Pseudo code 2.** Finding the optimal feature, in which the class differentiation between 2 HMMs is the highest.

## 4. RESULTS

### 4.1. State assignment and classification

The MLE for each HMM up to twenty states is used to define the BIC scores against the amount of states (Figure 3). The profile of the BIC score against the amount of states is similar between the HMMs. Thus, it is possible to identify a consensual optimal amount of states at thirteen. This makes intuitively sense as the exercise is 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 is described by four states.



**Figure 3.** 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 2) 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 3. 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. This could be caused by the difficulty in staging this type of execution or a lack of descriptive power in the gesture representation.

In order to increase the overall classification accuracy of the movements, an additional data processing step is proposed. The main misclassifications in the previous approach seem to be caused by a lack of descriptive power of the feature vector or a large overlap within movement I and III. Therefore, an analysis is performed to find the most distinctive parts of these movements by examining the overlap and difference of the best-defined distributions of the two most discriminative feature spaces of the HMMs, of those that are trained specifically on movements I and III. In this case, 'best-defined' means a feature space where HMM I and HMM III have the least overlapping samples (see optimization method presented in Section 3.5). In the recorded movements, the feature combination of the angular acceleration of the torso in the sagittal plane (feature 1) and the angular acceleration of the right upper leg in the sagittal plane (feature 2) define the best feature dimensions in non-overlapping samples.

Table 2: 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.

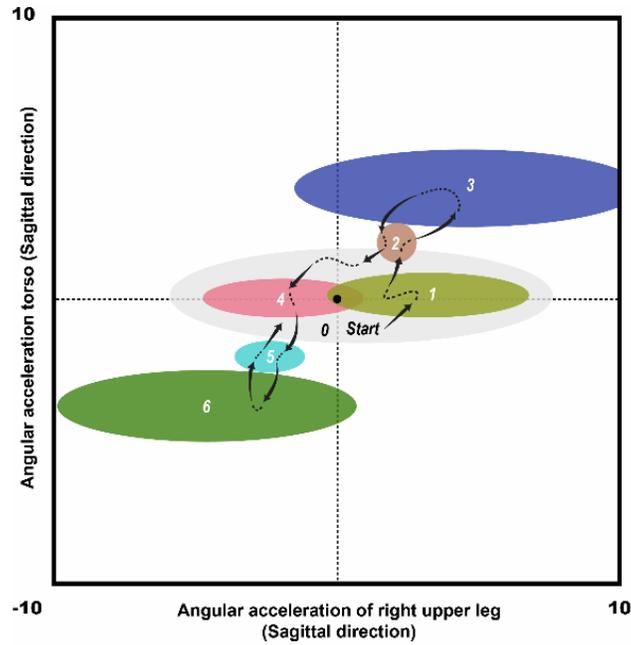
		Movement						Average Rank
		I	II	III	IV	V	VI	
Prediction	I	1	2.27	1.4	3.97	4	4	2.78
	II	2.7	1	2.8	6	6	5	3.92
	III	2.3	2.74	1.57	5	5	6	3.77
	IV	4.74	4	4.9	1.04	2	3	3.28
	V	4.27	5	4.34	1.97	1	2	3.1
	VI	6	6	6	3.04	3	1	4.18

Table 3: Performance of the classification of movements I-VI.

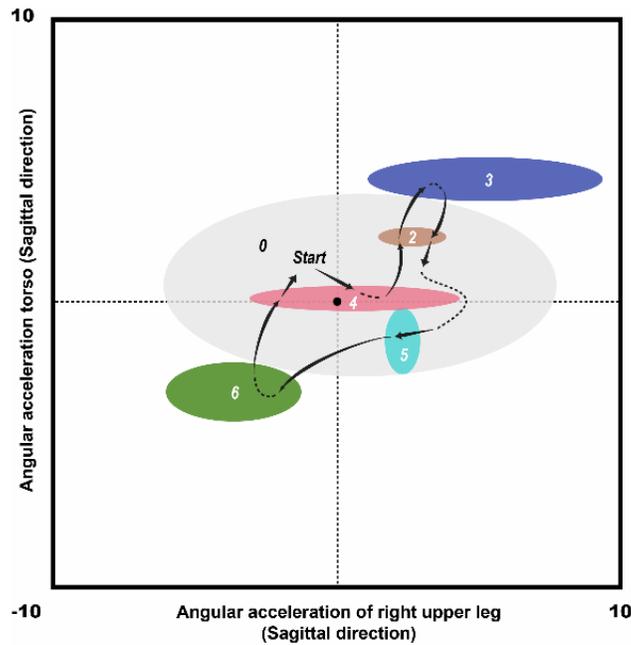
I	II	III	IV	V	VI
100%	100%	57%	97%	100%	100%

#### 4.2. Performance optimization

Figures 4 and 5 represent the 7 most prominent states (highest percentage of occurrence in classification) and the transitions between them in the feature space for HMM I and HMM III, respectively. The rest of the states are discarded in this representation as: (i) high deviation states are too general and mostly describing states that provide the function of a last resource in state assignment; and (ii) low deviation states on the other hand are too specific, which indicates a situation of overfitting.



**Figure 4.** Best defined states (HMM I) in the feature space described by features 1 and 2 in  $\text{rad}\cdot\text{s}^{-2}$ . The state distributions are visualized in terms of their first order standard deviation. The black arrows represent the most likely route of state transitions.

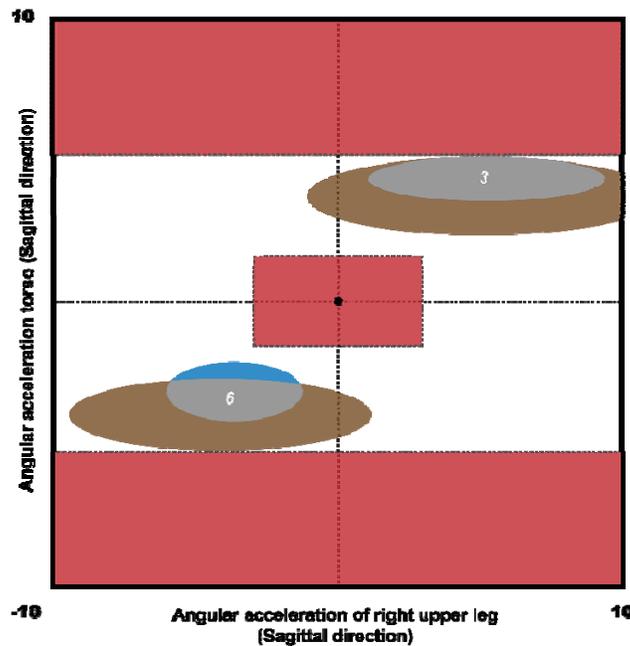


**Figure 5.** Best defined states (HMM III) in the feature space described by features 1 and 2 in  $\text{rad}\cdot\text{s}^{-2}$ . The state distributions are visualized in terms of their first order standard deviation. The black arrows represent the most likely route of state transitions.

As shown in Figures 4 and 5, a similar state transition occurs in an oscillating manner, from neutral (0) to high and low acceleration, visiting intermediate accelerations during this interval. Although the same feature space is described for the two types of movement, one state is missing in movement III (state 1). In addition, the variance of the most extreme states (3 and 6) is bigger for HMM I than HMM

III and the deceleration values seem to be higher for HMM I. Thus, the main states that can clearly differentiate movement I from movement III are states 3 and 6. The other states are highly overlapping, which means that they are not contributing to the model discriminative power.

There are several approaches to improve the classification at this point. The post variance of observation assigned as state 3 and 6 can be analysed and count as a weighted additional value. However, a value filter approach is used for values that repeat frequently and have a very low descriptive power (same predicted sample coverage for HMM I and HMM III). This approach is chosen as a trade-off between the critical amount of necessary observations in classification and a selection of the most discriminative values. An average loss of data that still allows an appropriate sample rate for a real-time classification (20Hz) is estimated as 60%, which is reached when applying the filter in 2 dimensions. This percentage is the basis for the filter boundaries. In Figure 6 the filtered region is shown as red rectangles, where the inner area is the filter area retrieved by analysing the 50% most overlapping bins per dimension. This filter excludes any observation for the classification, in which the values of the two features are  $<1.5 \text{ rad}\cdot\text{s}^{-2}$  and  $>-1.5 \text{ rad}\cdot\text{s}^{-2}$  for feature 1 (y-axis) and  $<3 \text{ rad}\cdot\text{s}^{-2}$  and  $>-3 \text{ rad}\cdot\text{s}^{-2}$  for feature 2 (x-axis). In addition, values  $>5 \text{ rad}\cdot\text{s}^{-2}$  and  $<-5 \text{ rad}\cdot\text{s}^{-2}$  for feature 1 (y-axis) and  $<10 \text{ rad}\cdot\text{s}^{-2}$  and  $>-10 \text{ rad}\cdot\text{s}^{-2}$  for feature 2 (x-axis) are eliminated, as well.



**Figure 6.** Representation of the remaining states after observation filtering. The brown distributions belong to HMM I and the grey/blue distributions to HMM III. The red squares represent areas where observations are not considered for the classification.

Thus, by applying the filter, roughly 60% of the data in each sequence is discarded before the reclassification. Most of these values are zeros, which causes no changes between the consecutive frames. These values could result from: (i) the variance in recording frequency, produced by the memory caches that may not cope with the short recording span per frame; or (ii) an actual undetectable movement (i.e., still body) between consecutive frames, which are all useless for

classifying movement. The results show that this technique improves the classification of movement III (from 57% to 83%) and does not alter the classification of the other movements (Table 4). Nevertheless, the classification accuracy of movement III is still slightly lower than the rest of the movements. The main difference between movement III and the other movements is the fact that it does not involve a translational motion of the torso. It suggests that the linear movements, and not only the angular rotations of the joints, must be considered as useful discriminative features.

*Table 4:* Performance of the classification of movements I-VI after applying the feature value filter.

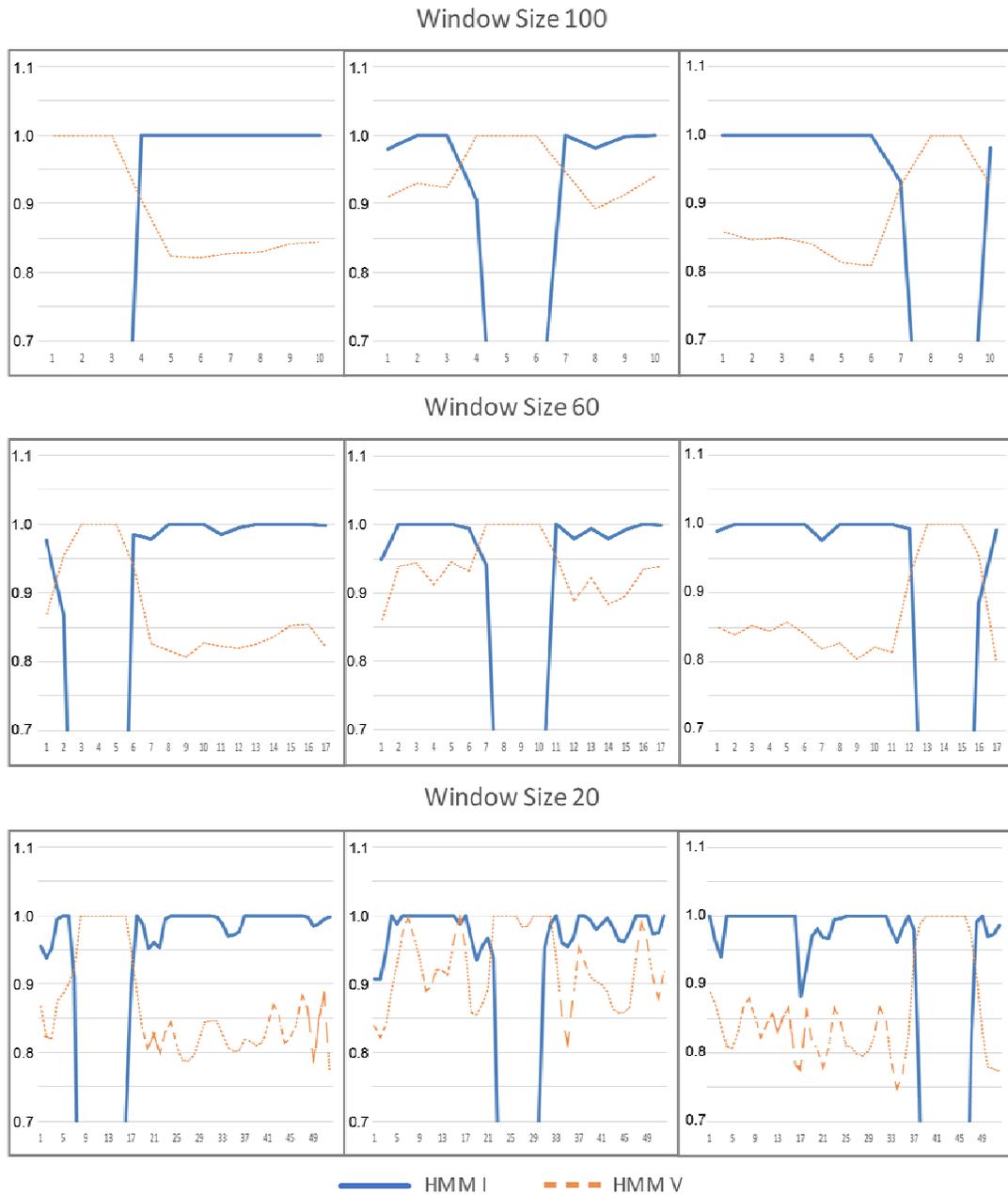
I	II	III	IV	V	VI
100%	100%	83%	97%	100%	100%

### 4.3. Real-time testing

In the previous section, entire executions are categorized according to the most typical compensatory movement. However, it does not provide insight regarding the severity of the execution error. An example of the lack of this insight is a low likelihood score caused by a long persisting small error vs. a short persisting large error. In other words, this section focuses on the duration, location and degree (e.g. bending knee a little or a lot) of the error. Since the objective of the platform is not only to provide an overall classification (see previous section) of the movement (correct vs. types of fault), but also to give a qualitative and quantitative assessment of the movement, a real-time classification is addressed. This classification aims to create awareness when the patient receives a feedback on phases of the movement, in which certain errors tend to occur. The result of this instantaneous classification will be displayed as a real-time feedback when the patient executes the exercise.

The samples of execution type VII, those that contain local errors within a correct execution, are used to evaluate the ability to apply the developed models in a real-time fashion. These models (HMM I-VI) provide the forward probabilities for an indefinite sequence, which would enable us to perform an assessment on a partial completion of the movement. Performing successive classifications during the execution of the exercise can disclose a switch in the class likelihood over time and, thus, localize the errors. The classification takes place over a selection of frames within the movement. Three different sizes of windows (length of the partial analyses) 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 7.** Classifications of execution of type VII where the beginning (left column), middle part (i.e., step to the side, middle column) and last part (right column) of the exercise is performed as type V. Three different window sizes are represented: 100, 60 and 20 samples. The orange 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. Nevertheless, 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 2). Figure 7 presents three examples of a correct sequence, except in the initial, middle or ending parts, which are performed as execution V (step with bended knee), respectively. 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.

## 5. CONCLUSIONS AND PERSPECTIVES

This study presents a HMM approach for real-time assessment of a physiotherapeutic exercise, which will be included in a project of tele-rehabilitation platform for patients after hip replacement surgery. To be able 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 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 (97%) 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 (windows size down to 200 ms). 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. A previous work (Rybarczyk et al., 2017b) has shown comparable accuracy utilizing a DTW approach. Nevertheless, this study addressed a problem of lower complexity, since it was limited to a movement classification between good and bad assessment that could be applied after the complete execution of the movement, only. In addition, the used feature representation did not account for intrapersonal differences, since the classification models were dependent on the location of the user with respect to the camera. On the contrary, the high classification accuracy and successful generalization obtained in the present study strengthens the further development of a HMM approach to assess the rehabilitation movements. The possibility to perform a real-time evaluation is a significant advantage of the HMM method, as it can provide the user with instantaneous feedback on the quality of the performed exercise. Another advantage of utilizing HMMs is that it enables the systems to be dynamically created and adjusted (Calin 2016). Since HMM is a probabilistic approach, the accuracy of the classification will increase with the individual use of the platform and the systematic update of the models.

Further work needs to include an optimization of the class separability. Robust and successful biologically inspired optimization methods such as Particle Swarm (PS), Gravitational Search Algorithms, Simulated Annealing, and Robust Ant Colony (Chen, Zhou and Luo, 2017) have shown to create stable systems (Vrkalovic, Teban and Borlea, 2017), which outperform models that are initially not optimized with such techniques (Precup, Sabau and Petriu, 2015). These methods can thus improve the performance by shaping a weighted vector of state impact in an evolutionary and robust way. This adds (i) quick insight whether multiple compensations (types of executions) can be considered simultaneously and (ii) mark possible candidate windows where errors occur (for the purpose of manual labelling of future data in the platform). In such an approach, the initial exploratory search considers the accuracy and the subsequent sequential search maximizes the classification difference between correct and incorrect HMMs. Baruah and Angelov (2014) propose a dynamic evolving clustering method, in which the weight per data point evolves (decreases), losing significance as time progresses. This notion could be integrated for optimizing the real-time application, where the sliding window can be updated in a similar fashion, creating a more suitable dynamic classification. In addition, integrating Genetic Algorithms (GA) while estimating the model parameters can increase diagnostic results of the HMMs (Zheng et al., 2017). Xue et al. (2006) show that the biologically inspired optimizations PS-HMMs outperforms both GA-HMMs and conventional HMMs. While Baum-Welch (EM algorithm) tends to get stuck in local optimum, the biologically inspired optimization methods can aid in a more robust parameter estimation.

Furthermore, the descriptive power of the movements can be extended by (i) including additional features (e.g., ankle/torso displacement and normalized speed/acceleration paths of these different joints, percentages of the maximum amplitude of the movement), and (ii) creating a preliminary detection method for the recognition of noise. This noise could be caused by computational overload. Therefore, vector quantization could be applied as it can reduce the computational costs (Mahapatra et al., 2014). In addition, exercises can be expressed into their state sequences to learn distributions of state duration as variable parameter, which provides a further insight on the ontological structure of an exercise. State sequences are modelled in terms of duration distributions and can be used as a transition model, like in the Hidden Semi-Markov Models (HSMMs) (Baratchi et al., 2014). This approach allows for a higher flexibility of the transition probability than in the HMMs, which at the end should increase the classification accuracy (Wang et al., 2014). Finally, applying new cognitive algorithms such as Linear Discriminant Analysis (LDA) and Deep Convolutional Neural Networks (DCNN) may help to find the optimal descriptor combination to distinguish between the different classes in a non-handcrafted manner (Yang et al., 2015), which diminishes the human error in selecting appropriate features.

## ACKNOWLEDGEMENT

The authors would like to thank CEDIA (Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia) for partially funding the project “Sistema de tele-rehabilitación para la auto-reeducación de los pacientes después de una cirugía de sustitución de cadera” (CEPRA XI-2017-15).

## REFERENCES

- Antón, D., Goñi, A., Illarramendi, A., 2015, Exercise recognition for Kinect-based telerehabilitation. *Methods Inf. Med.* **54**, 145-155.
- Baratchi, M., Meratnia, N., Havinga, P.J.M., Skidmore, A.K., Toxopeus, B.A.K.G., 2014, A hierarchical Hidden Semi-Markov Model for modeling mobility data. *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 13-17 Sept. 2014, Seattle, USA.
- Baruah, R.D., Angelov, P., 2014, DEC: Dynamically Evolving Clustering and its application to structure identification of evolving fuzzy models. *IEEE Transactions on Cybernetics* **44**, 1619-1631.
- Calin, A.D., 2016, Gesture recognition on Kinect time series data using Dynamic Time Warping and Hidden Markov Models. *Proceedings of the 18<sup>th</sup> International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. 24-27 Sept. 2016, Timisoara, Romania.
- Chen, Z., Zhou, S., Luo, J., 2017, A robust ant colony optimization for continuous functions. *Expert Systems with Applications* **81**, 309-320.
- Fiosina, J., Fiosins, M., 2014, Resampling based modelling of individual routing preferences in a distributed traffic network. *International Journal of Artificial Intelligence* **12**, 79-103.
- Hernández-Vela, A., Bautista, M.A., Perez-Sala, X., Ponce-López, V., Escalera, S., Baró, X., Pujol, O., Angulo, C., 2014, Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in RGB-D. *Pattern Recognition Letters* **50**, 112-121.
- Lin, J.F., Kulic, D., 2014, Online segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **22**, 168-180.
- Mahapatra, A., Mishra, T.K., Sa, P.K., Majhi, B., 2014, Human recognition system for outdoor videos using Hidden Markov Model. *International Journal of Electronics and Communications* **68**, 227-236.
- Müller, M., 2007, *Dynamic time warping*. Information Retrieval for Music and Motion, 69-84.
- Papadopoulos, G.T., Axenopoulos, A., Daras, P., 2014, Real-time skeleton-tracking-based human action recognition using Kinect data. In Gurrin, C, Hopfgartner, F., Hurst, W., Johansen, H., Lee, H., O'Connor, N. (Eds.), *MultiMedia Modeling, Lecture Notes in Computer Science* **8325**, Heidelberg: Springer.
- Precup, R.E., Sabau, M.C., Petriu, E.M., 2015, Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems. *Applied Soft Computing* **27**, 575-589.
- Riccadonna, S., Jurman, G., Visintainer, R., Filosi, M., Furlanello, C., 2016, DTW-MIC coexpression networks from time-course data. *Plos One* **11**, e0152648.
- Rybarczyk, Y., Vernay, D., 2016, Educative therapeutic tool to promote the empowerment of disabled people. *IEEE Latin America Transactions* **14**, 3410-3417.
- Rybarczyk, Y., Kleine Deters, J., Aladro, A., Gonzalez, M., Villarreal, S., Esparza, D., 2017a, ePHoRt project: a web-based platform for home motor rehabilitation. In: Rocha Á., Correia, A.M., Adeli, H.,

- Reis, L.P., Costanzo, S. (Eds.), *Recent Advances in Information Systems and Technologies, Advances in Intelligent Systems and Computing* **570**, Heidelberg: Springer.
- Rybarczyk, Y., Kleine Deters, J., Aladro, A., Esparza, D., Gonzalez, M., Villarreal, S., Nunes, I., 2017b, Recognition of physiotherapeutic exercises through DTW and low-cost vision-based motion capture. In Nunes, I. (Ed.), *Advances in Human Factors and Systems Interaction, Advances in Intelligent Systems and Computing* **592**, Heidelberg: Springer.
- Smyth, P., 1996, Clustering using Monte Carlo cross-validation. *Proceedings of the 2<sup>nd</sup> International Conference on Knowledge Discovery and Data Mining*, 2-4 Aug. 1996, Portland, Oregon, USA.
- Vrkalovic, S., Teban, T.A., Borlea, I.D., 2017, Stable Takagi-Sugeno fuzzy control designed by optimization. *International Journal of Artificial Intelligence* **15**, 17-29.
- Wang, Q., Xu, Y., Chen, Y.L., Wang, Y., Wu, X., 2014, Dynamic hand gesture early recognition based on Hidden Semi-Markov Models. *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics*. 5-10 Dec. 2014, Bali, Indonesia.
- Xue, L., Yin, J., Ji, Z., Jiang, L., 2006, A Particle Swarm optimization for Hidden Markov Model training. *Proceedings of the 8<sup>th</sup> IEEE International Conference on Signal Processing*, 16-20 Nov. 2006, Beijing, China.
- Yamato, J., Ohya, J., Ishii, K., 1992, Recognizing human action in time-sequential images using Hidden Markov Model. *Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 15-18 June 1992, Champaign, IL, USA.
- Yang, J., Nguyen, M., San, P., Li, X., Krishnaswamy, S., 2015, Deep convolutional neural networks on multichannel time series for human activity recognition. *Proceedings of the 24<sup>th</sup> International Conference on Artificial Intelligence*. 25-31 July 2015, Buenos Aires, Argentina.
- Yao, A., Gall, J., Fanelli, G., Van Gool, L., 2011, Does human action recognition benefit from pose estimation? *Proceedings of the 22<sup>nd</sup> British Machine Vision Conference*, 29 Aug.-2 Sep. 2011, Dundee, Scotland, UK.
- Zheng, H., Wang, R., Wang, Y., Zhu, W., 2017, Fault diagnosis of photovoltaic inverters using Hidden Markov Model. *Proceedings of the 36<sup>th</sup> IEEE Chinese Control Conference*, 26-28 July 2017, Dalian, China.