DE GRUYTER OPEN

JAISCR, 2014, Vol. 4, No. 4, pp. 267 – 285
10.1515/jaiscr-2015-0014

REALTIME MOTION ASSESSMENT FOR REHABILITATION EXERCISES: INTEGRATION OF KINEMATIC MODELING WITH FUZZY INFERENCE

Wenbing Zhao[1], Roanna Lun[1], Deborah D. Espy[2] and M. Ann Reinthal[2]

[1]*Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, Ohio 44115*
wenbing@ieee.org

[2]*School of Health Sciences Cleveland State University, Cleveland, Ohio 44115*

Abstract

This article describes a novel approach to realtime motion assessment for rehabilitation exercises based on the integration of comprehensive kinematic modeling with fuzzy inference. To facilitate the assessment of all important aspects of a rehabilitation exercise, a kinematic model is developed to capture the essential requirements for static poses, dynamic movements, as well as the invariance that must be observed during an exercise. The kinematic model is expressed in terms of a set of kinematic rules. During the actual execution of a rehabilitation exercise, the similarity between the measured motion data and the model is computed in terms of their distances, which are then used as inputs to a fuzzy interference system to derive the overall quality of the execution. The integrated approach provides both a detailed categorical assessment of the overall execution of the exercise and the degree of adherence to individual kinematic rules.

1 Introduction

Physical exercise is an essential part of rehabilitative healthcare. However, for a rehabilitative program to be effective, it often requires in the range of thousands of practice repetitions, all of which must be performed exactly as prescribed. Due to the large amount time needed and the high cost of clinical sessions, it is highly desirable that a patient carry out the bulk of practice at home. To facilitate practicing at home, a portable, low-cost motion tracking system that could provide realtime assessment of the exercise with live feedback would be a great help for patients [1]. Although numerous such systems have been proposed to help improve the chances of the patient carrying out the prescribed exercises correctly, many of them are incapable of providing feedback in realtime [2, 3], and those that do, only provide very limited information [4].

In this article, we present a novel approach to realtime motion assessment and feedback for rehabilitation exercises. The main novelty is the integration of comprehensive kinematic modeling with fuzzy inference. To facilitate the assessment of all important aspects of a rehabilitation exercise, a comprehensive kinematic model is developed to capture the essential requirements for static poses and dynamic movements, as well as the invariance that must be observed during an exercise. The kinematic model is expressed in terms of a set of kinematic rules. During the actual execution of a rehabilitation exercise, the similarity between the measured motion data and the set of kinematic rules is

computed in terms of their distances. The distances for the kinematic rules are then used as inputs to the fuzzy inference system to derive the overall quality of the execution. Both the output from the fuzzy inference system for each exercise, and the similarity results with respect to the kinematic rules can be presented to the patient as feedbacks in realtime. Hence, the integrated approach provides both a detailed categorical assessment of the overall execution of the exercise and the degree of adherence to individual kinematic rules.

A foundation of this research is the development of the kinematic rules for each rehabilitation exercise. The kinematic model is defined in terms of three types of kinematic rules:

- Rules for dynamic movement. Each rule is expressed in terms of the sequence of reference configurations of a particular joint or body segment (such as an arm or leg) that delineate monotonic segments of the movement.
- Rules for static poses. Some exercises only involve stationary poses. In other exercises, some body parts must remain stationary in designated positions while other parts are moving. In both cases, static rules are needed.
- Rules for movement invariance, each of which defines the requirement for a moving body segment that must be satisfied during every iteration of the exercise.

The rules for the kinematic model are stored in a file and encoded in terms of the eXtensible Markup Language (XML) for customizability and extensibility. This is very important because rehabilitation exercises often require adjustment for different patients, and even for the same patient during different stages of the recovery process. The use of XML to encode the kinematic rules also facilitates the reuse of existing rules to define other exercises that share similar characteristics.

The set of kinematic rules define the expected motion for an ideal execution of a rehabilitation exercise. Obviously, we cannot use such rules to assess an actual execution of a rehabilitation exercise both because it is virtually impossible for even the most well-trained clinician to perform the exercise exactly as defined in the rules, and because even the most high-end motion tracking system contains measurement errors, let alone for a portable, low-cost motion tracking system (such as those based on Kinect [1]). To accommodate human errors and measurement errors, the kinematic rules can be modified by adding a tolerance bound for each parameter that is part of a configuration. Basically, the tolerance-bound-based approach applies the classic logic which categories the execution of an exercise to either correct or wrong, as we have done in our previous work [5].

However, the use of a tolerance bound to determine whether or not the patient is doing an exercise correctly does not fully capture the spectrum of the subjective assessment usually done by a clinician. For example, depending on how the patient performed an exercise, the clinician might give the patient a feedback in terms of “excellent”, “very good”, “good”, “fair”, and “poor”. Furthermore, it is tricky to set the tolerance bound. A tight tolerance bound would exclude a “good” execution of an exercise, labeling it as “wrong”, while a loose tolerance bound might label an “fair” execution of an exercise as “correct”. Therefore, it is much more desirable to apply fuzzy inference in determining the quality of the execution of an exercise, which is incorporated in this research.

The integrated approach involves the following key components to assess each exercise, as illustrated in Figure 1:

- Model the exercise in terms of a set of kinematic rules. Define the membership functions for each input and the output. Define the set of fuzzy rules.
- While tracking the execution of the exercise, the similarity (in terms of distance) of the measured motion and the ideal model as defined in each kinematic rule is calculated. The distance is calculated based on the difference between the measured parameters and those defined in the kinematic rule.
- The distance measured in the above step is used as the input to the fuzzy inference system. The distance belongs to a fuzzy set with the corresponding membership function, which indicates the quality of the execution of a particular kinematic rule. The categorical information regarding the measured distance will be displayed for

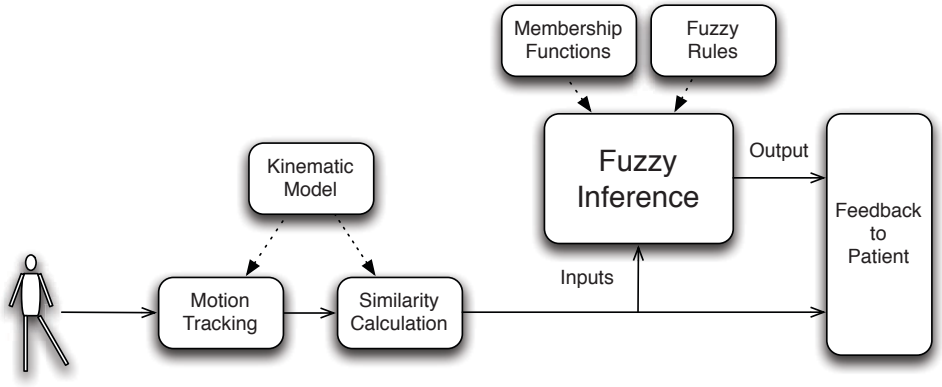

Figure 1. Architecture of the integrated realtime assessment and feedback system.

the patient as a live feedback for the corresponding kinematic rule.

– At the end of an iteration, the multiple inputs based on all kinematic rules will be aggregated and defuzzified. The defuzzified value also belongs to a fuzzy set indicating the overall quality of the execution of the current iteration. In addition to being used as a live feedback to the patient, the categorical information is used to decide if the current iteration should be counted toward the prescribed exercise load.

The remainder of this article is organized as follows. Section 2 describes the background and related work. Section 3 presents the specification for kinematic modeling of rehabilitation exercises. Section 4 elaborates the mechanisms to track the state of the execution of an exercise so that the key frames can be identified for similarity calculation. Section 5 provides the fuzzy inference framework for assessing the overall quality of the execution of an exercise. Section 6 presents two case studies, one using the hip abduction exercise and the other using the sit to stand exercise. Finally, section 7 concludes this article.

2 Background and Related Work

The foundation for human motion assessment is gesture and activity recognition. A gesture typically involves one or two hands, and possibly body poses, to convey some specific meaning, such as zoom in or zoom out. An activity usually refers to a sequence of full body movements that a person performs, such as walking, running, brushing teeth, etc., which does not necessarily convey a meaning to the computer or other persons. Rehabilitation exercises can be considered as a special form of human activity.

The approaches to human motion recognition can be roughly divided into two categories: (1) template based and (2) rule based. In the template based approach, the sequence of motions for a gesture or an activity is first recorded, which is then used as an exemplar to be compared with the observed gesture or activity either directly, or is used to train a model for the gesture, and the trained model is then used to classify the observed gesture or activity. The method used to train the model varies significantly, from simple ones such as obtaining average joint angles at a set of feature points [4], to particle filters [6], to finite state machines [7], and to sophisticated statistical methods such as hidden Markov models [8] and neural networks [9]. The main benefit of the template based approach is that either no model is needed, or the model parameters can be fitted automatically using exemplar data if a model is used. As a tradeoff, the feedback provided by these approaches often contains limited information (*e.g.,* only categorical information regarding the gesture or activity observed), which is not desirable for the purpose of rehabilitation exercise monitoring.

The rule based approach does not require the recording of exemplars and the training of sophisticated statistical models. Instead, a gesture or an activity is defined by a set of kinematic rules, created by experts, that capture the key features of the gesture or activity. This approach has a number of advantages over the template based approach:

– It is less computationally intensive because it does not require comprehensive pattern matching. Hence, it is suitable for realtime motion assessment, which is necessary for rehabilitation exercise monitoring.

– No scaling is needed because the rules reflect the invariance of the gesture or activity and it is independent from the person who performs it. This further reduces the complexity and computational cost, which makes the rule based approach more attractive for rehabilitation exercise monitoring.

– It can provide realtime feedback with much more specific information regarding exactly how the motion deviates from the predefined gesture or activity. This is particularly important for rehabilitation exercise monitoring. For example, it is far more useful to inform a patient that her leg is abducting out of the frontal plane when the leg must stay within the plane, instead of simply telling the patient that the current iteration is incorrect.

Hence, most existing research in rehabilitation exercise monitoring can be categorized into this approach. However, the rule based approach typically has the following limitations:

– The rules have to be carefully defined by experts and expressed in an implementable form. This would incur additional financial cost to a human motion recognition system and prevent a regular user from defining his/her own gestures. For rehabilitation exercises, however, this is largely not an issue because the clinician who prescribes an exercise is an expert in defining the exercise. All we need is an intuitive interface for the clinician to define an exercise using his/her familiar terminologies.

– For sophisticated gestures or activities, it may be difficult to define the rules precisely. Fortunately, rehabilitation exercises usually involve simple body movements that are easy to define.

– The rules are often hard-coded into each application, making it hard to extend or modify an existing application. In the context of rehabilitation exercise monitoring, this weakness can be problematic because the exercises prescribed for different patients may have to be customized to meet the specific needs of each patient, and the rules for the same exercise for the same patient may have to be varied during different stages of the recovery. In this research, we address this issue by encoding the kinematic rules using XML with a customizable motion recognition engine.

2.1 Template Based Approach

In the template based approach, the dominating methods for building gesture models via training data are based on machine learning, such as hidden Markov models (HMMs) and neural networks (NNs). This line of work in the context of gesture recognition has been reviewed in recent surveys [8] and [10]. Below, we highlight several studies that are closely related to rehabilitation exercise monitoring, and/or aim to provide realtime feedback.

In [11], a two-stage motion recognition method is proposed for automated rehabilitation exercise analysis with near realtime performance. The method exploits the fact that rehabilitation exercises involve periodic velocity patterns such as flexion and extension. Efficiency is achieved by identifying appropriate motion segments based on velocity peaks and zero velocity crossings in the measured joint angles in the first stage prior to applying HMM on the common motion segment in the second stage. The time it takes to segment an approximately 40-second trace is reduced to below 7 seconds compared with over 50 seconds using standard HMM or over 70 seconds using DTW. This method is not by any means fast enough for realtime feedback, but it does improve the efficiency drastically compared with traditional machine learning methods. Furthermore, the method is capable of identifying correct repetitions of an exercise, but does not provide any specific feedback regarding the incorrect iterations.

In [9], an NN based method is used to analyze the motion in rehabilitation exercises. An NN based model is used to provide more robustness against motion sensing errors due to occlusions. Furthermore, once trained, the model is capable of detecting in less than two seconds both correct movements as well as an iteration that is too fast or too slow, or a wrong static joint angle, by comparing the observed data with the predicted data based on the model. This is an important step towards realtime feedback with specific information to patients.

In [6], a method is designed specifically to enable gestural interaction with realtime non-visual feedback using a motivating example of gait analysis. The characteristics of the motion are modeled using Dynamic Movement Primitives (DMP) [12], which are capable of capturing the nonlinear dynamics of the motion. A major advantage of DMP models over other machine learning methods is that the parameters for the model possess kinematic identities important for the biomechanics of the movement of interest, such as rehabilitation exercises. Once the model is trained, a particle filter [13] is used to provide realtime feedback regarding a number of meaningful features of the movement, such as the deviation from a predicted trajectory, the probability of candidate gestures to which the current movement might belong, and the state of the gesture identified.

A gesture may also be modeled as a finite state machine (FSM) [7] that consists of a sequence of states in spatial-temporal space. FSM is different from HMM in that the number of states and state transitions are obtained dynamically from the training data instead of predefined as in HMM. Furthermore, FSM helps segment and align the training data while producing the model for a gesture. The most interesting feature for FSM is that gesture recognition is done based on the current data point, instead of operating on the entire segment of data as in HMM. This makes FSM a promising method to provide realtime feedback.

In MotionMA [4], the model for a gesture is not based on any statistical or machine learning algorithm. Instead, the model consists of a collection of joint. The training data is first filtered using a low-pass filter to remove noise and then the feature data is extracted on zero-derivatives (peaks, valleys, and inflexion points). The feature data is merged using k-means clustering. The merged data serves as the model for the gesture and is used to identify static and dynamic axes. This simple model enables the system to monitor violations in static axes continuously in realtime, and to count the repetitions for dynamic joints.

2.2 Rule Based Approaches

In the context of rehabilitation exercise monitoring, rule based approaches in general have different concerns than the template based approaches. The rules are primarily defined to assess the correctness of movements rather than to recognize gestures because it is assumed that the patient knows or is informed which particular exercise to perform. Hence, it is not necessary for the rules to completely define an exercise as long as they are in line with the therapeutic objectives of the exercise and are sufficient to automatically carry out correctness assessment and repetition count. Consequently, most studies focus on a very small set of rules and they are predominately expressed in terms of joint angles.

In [14] and [15], the rules for gait retraining are expressed in terms of the trunk flexion angle, trunk lean angle, and the distance that a set of joints for postural control traverse. In [16], the knee angle and the ankle angle are used to assess the quality of sit-to-stand and squat, and the shoulder angle is used to assess the shoulder abduction/adduction quality. In [17], the rules are expressed in terms of the knee angle in a robotic system for knee rehabilitation.

In [18], two metrics are used to evaluate the quality of the sit-to-stand exercise: (1) the minimum hip angle, in which a younger healthier person would typically have a larger value than an older person; and (2) the smoothness of the head movement, which is quantified as the area of a triangle that is determined by the second highest peak, the valley and lines that are parallel to the axes on the head-speed-versus-time plot.

Far more comprehensive rules have been developed for the purpose of recognizing hand gestures [19] and body gestures [20]. In [20], a Gesture Description Language (GDL) is introduced, in which a gesture is determined by a set of key frames. A frame contains joint positions reported by the motion sensing device (the Kinect sensor in this case). All rules are expressed in terms of one or more key frames except the final rule, which defines the gesture in terms of a sequence of basic rules. Because GDL is designed to be based on a set of key frames, it is resilient to motion sensing errors. However, as a tradeoff, it lacks the support for rules that depend on the entire trajectory of a gesture. It also lacks a guideline as to how to identify the key frames for each gesture.

In [19], a hand gesture is defined by a sequence of monotonic hand segments. A monotonic seg-

ment refers to a sequence of hand configurations in which the angles of the finger joints are either non-increasing or non-decreasing. The key frames used in GDL [20] often coincide with the reference configurations used in [19] to delineate monotonic segments when the concept is extended from hand gesture to body gesture recognition.

Our kinematic modeling resembles [19] in that dynamic movements in each rehabilitation exercise are defined in terms of monotonic segments. However, we also include kinematic rules regarding invariance requirements, which may not be important for general purpose gesture recognition, but are critical for the effectiveness of rehabilitation exercises. For example, for hip abduction, it is important that the abducting leg remain within the frontal plane the entire time, which deserves a separate invariance rule. We also accommodate rules that define static poses.

Fuzzy rules have been used in other system as a way to produce a single output score. For example, fuzzy rules are employed to capture the clinician's subjective requirements on performing rehabilitation exercises by Su [2]. Unlike in our approach, dynamic time warping is used in [2] to evaluate the similarity on the trajectory and the speed of a single joint (left or right hand) movement. It is unclear how to apply this approach to handle the more complicated exercises that we considered in this article, such as hip abduction and sit to stand. Similarly, fuzzy rules are used in [3] to generate a single score regardless the quality of golf swings. HMM is used to recognize individual features of the motion, which is different from our approach.

Finally, this research is extended from our previous work on kinematic modeling [1, 5]. In our previous work, a tolerance bound is specified for each parameter in the kinematic rules, and if the observed value for any parameter deviates from the expected value by more than the tolerance amount, the current iteration would be classified as "incorrect". In this research, we enhance our previous work by providing a more detailed feedback on the overall quality of the execution of each iteration using fuzzy inference.

3 Kinematic Modeling of Rehabilitation Exercises

The kinematic characteristics of a rehabilitation exercise are modeled in terms of three types of kinematic rules for each rehabilitation exercise:

- Rules for dynamic movement. Each rule is expressed in terms of the sequence of reference configurations of a particular joint or body segment (such as an arm or leg) that delineate the monotonic segments of each iteration (the term "segment" in monotonic segment refers to a period of continuous movements. It is not to be confused with the same term in body segment, which refers to a body part.) For a joint, a reference configuration is usually expressed in terms of the joint angle, which is defined as the angle between two adjacent body segments, or the distance between two joints. To describe the movement more accurately, one can define a reference configuration in terms of the position of a moving body segment with respect to anatomical planes (*i.e.,* the frontal, sagittal, or transverse plane).

- Rules for static poses. Some exercises only involve stationary poses. In other exercises, some body parts must remain stationary while other body segments move. In these cases, static rules are needed. In general, a rule for a static pose is also expressed in terms of the desired angle for a particular joint, or the position of a body segment with respect to anatomical planes. It is also possible to describe a static pose in terms of the distance between different joints or relative positions of different joints.

- Rules for movement invariance, each of which defines a requirement for a moving body segment that must be satisfied during every iteration of the exercise. In rehabilitation exercises, the requirement is typically defined in terms of the relative angle between the moving body segment and anatomical planes.

3.1 Encoding of Kinematic Rules

The rules are encoded using XML for its readability and extensibility. In this subsection, we describe how to encode each type of rules. Listing 1

shows an outline on how the correctness rules for each exercise are encoded. The rules start with an ExerciseName element for identification, and then a list of dynamic rules, each represented by a DynamicRule element, a set of static rules grouped together as a single StaticRule element, and a set of invariance rules grouped together as a single InvarianceRule element.

Listing 1. The set of rules that may be defined for an exercise.

```
<KinematicRules>
    <ExerciseName>...<ExerciseName>
    <DynamicRule> ... </DynamicRule>
    <DynamicRule> ... </DynamicRule>
    ...
    <DynamicRule> ... </DynamicRule>
    <StaticRule> ... </StaticRule>
    <InvarianceRule> ... </InvarianceRule>
</KinematicRules>
```

Enclosed within the DynamicRule element is a list of Configuration elements, each representing a reference configuration, as shown in Listing 2.

Listing 2. Composition of a dynamic rule.

```
<DynamicRule>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</DynamicRule>
```

The StaticRule element consists of a list of Configuration elements, as shown in Listing 3. Each Configuration element encodes the desirable position for a joint or a body segment, and it has the same format as the Configuration element used in the DynamicRule element.

Listing 3. Composition of a static rule.

```
<StaticRule>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</StaticRule>
```

Similar to the StaticRule element, an InvarianceRule element also consists of a list of Configuration elements, as shown in Listing 4. Each Configuration element encodes the restriction of a moving body segment throughout the entire iteration. Again, it has an identical format to the Configuration element used in the DynamicRule element, but differs in semantics.

Listing 4. Composition of an invariance rule.

```
<InvarianceRule>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</InvarianceRule>
```

There are three different types of Configuration elements, as shown in Listings 5, 6, and 7, respectively. The first type of Configuration element is for a joint angle, which starts with a Type element for readability and parsing. The joint angle is defined by three joints: the current joint represented by the CenterJoint element, and two adjacent joints represented by the DownstreamJoint and UpstreamJoint elements. The designated angle for the joint for a configuration is specified in the Angle element. The MaxAngleDeviation element specifies the largest deviation value from the designated angle for the configuration. This parameter is primarily used for the similarity calculation to be elaborated in Section 4.2. When determining this heuristic parameter, one must consider the following factors:

– Deviation by the amount specified in MaxAngleDeviation must not result in overlapping of the current configuration with another configuration defined in the dynamic rule for the exercise.

– Deviation from the designated value must not lead to an unsafe posture for the patient.

The MaxAngleDeviation element is also used to help determine the initial configuration in an exercise.

Listing 5. A configuration in terms of a joint angle.

```
<Configuration>
    <Type>"JointAngle"</Type>
    <CenterJoint>"JointName"</CenterJoint>
    <DownstreamJoint>"JointName"
        </DownstreamJoint>
    <UpstreamJoint>"JointName"
        </UpstreamJoint>
    <Angle>"AngleValue"</Angle>
    <MaxAngleDeviation> "..."
        </MaxAngleDeviation>
    <Duration>"DurationValue"</Duration>
    <MaxDurationDeviation> "..."
        </MaxDurationDeviation>
</Configuration>
```

The second type of Configuration element describes the required distance between a moving joint, represented by the MovingJoint element, and

a stationary one, represented by the StationaryJoint element, shown in Listing 6. The distance is represented in the Distance element. The MaxDistDeviation element defines the maximum deviation allowed from the ideal distance specified in the Distance element.

Listing 6. A configuration in terms of the distance between two joints.

```
<Configuration>
    <Type>"JointDistance"</Type>
    <Joint1>"JointName"</Joint1>
    <Joint2>"JointName"</Joint2>
    <Distance>"Value"</Distance>
    <MaxDistDeviation> "..."
        </MaxDistDeviation>
    <Duration>"DurationValue"</Duration>
    <MaxDurationDeviation> "..."
        </MaxDurationDeviation>
</Configuration>
```

The last type of Configuration element describes the orientation of a body segment. The body segment is encoded by two elements, DownstreamJoint and UpstreamJoint, to give the direction of the body segment. The orientation of the body segment is defined in terms of the angle between the segment and one or more of the anatomical planes, including, FrontalAngle, which denotes the angle between the body segment and the frontal plane, SagittalAngle, which denotes the angle between the body segment and the sagittal plane, and TransverseAngle, which denotes the angle between the body segment and the transverse plane. Only two of the angles are needed to uniquely determine the orientation of the body segment. If an angle is not used, a value -1 is used. When the Configuration element is used in an invariance rule, typically only one of the three angles is used to define the envelope of the movement with respect to one of the three anatomical planes. Because this configuration also specifies an angle (although not joint angle), we use the MaxAngleDeviation element to define the maximum deviation from the expected angle.

Listing 7. A configuration in terms of bone orientations.

```
<Configuration>
    <Type>"BoneOrientation"</Type>
    <DownstreamJoint>"JointName"
        </DownstreamJoint>
    <UpstreamJoint>"JointName"
        </UpstreamJoint>
    <FrontalAngle>"AngleValue"
        </FrontalAngle>
    <SagittalAngle>"AngleValue"
        </SagittalAngle>
    <TransverseAngle>"AngleValue"
        </TransverseAngle>
    <MaxAngleDeviation> "..."
        </MaxAngleDeviation>
    <Duration>"DurationValue"</Duration>
    <MaxDurationDeviation> "..."
        </MaxDurationDeviation>
</Configuration>
```

Common to all three Configuration elements are a pair of elements, Duration and MaxDurationDeviation, that define the expected duration of the monotonic segment that begins with the current configuration, and the maximum deviation from the expected value. These two elements are not used when the Configuration element is used for static and invariance rules. If the speed of the movement in an exercise is not important, these elements again are not used. A value -1 for each of the elements indicates that it is not used. Alternatively, elements that are not used can be omitted in the configuration.

4 Realtime Motion Tracking

While a patient is performing a rehabilitation exercise, the data captured by a motion sensing device, such as Microsoft Kinect, will be fed to a motion tracking engine frame-by-frame for analysis based on the kinematic rules. The primary objective of motion tracking is to determine the similarity between the actual execution of an exercise and the set of kinematic rules for the exercise for each iteration. The motion tracking is driven by the dynamic kinematic rule for each exercise because the iteration for the exercise is determined by the dynamic rule. The overall quality of the execution of an exercise is calculated at the end of each iteration.

In the presence of static kinematic rules and/or invariance kinematic rules, every newly arrived frame is compared against the configurations defined in these rules, for the entire duration of each iteration. That is, the similarity calculation for these rules is done on a frame-by-frame basis. At the end of the current iteration, the similarity is determined by the aggregated distances computed for all frames received within the iteration.

4.1 Motion Tracking for Dynamic Rules

A preliminary step in tracking the execution of an exercise is to know exactly when an iteration is started. The initial configuration in the dynamic rule defines the initial pose of each iteration. The motion tracking will not start until the initial configuration condition is met. Even though we could insist that the patient position himself/herself exactly in the fashion specified by the initial configuration, doing so might frustrate the patient because it might take numerous attempts for the patient to master this correct starting position. Hence, we apply a tolerance bound on each parameter in the initial configuration for the convenience of the patient. The tolerance bound can be derived from the maximum deviation parameter specified for each parameter, *e.g.,* the tolerance bound can be set to half of the maximum deviation. Note that a patient might decide to abort the current iteration before he/she completes the current one, either arbitrarily or due to a negative feedback received. In this case, the patient is assumed to restart from the initial configuration again.

To track the similarity of the execution of an exercise with respect to a dynamic kinematic rule, we must identify the exact frame received from the motion sensor that matches each reference configuration that is defined in the dynamic rule. This requires the tracking of the state of the execution of an exercise. To facilitate the tracking of the state, we model the exercise as a finite state machine. The number of states are determined by the number of monotonic segments in the dynamic rule, which coincides with the number of unique reference configurations defined in the dynamic rule.

Due to the repetitive nature of rehabilitation exercises, one iteration actually involves two mirrored activities. For example, one iteration of the hip abduction exercise consists of a hip abduction activity and its mirrored activity, hip adduction; and similarly, one iteration of the sit to stand exercise consists of a sit to stand activity and its mirrored activity, stand to sit.

Furthermore, the final pose of the first activity may be a transient pose or a stable pose. A patient normally would not stay in the final pose for a significant amount of time if it is a transient pose, and would continue completing the mirrored activity immediately after reaching the final pose. The final pose of the hip abduction activity is such a transient pose. On the other hand, the final pose might be stable, in that the patient could stay comfortably in that pose for a significant amount time, such as the standing pose in the sit to stand exercise.

For an exercise with a transient final pose, it is more convenient to track the execution of the exercise as a single whole activity. For an exercise with a stable final pose, we must track the execution of the exercise as two separate finite state machines, *i.e.,* we would need to define the kinematic rules for the first activity and its mirrored activity separately in two finite state machines, because the patient might move around in the stable pose as we have observed in the sit to stand exercise, which could disrupt the tracking of the state of the exercise.

Figure 2 shows the two types of finite state machine specifications. The top figure illustrates the state transitions for an exercise with a transient final pose as a single finite state machine. The bottom figure shows the state transitions for an exercise with a stable final pose using two mirrored, separate finite state machines.

We first describe the finite state machine specification for exercises with a transient final pose. An exercise is defined with a sequence of $2k-1$ reference configurations, C_1, C_2, ..., C_{k-1}, C_k, C'_{k-1}, ..., C'_2, C'_1, where C'_i is identical to C_i. (The repeated reference configuration C'_i is explicitly included in the sequence for the convenience of implementation of finite state machine of both types.) There are $2k-1$ corresponding states, S_1, S_2, ..., S_{k-1}, S_k, S'_{k-1}, ..., S'_2, S'_1, where S'_i is the mirrorred motion segment of S_i. Each state S_i is initiated by detecting the corresponding reference configuration C_i. Hence, the finite state machine is transitioned to S_i on detecting configuration C_i and it will stay in state S_i until the next reference configuration C_{i+1} is detected, as is illustrated in Figure 2. Similarly, a state S'_i is initiated by the detection of configuration C'_i, and the finite state machine will stay in the current state until the next configuration C'_{i-1} is detected.

For exercises with a stable final pose, two finite state machines are expected to run sequentially. The state transition for each finite state machine is identical to that in the first type of finite state machine.

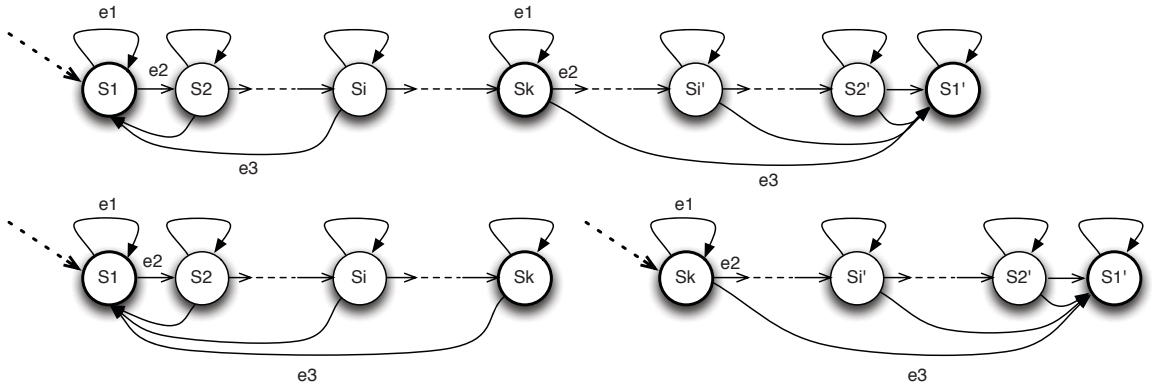

Figure 2. Finite state machines for a dynamic rule. The finite state machine on the top is used to track exercises with a transient final pose. The set of two mirrored finite state machine on the bottom are used to track exercises with a stable final pose.

Obviously we cannot expect a patient to execute an exercise exactly as defined in the kinematic rules. Hence, it is impractical to check each arriving frame against the next reference configuration because we might never see a matching frame. Instead, we have to track the actual monotonic segments dynamically in realtime while the patient is doing the exercise. The monotonic segment detection can be accomplished by tracking the change of the variables of interest (*i.e.,* joint angle, joint distance, or body segment orientation angles). When a change of sign in speed of the movement is detected (*i.e.,* from increasing to decreasing, or vice versa), the current monotonic segment has just ended. It is only at this point that the condition for the next reference configuration is checked against the last frame received for similarity calculation. (The similarity calculation method will be given in the next subsection.) Normally, the patient would continue doing the exercise to the next state.

A patient could always abort the current iteration, either arbitrarily or based on the live feedback received for each segment of the exercise. Hence, as shown in Figure 2, in the finite state machine for the exercise, there could be a transition from any state S_i to the initial state. As a result, the motion tracking engine must always check the current frame with respect to the initial configuration of the current finite state machine.

More formally, at state S_i (or$S_i^{'}$), there may be three types of events as shown in Figure 2:

– Event $e1$: The newly arrived frame does not end the current segment. In this case, the finite state machine stays in the current state S_i (or $S_i^{'}$).

– Event $e2$: The newly arrived frame ends the current segment. The finite state machine transitions to state S_{i+1} (or $S_{i-1}^{'}$) as a result of $e2$.

– Event $e3$: The newly arrived frame matches the initial configuration. Hence, the finite state machine transitions to the initial state.

Unfortunately, the addition of the mechanism to detect the actual monotonic segments at runtime makes the system vulnerable to motion sensing errors and small movement errors from the patient. The mechanism for monotonic segment detection could introduce artificially short segments in these scenarios. To overcome this problem, an additional mechanism is used. In our implementation, we use the simple mechanism described below.

For each finite state machine, we keep track of the maximum and minimum value of the variable of interest in each state. For a monotonic segment with increasing values, we delay declaring the end of the segment until the current value is smaller than the last seen maximum value by a predefined heuristic value to rule out small fluctuations of the measured variable. Similarly, for a monotonic segment with decreasing values, we delay declaring the end of the segment until the current value is larger than the last seen minimum value by a predefined heuristic value. This mechanism would inevitably introduce a small delay in state transitions and ultimately the repetition count display.

4.2 Similarity Calculation

Given a reference configuration and the matching frame received, the similarity between the two is determined by the distance between them. The smaller the distance is, the more similar between the two. The distance is calculated based on the difference between each variable defined in the reference configuration and the corresponding observed variable. Recall that there are three types of configurations: (1) joint angle configuration; (2) distance configuration; and (3) bone orientation configuration. The bone orientation configuration is typically used for invariance rules. In each of the first two types of configurations, at most two variables can be defined, where one must be either a joint angle, or a distance between two joints, and the other the duration of the movement segment. The last type of configuration typically defines one or two angles.

Given a pair of variables X_r (defined in the reference configuration) and X_o (observed variable), and the maximum deviation from the reference value as defined in the reference configuration (*e.g.,* the MaxAngleDeviation, or MaxDistDeviation, or MaxDurationDeviation) E_X, the distance between the two, D_X is defined in Equation 1:

$$D_X = \frac{|X_r - X_o|}{E_X} \tag{1}$$

If the patient moves within the maximum deviation defined in E_X, then D_X would range between 0 and 1. The distance is normalized against the maximum deviation for each variable so that the distances can be aggregated in a meaningful way.

Given a set of n variables defined in a reference configuration C, $X_1, X_2, ..., X_n$, the total distance between the reference configuration and the observed configuration is defined as the weighted sum of individual (normalized) distances for the n variables, as shown in in Equation 2:

$$D_C = \sum_{i=1}^{n} w_i D_{X_i} \tag{2}$$

where w_i is the weight for the variable X_i, and the sum of all weight must be 1 (*i.e.,* $\sum_{i=1}^{n} w_i = 1$). By default, the weight is the same for all variables, *i.e.,* $w_i = 1/n$.

So far, we have presented how to calculate the distance between a reference configuration and a single corresponding observed frame. For static and invariance kinematic rules, every frame observed will be compared against each of the reference configurations defined in each such rule. We must aggregate the individual distances at the end of the iteration for each reference configuration, and further aggregate them together for the entire rule. Similarly, for a dynamic kinematic rule consisting of several reference configurations, the distances for the individual configurations must also be aggregated as well.

Let m be the number of frames received within an iteration, and $D^i_{C_{static}}$ be the distance between the received frame i (where $i = 1, 2, ...m$) and a reference configuration C_{static} defined in the static rule. The aggregated distance for the entire iteration with respect to a reference configuration is defined in Equation 3:

$$D_{C_{static}} = \frac{\sum_{i=1}^{m} D^i_{C_{static}}}{m} \tag{3}$$

Assuming that there are k reference configurations in the static rule, the total distance for the static rule is given in Equation 4:

$$D_{static} = \frac{\sum_{j=1}^{k} D_{C^j_{static}}}{k} \tag{4}$$

The calculation for the aggregated distance for the entire iteration with respect to the invariance rule is rather similar. In the first step, the distance is aggregated for each reference configuration in the rule, as defined in Equation 5:

$$D_{C_{invariance}} = \frac{\sum_{i=1}^{m} D^i_{C_{invariance}}}{m} \tag{5}$$

Assuming that there are k reference configurations in the invariance rule, the total distance for the invariance rule is given in Equation 6:

$$D_{invariance} = \frac{\sum_{j=1}^{k} D_{C^j_{invariance}}}{k} \tag{6}$$

Assuming that a dynamic rule consists of k reference configurations, and the distance for reference configuration j (where $j = 1,2,...,k$) is $D^j_{C_{dynamic}}$, the aggregated distance for the entire iteration with respect to the dynamic rule, $D_{C_{dynamic}}$, is defined in equation 7:

$$D_{dynamic} = \frac{\sum_{j=1}^{k} D_{C^j_{dynamic}}}{k} \tag{7}$$

5 Fuzzy Inference

The distance we computed for each kinematic rule is used as the input to a fuzzy inference system that determines the overall quality of the execution of the current iteration. The overall structure of the fuzzy inference system is illustrated in Figure 3. For simplicity and clarity, we assume the kinematic model consists of only one dynamic rule, one static rule, and one invariance rule in this section.

For each input, a membership function is defined so that the degree to which it belongs in the corresponding fuzzy set can be determined. For rehabilitation exercises, we recommend to include the following classes in the fuzzy set: "excellent", "very good", "good", "fair", and "poor". The input value is limited between 0 and 1 inclusive. A distance measured might be larger than 1 if the patient deviates from a kinematic rule significantly, in which case, the distance is rounded down to 1. An example membership function based on the Guassian curve for the input is shown in Figure 4.

We define the following five rules using linguistic terms:

1 If the input from every kinematic rule is excellent, then the execution of the current iteration is excellent.

2 If the input from every kinematic rule is very good or excellent, with at least one input belongs to the "very good" category, then the execution of the current iteration is very good.

3 If the input from every kinematic rule is good, very good or excellent, with at least one input belongs to the "good" category, then the execution of the current iteration is good.

4 If the input from every kinematic rule is fair, good, very good or excellent, with at least one input belongs to the "fair" category, then the execution of the current iteration is fair.

5 If any of the inputs belongs to the "wrong" category, then the execution of the current iteration is wrong.

The five linguistic rules will be expressed in terms of 125 fuzzy rules because we have three inputs and each input has five categories. In fuzzy inference, we assume that the rules have equal weight. The output from each fuzzy rule evaluation also belongs to a fuzzy set. As implied in the fuzzy rule specification, we choose to use the same membership function as that of the input.

The surface plot for the output-input of the proposed fuzzy inference system is shown in Figure 5. The plot is generated using Matlab's Fuzzy Logic Toolbox (using the default Mamdani's fuzzy inference method) with all 125 rules and the 3 inputs. The figure shows the dependency of the output of the fuzzy inference system (*i.e.,* overall quality of the iteration) with respect to the inputs from the dynamic rule and the invariance rule (the input from the static rule is not shown). As can be seen, when the inputs are close to 0, which means that the patient's execution is very close to what is dictated by the ideal kinematic model, the quality is high (*i.e.,* close to 0).

To provide easily understood feedback to the patient, and to assist in deciding whether or not the current iteration should be counted, the output value is fuzzified again based on the membership function for the output. If the output belongs to "excellent", "very good", or "good", the current iteration is counted. Otherwise, the current iteration is not counted towards correct repetitions.

6 Case Studies

In this section, we demonstrate how to apply our approach to rehabilitation exercise assessment with two case studies, one using the hip abduction exercise, and the other using the sit to stand exercise. Both are common rehabilitation exercises. For each case study, we first define the kinematic rules for the exercise, then we show how to perform the similarity calculation with two different traces, one

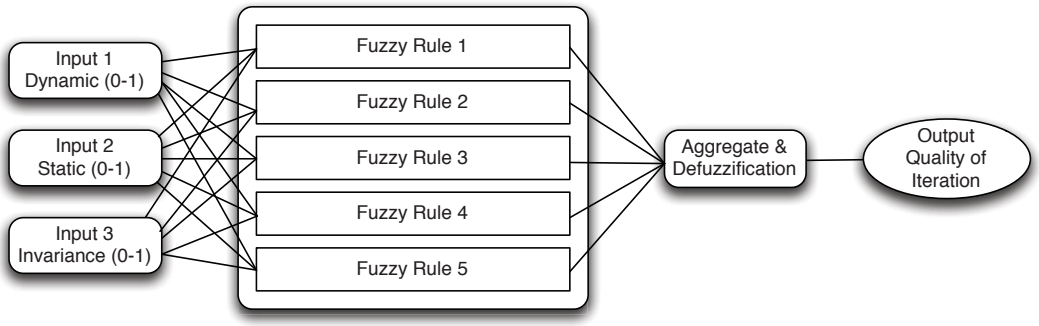

Figure 3. Overall structure of the fuzzy inference system for feedback computation.

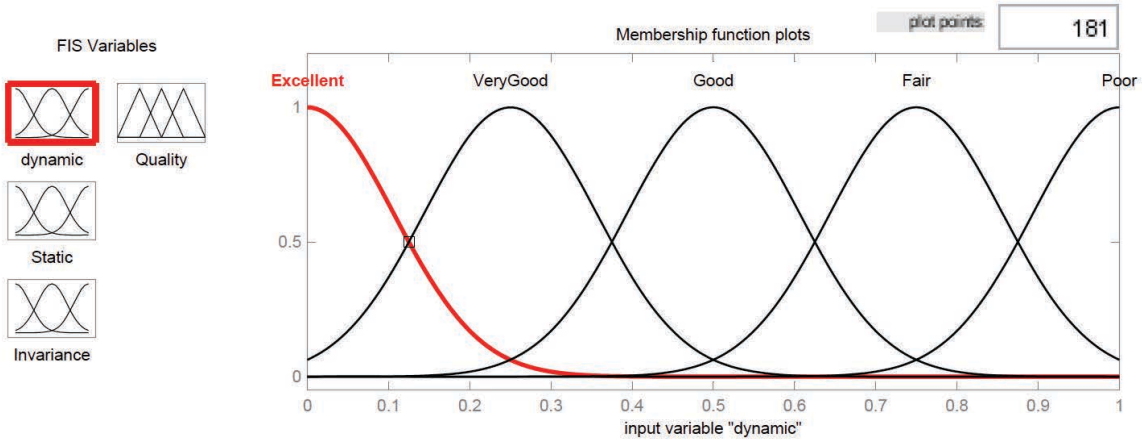

Figure 4. An example membership function for the input to our fuzzy inference system.

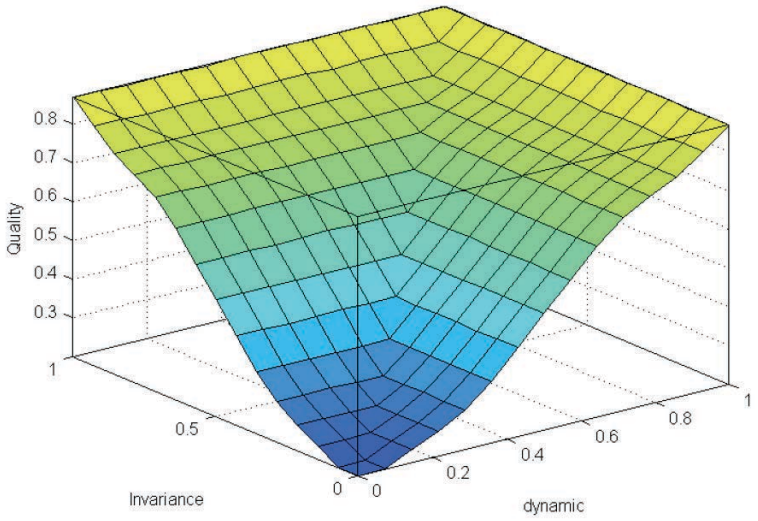

Figure 5. The surface plot for the output-inputs of the proposed fuzzy inference system. Only two of the inputs are shown.

a correct execution and the other an incorrect execution of the exercise according to the subjective judgment of the authors. In the last step, we show the result of the fuzzy inference.

6.1 Hip Abduction

For rehabilitation purposes, the hip abduction exercise involves movement of the hip in which the abducting leg moves away from the body in the same frontal plane as the rest of the body. To complete multiple iterations during an exercise, hip adduction follows the abduction movement so that the leg goes back to the midline.

6.1.1 Kinematic Rules Specification

The kinematic rules for hip abduction are shown in Listing 8. For simplicity, we omit the duration requirement for each movement segment.

Listing 8. The rules for hip abduction.

```
<KinematicRules>
  <ExerciseName>"Hip Abduction"
      <ExerciseName>
  <DynamicRule>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"HipCenter"
          </CenterJoint>
      <DownstreamJoint>"RightAnkle"
          </DownstreamJoint>
      <UpstreamJoint>"LeftAnkle"
          </UpstreamJoint>
      <Angle>"0"</Angle>
      <MaxAngleDeviation> "20"
          </MaxAngleDeviation>
    </Configuration>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"HipCenter"
          </CenterJoint>
      <DownstreamJoint>"RightAnkle"
          </DownstreamJoint>
      <UpstreamJoint>"LeftAnkle"
          </UpstreamJoint>
      <Angle>"45"</Angle>
      <MaxAngleDeviation> "20"
          </MaxAngleDeviation>
    </Configuration>
  </DynamicRule>
  <InvarianceRule>
    <Configuration>
      <Type>"BoneOrientation"</Type>
      <DownstreamJoint>"HipCenter"
          </DownstreamJoint>
      <UpstreamJoint>"RightAnkle"
          </UpstreamJoint>
      <FrontalAngle>"0"</FrontalAngle>
      <MaxAngleDeviation>"15"
          </MaxAngleDeviation>
      <SagittalAngle>"-1"</SagittalAngle>
      <TransverseAngle>"-1"
          </TransverseAngle>
    </Configuration>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"RightKnee"
          </CenterJoint>
      <DownstreamJoint>"HipCenter"
          </DownstreamJoint>
      <UpstreamJoint>"RightAnkle"
          </UpstreamJoint>
      <Angle>"170"</Angle>
      <MaxAngleDeviation>"15"
          </MaxAngleDeviation>
    </Configuration>
    </InvarianceRule>
</KinematicRules>
```

The rules for hip abduction include one dynamic rule and one invariance rule. The dynamic rule concerns the movement of the abducting leg (*i.e.,* the right leg in our example) and the movement is described in terms of two reference configurations, first when the abducting leg is at the midline of the body, and the second one when the leg is at the out-most position. A single variable, the hip angle, is defined in both configurations. In the first reference configuration, the expected angle value is 0 degree, and in the second reference configuration, the expected angle value is 45 degrees. The first configuration in the invariance rule dictates that the abducting leg must remain within the frontal plane while moving. The second configuration specifies that the abducting knee must remain straight. Due to the systematic error from the Kinect sensor [21], we set the expected knee angle to be 170 degrees instead of 180. We set the maximum angle deviation for the parameters in the dynamic rule to 20 degrees and set the maximum angle deviation for the parameters in the invariance rule to 15 degrees.

6.1.2 Similarity Calculation and Fuzzy Inference

Figure 6 shows three measured variables during a run of multiple hip abduction iterations using Microsoft Kinect, namely, hip angle, off-frontal-plane angle, and knee angle, which are needed to assess the dynamic rule and the invariance rule. In this run, the first three iterations are executed correctly, and the last two iterations are done intentionally wrong by moving the abducting leg out of the frontal plane. The hip angle is defined as the angle

between the two legs (specifically using the vector from the left hip to the left ankle, and the vector from the right hip to the right ankle). The knee angle refers to the angle formed between the upper leg and the lower leg (using the vector from the knee to the hip and the vector from the knee to the ankle). The off-frontal-plane angle refers to the angle formed between the abducting leg and the frontal plane.

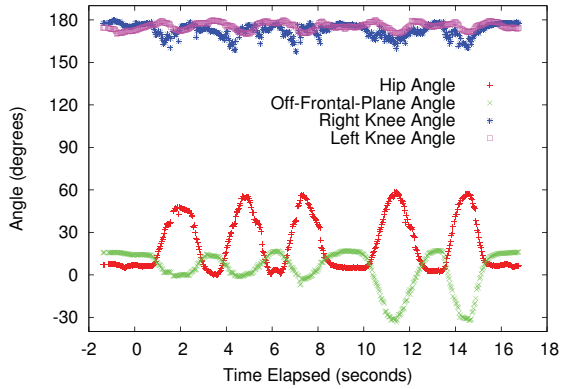

Figure 6. The measured variables a hip abduction run with five iterations using Microsoft Kinect sensor. The first three iterations are determined subjectively as "correct" while the remaining two are intentionally executed incorrectly.

The distance between the observed hip angle and that defined in the reference configuration is calculated based on Equation 1. Because it is the only parameter in the configuration, it is also the distance for the configuration itself. The total distance with respect to the dynamic rule can then be computed using Equation 7. The invariance rule is defined by two reference configurations, each with only a single parameter. The similarity calculation can be done by following Equations 1, 5 and 6.

The results of the similarity calculation for the dynamic rule and the invariance rule are used as the two inputs to the fuzzy inference system defined in Section 5. The output of the fuzzy inference system is obtained via Matlab. The inputs and the outputs for the three correct iterations and two wrong iterations of hip abduction are summarized in Table 6.1.2. The outputs are then fuzzified again so that we can provide categorical feedback to the patient. As shown in Figure 7. the first three iterations are categorized as "Good" or "Very Good" and the last two iterations are classified as "Fair".

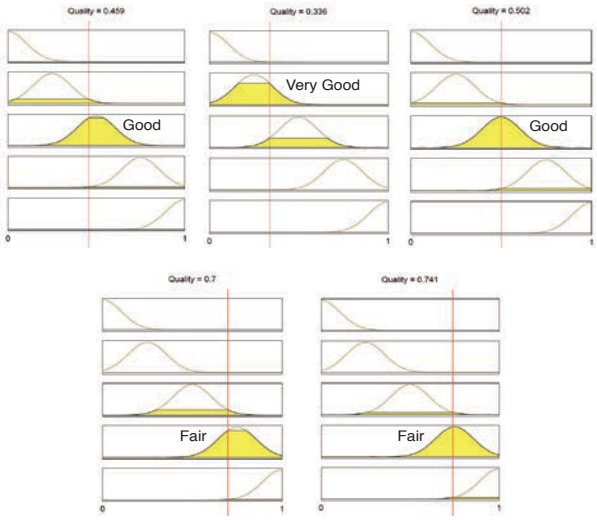

Figure 7. Fuzzified output with categorical feedback for the five iterations of hip abduction.

Note that the similarity input and consequently the output and its fuzzy classification closely depend on the maximum deviation parameters specified in the kinematic rules. A large value might categorize a wrong iteration as "Good", while a small maximum deviation value might classify a correct exercise as "Fair". For example, if we set the maximum deviation to be 20 for the parameters in the invariance rule, the last two iterations would be classified as "Good". On the other hand, if the maximum deviation is set to be 10, two of the three correct iterations would be classified as "Fair". Obviously, it is desirable to automatically set such parameters instead of manually tune them, which will be addressed in our future research using machine learning methods based on training data.

6.2 Sit to Stand

A sit to stand exercise can be used as a strengthening exercise for the large muscle groups of the legs or it can be a motor re-learning activity, or both. A patient who has multiple sclerosis, for example, may practice sit to stand to improve strength and coordinated movement of the gluteus maximus and quadriceps muscles. That would entail having both feet placed evenly on the floor at all times, and the hip angle, left/right knee angles, and left/right ankle angles all at about 90 degrees of flexion at the beginning of the exercise. The person would then lean forward with his or her trunk, moving into more hip flexion, and stand in a typical manner from that point.

Table 1. The inputs and the output for five iterations of hip abduction.

		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Input 1	C1	0.357	0.003	0.019	0.237	0.123
(dynamic)	C2	0.131	0.509	0.563	0.670	0.604
	$C_{dynamic}$	0.244	0.256	0.291	0.454	0.354
Input 2	C1	0.329	0.276	0.328	0.256	0.396
(invariance)	C2	0.578	0.395	0.676	1.159	1.202
	$C_{invariance}$	0.453	0.336	0.502	0.707	0.799
Output		0.459	0.336	0.502	0.700	0.741

6.2.1 Kinematic Rules Specification

To make multiple iterations of the exercise, the patient performs the mirrored activity (*i.e.,* stand to sit) followed by sit to stand. Unlike the hip abduction exercise, where the end pose of the abduction action is not stable (few individuals can maintain this one-leg pose for a significant amount time), the end pose of the sit to stand is stable in that a patient may choose to stay (more or less) in the standing pose for some time before he/she proceeds to sitting down. As we will show later in this section, the patient may move slightly in the standing pose, which would disrupt the motion segment tracking. Hence, it is necessary to define the sit to stand, and its mirrored activity (stand to sit) separately.

For conciseness, we define the kinematic rules only for sit to stand, as shown in Listing 9. The parameters in the square parenthesis are the actual parameters used in the similarity calculation to adjust to the systematic error of the Kinect measurement. The rules for stand to sit are identical except that the dynamic rule would consist of a set of reference configurations in the reverse order.

Listing 9. The rules for sit to stand.

```
<KinematicRules>
  <ExerciseName>"Sit to Stand"
      <ExerciseName>
  <DynamicRule>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"HipCenter"
          </CenterJoint>
      <DownstreamJoint>"ShoulderCenter"
          </DownstreamJoint>
      <UpstreamJoint>"Left Knee"
          </UpstreamJoint>
      <Angle>"90[140]"</Angle>
      <MaxAngleDeviation> "10"
          </MaxAngleDeviation>
    </Configuration>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"HipCenter"
          </CenterJoint>
      <DownstreamJoint>"ShoulderCenter"
          </DownstreamJoint>
      <UpstreamJoint>"LeftKnee"
          </UpstreamJoint>
      <Angle>"60[130]"</Angle>
      <MaxAngleDeviation> "10"
          </MaxAngleDeviation>
    </Configuration>
    <Configuration>
      <Type>"JointAngle"</Type>
      <CenterJoint>"HipCenter"
          </CenterJoint>
      <DownstreamJoint>"ShoulderCenter"
          </DownstreamJoint>
      <UpstreamJoint>"LeftKnee"
          </UpstreamJoint>
      <Angle>"180"</Angle>
      <MaxAngleDeviation> "10"
          </MaxAngleDeviation>
    </Configuration>
  </DynamicRule>
  <StaticRule>
    <Configuration>
      <Type>"BoneOrientation"</Type>
      <DownstreamJoint>"LeftAnkle"
          </DownstreamJoint>
      <UpstreamJoint>"RightAnkle"
          </UpstreamJoint>
      <FrontalAngle>"0"</FrontalAngle>
      <SagittalAngle>"90"</SagittalAngle>
      <TransverseAngle>"-1"
          </TransverseAngle>
      <MaxAngleDeviation> "20"
          </MaxAngleDeviation>
    </Configuration>
  </StaticRule>
</KinematicRules>
```

We define one dynamic rule and one static rule for sit to stand. The dynamic rule concerns the movement of the hip angle (formed by the torso and the leg), while the static rule dictates the foot placement during the entire exercise. The dynamic rule includes three reference configurations, which represent the two monotonic segments for each itera-

tion in terms of the hip angle. The first configuration specifies that the hip angle must be 90 degrees in the initial pose. During the first monotonic segment, the hip angle continuously decreases until it reaches a minimum value of 60 degrees, which is the second reference configuration and the start of the second monotonic segment. During the second segment, the hip angle continuously increases, until it reaches a maximum of 180 degrees (*i.e.,* the standing pose). The static rule specifies that the two ankles must be positioned in parallel to the frontal plane and perpendicular to the sagittal plane.

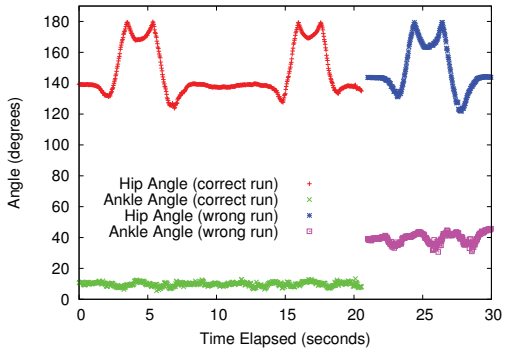

Figure 8. The measured various variables for two sit to stand runs using Microsoft Kinect sensor.

6.2.2 Similarity Calculation and Fuzzy Inference

Figure 8 shows the measured hip angle (for the dynamic rule) and the ankle angle (for the static rule) for two runs of sit to stand using an Microsoft Kinect motion sensor. One run is correct with two iterations, and the other is incorrect with a single iteration. In the wrong run, the static rule is intentionally violated by placing one foot in front the other (instead of putting the two feet in parallel). The hip angle refers to the angle formed between the torso and the leg. It is calculated using two vectors: one vector from the hip center to the head, and the other from the hip center to the center of the left and right knees (*i.e.,* (left knee position + right knee position)/2). The ankle angle refers to the angle formed between the two ankles and the frontal plane and it denotes the deviation from the expected angle specified in the static rule for sit to stand.

As can be seen, for the correct run, the measured hip angle for the sitting pose is about 140 degrees instead of 90. Similarly, the minimum angles are about 130 degrees for the second configuration instead of 60 degrees. On the other hand, the maximum hip angle is about 175 degrees for the standing pose. The measured deviation from the ideal orientation (termed "ankle angle" in the figure) fluctuates slightly around 10 degrees.

The distance between the observed hip angle and the expected value can be calculated based on Equation 1. Because it is the only parameter in each configuration in the dynamic rule, it is also the distance for the configuration itself. The total distance with respect to the dynamic rule can then be computed using Equation 7.

The static rule defines one configuration with a single parameter. The distance between the observed deviation from the expected angle can be calculated based on Equation 1 for each frame. The distance is then aggregated using Equation 3. Because there is only a single configuration for the static rule, the outcome is also the overall distance with respect to the static rule. The fuzzified outputs are shown in Figure 9. The first two iterations are categorized as "Good" and the last iteration is classified as "Fair/Poor".

Table 2. The inputs and the output for three iterations of sit to stand.

	Iteration 1	Iteration 2	Iteration 3
C1	0.076	0.041	0.358
C2	0.118	0.240	0.109
C3	0.052	0.070	0.054
$C_{dynamic}$	0.082	0.117	0.174
C_{static}	0.479	0.468	1.926
Output	0.480	0.466	0.877

7 Conclusion

In this article, we presented a novel approach to realtime motion assessment and feedback for rehabilitation exercises based on the integration of comprehensive kinematic modeling with fuzzy inference. To facilitate the assessment of all important aspects of a rehabilitation exercise, we developed a comprehensive kinematic model that captures the essential requirements for static poses, dynamic movements, as well as the invariance that must be observed during an exercise. This model was then expressed in terms of a set of kinematic rules. During the actual execution of a rehabilita-

tion exercise, we compare the similarity between the observed motion data and the set of kinematic rules by computing their distances. We then use the distances for the kinematic rule as the inputs to a fuzzy inference system to derive the overall quality of the execution of the current iteration. Both the output from the fuzzy inference system for each exercise, and the similarity results with respect to the kinematic rules can be presented to the patient as feedbacks in realtime. Hence, the integrated approach provides both a detailed categorical assessment of the overall execution of the exercise and the degree of adherence to individual kinematic rules.

Acknowledgment

An earlier version of this work was presented at the 2014 IEEE Symposium on Computational Intelligence in Health and e-Health [5]. This study is supported in part by a Faculty Research Development award, a Graduate Faculty Travel award, and several Undergraduate Summer Research awards, all from the Office of Research, Cleveland State University. Furthermore, the following undergraduate and graduate students were involved in the experiments: Hai Feng, Tracy Jennemann, Sam Yokoyama, and Vitaliy Sinyuk.

References

[1] W. Zhao, H. Feng, R. Lun, D. D. Espy, and M. Reinthal, "A kinect-based rehabilitation exercise monitoring and guidance systems," in *Proceedings of the 5th IEEE International Conference on Software Engineering and Service Science*. IEEE, 2014, pp. 762–765.

[2] C.-J. Su, "Personal rehabilitation exercise assistant with kinect and dynamic time warping," *International Journal of Information and Education Technology*, pp. 448–454, 2013.

[3] L. Zhang, J.-C. Hsieh, and J. Wang, "A kinect-based golf swing classification system using hmm and neuro-fuzzy," in *Computer Science and Information Processing (CSIP), 2012 International Conference on*, 2012, pp. 1163–1166.

[4] E. Velloso, A. Bulling, and H. Gellersen, "Motionma: motion modelling and analysis by demonstration," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 1309–1318.

[5] W. Zhao, R. Lun, D. D. Espy, and M. A. Reinthal, "Rule based realtime motion assessment for rehabilitation exercises," in *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-Health*, December 2014, pp. 133–140.

[6] Y. Visell and J. Cooperstock, "Enabling gestural interaction by means of tracking dynamical systems models and assistive feedback," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*. IEEE, 2007, pp. 3373–3378.

[7] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*. IEEE, 2000, pp. 410–415.

[8] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1473–1488, 2008.

[9] S. Nomm and K. Buhhalko, "Monitoring of the human motor functions rehabilitation by neural networks based system with kinect sensor," in *Analysis, Design, and Evaluation of Human-Machine Systems*, vol. 12, no. 1, 2013, pp. 249–253.

[10] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[11] J.-S. Lin and D. Kulic, "Online segmentation of human motion for automated rehabilitation exercise analysis," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 22, no. 1, pp. 168–180, 2014.

[12] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[13] N. Gordon, B. Ristic, and S. Arulampalam, "Beyond the kalman filter: Particle filters for tracking applications," *Artech House, London*, 2004.

[14] R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant, "Validity of the microsoft kinect for assessment of postural control," *Gait and posture*, vol. 36, no. 3, pp. 372–377, 2012.

[15] R. A. Clark, Y.-H. Pua, A. L. Bryant, and M. A. Hunt, "Validity of the microsoft kinect for providing lateral trunk lean feedback during gait retraining," *Gait & posture*, vol. 38, no. 4, pp. 1064–1066, 2013.

[16] A. Bo, M. Hayashibe, P. Poignet *et al.*, "Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect," in *EMBC'11: 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 3479–3483.

[17] E. Akdoğan, E. Taçgın, and M. A. Adli, "Knee rehabilitation using an intelligent robotic system," *Journal of Intelligent Manufacturing*, vol. 20, no. 2, pp. 195–202, 2009.

[18] Q. Wang, P. Turaga, G. Coleman, and T. Ingalls, "Somatech: an exploratory interface for altering movement habits," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014, pp. 1765–1770.

[19] B. C. Bedregal, A. C. Costa, and G. P. Dimuro, "Fuzzy rule-based hand gesture recognition," in *Artificial Intelligence in Theory and Practice*. Springer, 2006, pp. 285–294.

[20] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Systems*, vol. 20, no. 1, pp. 81–99, 2014.

[21] W. Zhao, D. D. Espy, M. Reinthal, and H. Feng, "A feasibility study of using a single kinect sensor for rehabilitation exercises monitoring: A rule based approach," in *Computational Intelligence in Healthcare and e-health (CICARE), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–8.