Rule Based Realtime Motion Assessment for Rehabilitation Exercises

Wenbing Zhao and Roanna Lun
Department of Electrical and Computer Engineering
Cleveland State University, Cleveland, Ohio 44115
Email: wenbing@ieee.org

Deborah D. Espy and M. Ann Reinthal
Department of Health Sciences
Cleveland State University
Cleveland, Ohio 44115

***Abstract*—In this paper, we describe a rule based approach to realtime motion assessment of rehabilitation exercises. We use three types of rules to define each exercise: (1) dynamic rules, with each rule specifying a sequence of monotonic segments of the moving joint or body segment, (2) static rules for stationary joints or body segments, and (3) invariance rules that dictate the requirements of moving joints or body segments. A finite state machine based approach is used in dynamic rule specification and realtime assessment. In addition to the typical advantages of the rule based approach, such as realtime motion assessment with specific feedback, our approach has the following advantages: (1) increased reusability of the defined rules as well as the rule assessment engine facilitated by a set of generic rule elements; (2) increased customizability of the rules for each exercise enabled by the use of a set of generic rule elements and the use of extensible rule encoding method; and (3) increased robustness without relying on expensive statistical algorithms to tolerate motion sensing errors and subtle patient errors.**

***Keywords*—Realtime Motion Assessment, Interactive Feedback, Therapeutic Systems and Technologies, Rehabilitation Exercises, Finite State Machine**

I. Introduction

In rehabilitative healthcare, physical exercise is a powerful intervention. However, for a rehabilitative program to be effective, it may require in the range of thousands of practice repetitions and 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 imperative that a patient carry out the bulk of practice at home. It is well known that video based guidance is helpful in improving the chances of the patient carrying out the prescribed exercises correctly. However, how to perform realtime motion assessment and provide live feedback regarding the quality of the exercise to the patient largely remains an open issue.

A foundation for human motion assessment is gesture recognition. The approaches to gesture recognition can be roughly divided into two camps: (1) template based and (2) rule based. In the template based approach, the sequence of motions for a gesture is recorded *a priori*, which is then used as an exemplar to be compared with the observed gesture directly via dynamic time warping (DTW) [1], or is used to train a model for the gesture, and the trained model is then used to classify the observed gesture. The method used to train the model varies significantly, from simple ones such as obtaining average joint angles at a set of feature points [2], to particle filters [3], to finite state machines [4], and to sophisticated statistical methods such as hidden Markov models [5] and neural networks [6]. The main benefit of the template based approach is the automatic construction of the gesture model using exemplar data (or making it unnecessary to construct a model in the case of DTW). As a tradeoff, the feedback provided by these approaches often contains limited information, which is not desirable for the purpose of rehabilitation exercise monitoring.

The rule based approach (also referred to as the algorithmic approach) does not require the recording of exemplars and the dynamic construction of models. Instead, a gesture is defined by a set of rules, created by experts, that capture the key features of the gesture. This approach has a number of advantages over the template based approach:

- It is less computationally intensive without the need for model construction and 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 and it is independent from the person who performs it. This further reduces the complexity and computational cost of gesture recognition, 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. 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. Of course, the rule based approach is not without its limitations:

- The rules for each gesture have to be carefully defined by experts and expressed in an implementable form. This would incur additional financial cost to a gesture 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, 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.
- The rule based system is prone to generate false positives (claiming that the patient's movement is wrong when in fact it is correct) in the presence of motion sensing errors. For example, a rule could dictate that a patient must move his/her leg from one position to another continuously, in which case, we may expect that the angle between the two legs continuously increases or decreases until a set value. A measurement error may indicate that the patient suddenly starts to move in the opposite direction while in fact moving the leg towards the correct position. This would lead to a false positive output.

In this paper, we take the rule based approach due to the need for realtime assessment and feedback. We aim to address the weaknesses of the traditional rule based approach by enabling reusability and customizability, and by increasing the robustness of the rule enforcement engine.

- *Reusability*: We introduce a set of basic rule elements that can be used to define correctness rules for common rehabilitation exercises. The rules are expressed in terms of eXtensible Markup Language (XML) for its extensibility and readability. Furthermore, we have designed and implemented a rule interpretation and enforcement engine. The rules are loaded into the engine at the launch time or dynamically at runtime. This enables the description and assessment of other rehabilitation exercises beyond those used in this study.
- *Customizability*: Facilitated by the XML encoding, the rules defined for an exercise can be customizable for different patients or for the same patient at different stages of rehabilitation by altering the rule parameters or by enabling/disabling a subset of rules. The customization can be done at the runtime without the need for any modification to the application code or executable.
- *Robustness*: The rule enforcement engine is robust to small measurement errors in the motion sensing device and subtle patient errors that should be tolerated. The robustness reduces false positive rates, which is important not to discourage patients. Implementation-wise, it is not trivial to ensure the robustness because the errors cannot always be masked by smoothing algorithms. In this paper, we introduce a simple mechanism that can be used to overcome these issues for increased robustness.

II. Background and Related Work

A. Template Based Approaches

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 [5] and [7]. Below, we highlight several studies that are closely related to rehabilitation exercise monitoring, and/or aim to provide realtime feedback.

In [8], 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. The efficiency is achieved by identifying candidates of 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 a 40-second or so 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 [6], an NN based method is used to analyze the motion in rehabilitation exercises. An NN based model is used for more robustness against motion sensing errors due to occlusions. Furthermore, once trained, the model is capable of detecting an iteration that is too fast or too slow, or a wrong static joint angle, by comparing the observed data with the predicted one based on the model, in less than 2 seconds (it is obviously capable of detecting correct movements). This is an important step towards realtime feedback with specific information to patients.

In [3], a method is designed specifically to enable gestural interaction with realtime non-visual feedback using a motivating example of gait analysis. The characteristic of the motion is modeled using Dynamic Movement Primitives (DMP) [9], 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 [10] is used to provide realtime feedback regarding a number of meaningful features of the movement, such as the deviation from the predicted trajectory, the probability of candidate gestures that the current movement might belong to, and the state of the hypothesis of the gesture identified.

A gesture may also be modeled as a finite state machine (FSM) [4] 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 takes care of segmenting and aligning 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 [2], the model for a gesture is not based on any statistical or machine learning algorithm. Instead, the model consists of a collection of joint angles, which are sufficient for rehabilitation exercise monitoring in many cases. The training data is first filtered using a low-pass filter to remove noise and 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.

B. 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 [11] and [12], the rules are expressed in terms of the trunk flexion angle and the distance traversed of a set of joints for postural control, and in terms of the trunk lean angle for gait retraining. In [13], 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 [14], the rules are expressed in terms of the knee angle in a robotic system for knee rehabilitation.

In [15], 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 the 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 [16] and body gestures [17]. In [17], 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 [16], a hand gesture is defined by a sequence of monotonic hand segments. A monotonic segment 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 [17] coincide often with the reference configurations used in [16] to delineate monotonic segments if the concept is extended from hand gesture to body gesture recognition.

Our rule based approach resembles [16] in that dynamic movements in each rehabilitation exercise are defined in terms of monotonic segments. However, we also include rules regarding invariance requirements, which may not be important for general purpose gesture recognition, but 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.

III. Specification of Correctness Rules

We define three types of 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 monotonic segments[1] of each iteration. 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 could define a reference configuration in terms of the the position of a moving body segment with respect to the anatomical planes (*i.e.,* the frontal, sagittal, or transverse plane).
- Rules for static poses. Some exercises only involve stationary poses. It is also possible for some body parts to remain stationary at their desirable positions while other parts are moving in some other exercises. 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 the requirement for a moving body segment that must be satisfied during every iteration of the exercise. In rehabilitation exercises, the requirement is typically

[1]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.

defined in terms of the relative angle between the moving body segment and anatomical planes.

A. Encoding of the 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.

```
<CorrectnessRules>
    <ExerciseName>...<ExerciseName>
    <DynamicRule> ... </DynamicRule>
    <DynamicRule> ... </DynamicRule>
    ...
    <DynamicRule> ... </DynamicRule>
    <StaticRules> ... </StaticRules>
    <InvarianceRules> ... </InvarianceRules>
</CorrectnessRules>
```

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

Listing 2. Composition of a dynamic rule.

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

The StaticRules 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. However, semantically it does not represent any reference configuration (that separate two monotonic segments).

Listing 3. Composition of static rules.

```
<StaticRules>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</StaticRules>
```

Similar to the StaticRules element, an InvarianceRules 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 identical format to the Configuration element used in the DynamicRule element, but differs in semantics.

Listing 4. Composition of invariance rules.

```
<InvarianceRules>
    <Configuration> ... </Configuration>
    <Configuration> ... </Configuration>
    ...
    <Configuration> ... </Configuration>
</InvarianceRules>
```

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 the configuration is specified in the Angle element. The Tolerance element specifies the amount of deviation that can be tolerated (in degrees) to accommodate motion sensing error and the tolerance of the exercise by design.

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>
    <Tolerance> "ToleranceValue"</Tolerance>
    <MaxDuration>...</MaxDuration>
    <MinDuration>...</MinDuration>
</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 and the tolerance are represented in the Distance and Tolerance elements, respectively.

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>
    <Tolerance> "ToleranceValue"</Tolerance>
    <MaxDuration>...</MaxDuration>
    <MinDuration>...</MinDuration>
</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. Three angle elements, 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. If the Configuration element is used in an invariance rule, only one of the three angles is used.

Listing 7. A configuration in terms of body orientation.

```
<Configuration>
    <Type>"BoneOrientation"</Type>
    <DownstreamJoint>"JointName"</DownstreamJoint>
    <UpstreamJoint>"JointName"</UpstreamJoint>
    <FrontalAngle>"AngleValue"</FrontalAngle>
    <SagittalAngle>"AngleValue"</SagittalAngle>
    <TransverseAngle>"AngleValue"</TransverseAngle>
    <FrontalAngleTolerance> "ToleranceValue"
        </FrontalAngleTolerance>
    <SagittalAngleTolerance> "ToleranceValue"
         </SagittalAngleTolerance>
    <TransverseAngleTolerance> "ToleranceValue"
         </TransverseAngleTolerance>
    <MaxDuration>...</MaxDuration>
    <MinDuration>...</MinDuration>
</Configuration>
```

Common to all three Configuration elements are a pair of elements, MaxDuration and MinDuration, that define the maximum and minimum duration of the monotonic segment that begins with the current configuration. 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 are not used either. A value -1 for each of the elements indicates that it is not used.

B. *Example Correctness Rules*

We define the correctness rules for two rehabilitation exercises, hip abduction and sit to stand, as examples.

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 make multiple iterations during an exercise, hip abduction follows the abduction movement so that the leg goes back to the midline. The correctness rules for hip abduction are shown in Listing 8.

Listing 8. The rules for hip abduction.

```
<CorrectnessRules>
    <ExerciseName>"Hip Abduction"<ExerciseName>
    <DynamicRule>
            <Configuration>
                <Type>"BoneOrientation"</Type>
                <DownstreamJoint>"HipCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"RightAnkle"
                     </UpstreamJoint>
                <FrontalAngle>0</FrontalAngle>
                <SagittalAngle>0</SagittalAngle>
                <TransverseAngle>−1</TransverseAngle>
                <FrontalAngleTolerance> 5
                     </FrontalAngleTolerance>
                <sagittalAngleTolerance> 5
                     </sagittalAngleTolerance>
                <TransverseAngleTolerance> −1
                     </TransverseAngleTolerance>
            </Configuration>
            <Configuration>
                <Type>"BoneOrientation"</Type>
                <DownstreamJoint>"HipCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"RightAnkle"
                     </UpstreamJoint>
                <FrontalAngle>0</FrontalAngle>
                <SagittalAngle>45</SagittalAngle>
                <TransverseAngle>−1</TransverseAngle>
                <FrontalAngleTolerance> 5
                     </FrontalAngleTolerance>
                <sagittalAngleTolerance> 5
                     </sagittalAngleTolerance>
                <TransverseAngleTolerance> −1
                     </TransverseAngleTolerance>
            </Configuration>
    </DynamicRule>
    <InvarianceRules>
            <Configuration>
                <Type>"BoneOrientation"</Type>
                <DownstreamJoint>"HipCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"RightAnkle"
                     </UpstreamJoint>
                <FrontalAngle>0</FrontalAngle>
                <SagittalAngle>−1</SagittalAngle>
                <TransverseAngle>−1</TransverseAngle>
                <FrontalAngleTolerance> 5
                     </FrontalAngleTolerance>
                <sagittalAngleTolerance> −1
                     </sagittalAngleTolerance>
                <TransverseAngleTolerance> −1
                     </TransverseAngleTolerance>
            </Configuration>
            <Configuration>
                <Type>"JointAngle"</Type>
                <CenterJoint>"RightKnee"</CenterJoint>
                <DownstreamJoint>"HipCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"RightAnkle"
                     </UpstreamJoint>
                <Angle>180</Angle>
                <Tolerance> 5</Tolerance>
        </Configuration>
    </InvarianceRules>
</CorrectnessRules>
```

The rules for hip abduction include one dynamic rule and two invariance rules. 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 (*i.e.,* at 45 degrees off the sagittal plane in our example). In both cases, a 5 degrees tolerance is allowed. The first invariance rule dictates that the abducting leg must remain within the frontal plane while moving. The second invariance rule specifies that the abducting leg must remain straight (*i.e.,* the knee angle is 180 degrees).

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 and quadriceps muscles. That would entail having both feet even 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. The correctness rules for sit to stand for a typical patient are shown in Listing 9.

Listing 9. The rules for sit to stand.

```
<CorrectnessRules>
    <ExerciseName>"Sit to Stand"<ExerciseName>
    <DynamicRule>
            <Configuration>
                <Type>"JointAngle"</Type>
                <CenterJoint>"HipCenter"</CenterJoint>
                <DownstreamJoint>"ShoulderCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"Left Knee"
                     </UpstreamJoint>
                <Angle>90</Angle>
                <Tolerance>10</Tolerance>
            </Configuration>
            <Configuration>
                <Type>"JointAngle"</Type>
                <CenterJoint>"HipCenter"</CenterJoint>
                <DownstreamJoint>"ShoulderCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"LeftKnee"
                     </UpstreamJoint>
                <Angle>60</Angle>
                <Tolerance>10</Tolerance>
            </Configuration>
            <Configuration>
                <Type>"JointAngle"</Type>
                <CenterJoint>"HipCenter"</CenterJoint>
                <DownstreamJoint>"ShoulderCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"LeftKnee"
                     </UpstreamJoint>
                <Angle>180</Angle>
                <Tolerance>5</Tolerance>
            </Configuration>
            <Configuration>
                <Type>"JointAngle"</Type>
                <CenterJoint>"HipCenter"</CenterJoint>
                <DownstreamJoint>"ShoulderCenter"
                     </DownstreamJoint>
                <UpstreamJoint>"LeftKnee"
                     </UpstreamJoint>
                <Angle>60</Angle>
                <Tolerance>10</Tolerance>
            </Configuration>
    </DynamicRule>
    <StaticRules>
            <Configuration>
                <Type>"BoneOrientation"</Type>
                <DownstreamJoint>"LeftAnkle"
                     </DownstreamJoint>
                <UpstreamJoint>"RightAnkle"
                     </UpstreamJoint>
                <FrontalAngle>0</FrontalAngle>
                <SagittalAngle>90</SagittalAngle>
                <TransverseAngle>−1</TransverseAngle>
                <FrontalAngleTolerance> 5
                     </FrontalAngleTolerance>
                <sagittalAngleTolerance> 5
                     </sagittalAngleTolerance>
                <TransverseAngleTolerance> −1
                     </TransverseAngleTolerance>
            </Configuration>
    </StaticRules>
</CorrectnessRules>
```

The rules shown here for sit to stand consist of one dynamic rule and one static rule. The dynamic rule concerns the movement of the hip angle, while the static rule dictates the foot placement during the entire exercise. The dynamic rule on hip angle includes 4 reference configurations, which represent the 4 monotonic segments for each iteration. The first configuration specifies that the hip angle must be 90 degrees with 10 degrees tolerance in the initial pose. During the first monotonic segment, the hip angle continuously decreases until it reaches a minimum value of 60 degrees (with 10 degrees tolerance), 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), which starts the third monotonic segment. During the third segment, the hip angle decreases until it reaches another minimum angle of 60 degrees, which would transition to the final monotonic segment where the hip angle would continuously increase until the sitting pose. The static rule specifies that the two ankles must be positioned in parallel to the frontal plane and perpendicular to the sagittal plane.

We should note that the rules outlined are only appropriate for some patients. For a person who has had hip replacement surgery and must avoid excessive hip flexion or risk dislocation, he/she would practice sit to stand in order to learn the proper and safe movement pattern, as well as to strengthen the involved muscles. His/her starting position would be with one leg at the same set of angles as above, but the leg involved with the replaced hip would have the foot far forward of the other one, the knee extended and the hip maintaining greater than 90 degrees of flexion. He/she would scoot to the very edge of the chair and lean a bit backwards, using primarily the uninvolved leg and his/her arms to assist would then stand up. For this type of patients, the rules would be very different.

IV. Realtime Assessment of Correctness Rules

Once the rules for an exercise are loaded in the exercise monitoring program, the movement can be assessed in realtime and specific feedback can be provided to the patient. It is relatively simple to evaluate static rules and invariance rules because they must be satisfied by every frame supplied by the motion sensing device. The calculation of the joint distance, joint angle, and body segment orientation is straightforward based on 3D vector math. We here focus on how the dynamic rules are assessed.

Each dynamic rule is assessed in realtime by a finite state machine. The number of states are determined by the number of monotonic segments in the rule, *i.e.,* the number of reference configurations. Given k reference configurations, C_1, C_2, ..., C_k, there are k number of states, S_1, S_2, ..., S_k, and each state S_i is initiated by the detecting of 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} specified in the dynamic rule is detected, as is illustrated in Figure 1. Due to the repetitive nature of rehabilitation exercise, for each iteration, the finite state machine starts and ends with the same state S_1.

On receiving a new frame containing valid motion data, the relevant metrics are calculated based on 3D vector math from the motion data. Then the calculated metrics are compared against the condition specified in C_{i+1} if the current state is S_i. When the finite state machine is initiated, it is in a special Init state and each new frame is checked against the condition in C_1, and when C_1 is met, the finite state machine transitions to state S_1 and proceeds forward. When the state transitions from S_k back to S_1, the current iteration is completed and the iteration count is incremented and appropriate feedback can be provided to the patient.

If the time range for a monotonic segment is specified in the corresponding reference configuration, the finite state machine takes a timestamp on transitioning into the current state, and checks the duration in the current state on receiving each new frame. If the duration in the current state exceeds the MaxDuration value, the patient is not doing the exercise correctly and a notification is generated to inform the patient that he/she is performing the current monotonic segment too slowly. On the other hand, when the condition for the next configuration is met and the elapsed time is less than the MinDuration value, the patient is informed that he/she is performing the current monotonic segment too quickly.

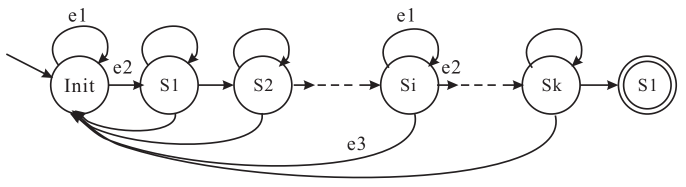

Fig. 1. Finite state machine for a dynamic rule with k monotonic segments.

V. Implementation Issues

In the previous section, we implicitly assumed that the patient would adhere to the predefined sequence of monotonic segments and could satisfy the condition for each reference configuration (except about movement speed) when practicing the prescribed exercise. This obviously might not be the case. For example, in hip abduction, if the abducting angle is smaller than what is specified in the rule, the finite state machine would be stuck forever at $S1$ if the duration for $S1$ is not specified. The patient would be frustrated by the situation (his/her iterations would not be counted, and no meaningful feedback would be provided). Such a system is neither robust nor desirable.

To be more resilient to patient mistakes in doing rehabilitation exercises, additional mechanisms must be added to the operation of the finite state machine. A key mechanism is the detecting of actual monotonic segments in realtime while the patient is doing the exercise by tracking the change of the metrics of interest (*i.e.,* joint angle, joint distance, or body segment orientation angles). When a change of sign in speed is detected (*i.e.,* from increasing to decreasing, or vice versa), the current monotonic segment has just ended. *Only at this point, the condition for the next reference configuration is checked.* If the condition is not met, then an error is detected and the patient will be informed via appropriate feedback. When this happens, the finite state machine goes back to the initial state and waits for the first frame that satisfies the condition in C_1. There are in fact two scenarios in which the error occurs:

1) The actual metric observed is smaller than the one specified in the next reference configuration.
2) The actual metric observed is larger than the one specified in the next reference configuration.

The feedback is generated for the patient accordingly.

Hence, at state S_i, there may be three types of events as shown in Figure 1:

- Event $e1$: The arrival of a frame that does not yet satisfy the condition in C_{i+1} and the elapsed time in the current state is smaller than MaxDuration, if one is specified. The finite state machine stays at the current state S_i upon this type of events.

- Event $e2$: The arrival of the first frame that satisfies the condition in C_{i+1} and the elapsed time in the current state is larger than MinDuration, if one is specified. The finite state machine transitions to state S_{i+1} as a result of $e2$.

- Event $e3$: The detection of an error, which could be any of the following:
    - Elapsed time at the current state is too short upon receiving a frame that satisfies the condition for C_{i+1}.
    - Elapsed time at the current state is too long.
    - The current monotonic segment ends too early or too late (*i.e.,* not at the specified target value).

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. As shown in Figure 3, the subject has apparent instability in the standing pose as seen from the measured hip angle. The mechanism for monotonic segment detection would introduce artificially short segments. 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 values of the metric 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 (for sit to stand, 5 degrees would be sufficient) to rule out small fluctuations of the measured angles. 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 (less than 1 second delay for sit to stand) in state transitions and ultimately the repetition count display. In addition to the rule execution, the values (including the tolerance values) used in the rules should also be carefully tuned to accommodate the systematic errors of the motion sensing device.

In the following, we present the experimental result for hip abduction and sit to stand as examples and discuss how to tune the parameters for the rules to ensure correct assessment of the movements.

A. Hip Abduction

Figure 2 shows the three measured metrics during a correct run of hip abduction using Microsoft Kinect, namely, hip angle, off-frontal-plane angle, knee angle, which are needed to assess the dynamic rule and the two invariance rules, respectively. As can be seen, the off-frontal-plane angle varies between 0 and 15 degrees, and the knee angle varies between 160-180 degrees. Hence, a tolerance of 15 degrees or larger is needed for the first invariance rule to avoid false positives, and similarly, a tolerance of 20 degrees or larger is needed.

B. Sit to Stand

Figure 3 shows the measured hip angle (for the dynamic rule) and the ankle angle (angle formed between two ankles and the frontal plane for the static rule) during a correct run of sit to stand using Microsoft Kinect. As can be seen, the measured hip angle for the sitting pose is about 140 degrees instead of 90. Similarly, the minimum angles are about 110 and 120 degrees for C_2 and C_4 instead of 60 degrees. On the other hand, the hip angle varies between 160-175 degrees for

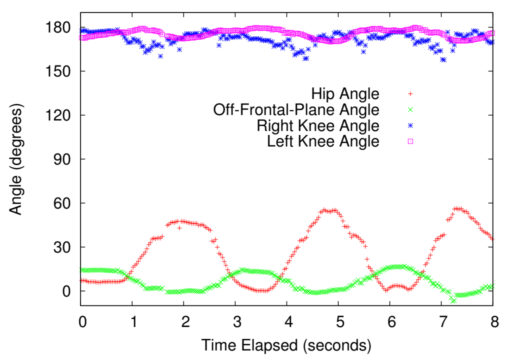

Fig. 2. The measured various metrics for a correct hip abduction run using Microsoft Kinect sensor.

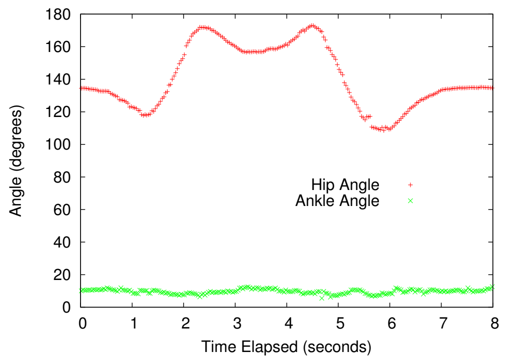

Fig. 3. The measured various metrics for a correct sit to stand run using Microsoft Kinect sensor.

the standing pose. The measured ankle angle fluctuates slightly around 10 degrees. Hence, the parameters for the rules must be set accordingly to avoid false positives.

In effect, to find the right values in the rules, the trace of a demonstration by a clinician is needed. We plan to enhance our system to automatically discover and set the right values for the correctness rules as future work. In a way, it would make our system resemble MotionMA [2], but with a much more precise model (as represented by our rules) instead.

VI. Conclusion

In this paper, we argued that the rule based approach is a good fit for rehabilitation exercise monitoring. We then introduced a specification for defining correctness rules for rehabilitation exercises using a set of carefully designed XML elements. We divided the rules into three categories: (1) dynamic rules, with each rule specifying a sequence of monotonic segments of the moving joint or body segment, (2) static rules for stationary joints or body segments, and (3) invariance rules that dictate the requirements of moving joints or body segments. This was followed by the methodology on realtime rule assessment. We also provided a comprehensive discussion on practical issues with motion assessment, in which we introduced several mechanisms to enhance the robustness of motion assessment. We believe that our rule based method for rehabilitation exercise monitoring provides the much needed features on reusability and extensibility of both the correctness rules and the rule execution engine implementation. Furthermore, the mechanisms introduced to enhance the system robustness do not depend on expensive statistical algorithms, which is critical for specific and realtime feedback to patients.

Acknowledgment

This study is supported in part by a Faculty Research Development award, and by a Graduate Faculty Travel award, both from the Office of Research, Cleveland State University.

References

[1] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *Proceedings of hte Workshop on Knowledge Discovery in Databases*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

[11] 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 & posture*, vol. 36, no. 3, pp. 372–377, 2012.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.