# On-line Segmentation of Human Motion for Automated Rehabilitation Exercise Analysis

Jonathan Feng-Shun Lin and Dana Kulić

***Abstract*— To enable automated analysis of rehabilitation movements, an approach for accurately identifying and segmenting movement repetitions is required. This paper proposes an approach for on-line, automated segmentation and identification of movement segments from continuous time-series data of human movement, obtained from body-mounted inertial measurement units or from motion capture data. The proposed approach uses a two-stage identification and recognition process, based on velocity features and stochastic modeling of each motion to be identified. In the first stage, motion segment candidates are identified based on a characteristic sequence of velocity features such as velocity peaks and zero velocity crossings. In the second stage, hidden Markov models are used to accurately identify segment locations from the identified candidates. The proposed approach is capable of on-line segmentation and identification, enabling interactive feedback in rehabilitation applications. The approach is validated on 20 healthy subjects and 4 rehabilitation patients performing rehabilitation movements, achieving segmentation accuracy of 87% with user specific templates and 79-83% accuracy with user-independent templates.**

***Index terms*— feature extraction, hidden Markov models, motion analysis, pattern recognition.**

## I. INTRODUCTION

Physical rehabilitation is a branch of modern health care that focuses on the development, maintenance and restoration of body movement and function, particularly after injury or surgery. A rehabilitation session consists of a physiotherapist's assessment of the patient's current condition, as well as the performance of physical exercises recommended by the physiotherapist. Typically, the physiotherapist will supervise the performance of the exercises to determine patient progress, as well as provide corrective feedback. Technology to measure and analyze human motion has the potential to provide the physiotherapist with more accurate tools for assessment and progress measurement, as well as to provide the patient with real-time feedback.

To enable automated measurement and analysis, the system must assess the human movement and identify exercise movement segments from measured time-series data. Human movement can be measured via motion capture systems [1], [2] or ambulatory sensors such as inertial measurement units (IMUs) [3]–[5]. Given the measured time-series data, *segmentation* is the process of locating the starting and ending time points of each movement of interest. If the patient is performing more than one type of exercise in a given recording session, *identification* (*i.e.*, labeling) of each segment with the appropriate motion type is also required. Following segmentation and identification, analysis of the identified segments can provide the physiotherapist with exercise performance metrics such as the number of repetitions performed, the duration of each repetition, the range of motion in each repetition and the variance in the range of motion and velocity over all the repetitions, many of which are hard to assess by visual inspection.

One difficulty in performing accurate segmentation and identification is the potential for a large number of degrees of freedom (DoF) in the movement data. Data streams used for single limb studies contain 4-6 DoFs, while full body human measurement can consist of 20-30 DoFs [6], making scalability an important characteristic for any segmentation algorithm. Another desirable feature is the capability for on-line processing. For rehabilitation, on-line processing is essential to allow the system to provide immediate feedback to the therapist and the patient.

Both segmentation and identification are also made more difficult due to the variability observed in human movement. Motion can vary between individuals due to differing kinematic or dynamic characteristics, and also within a single individual over time, due to short term factors such as fatigue, or long term factors such as recovery or disease progression. Moreover, these factors can introduce both spatial and temporal variability, which a good segmentation algorithm must be able to handle.

This paper proposes a template-based on-line technique based on both velocity features and probabilistic templates. A novel aggregated velocity time series feature is introduced to allow the approach to handle high dimensional data series. The system consists of a training phase, where exemplar data are used to create motion templates, and a segmentation phase, where the observed data is swept for characteristic features that match the feature templates. A semi-automated approach for feature extraction and template training is introduced, which does not require the user to specify any motion characteristics. When a segment candidate is found, the probabilistic templates are used to identify the motion and confirm the exact segment point locations[1].

Unlike many of the existing approaches to temporal segmentation, the proposed algorithm is capable of running on-

Manuscript received on Oct 3, 2012; revised on Feb 28, 2013; accepted on Apr 12, 2013. J. Lin and D. Kulić are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (email: jf2lin@uwaterloo.ca; dkulic@ece.uwaterloo.ca). Digital Object Identifier: to be assigned

[1]An early version of this work was presented in [7], [8]. This paper expands upon the initial work with a generalized approach for constructing templates capable of handling arbitrary DoF motion, where any number of DoFs may be moving simultaneously, and considers inter-individual segmentation. Additional experimental results and analysis are also provided.

line during motion execution and provides accurate segments that reject false positives, with the assistance of template information. It is suitable for high dimensionality signals from full body motion, and explicitly models both spatial and temporal variability to provide accurate segmentation results without the use of data warping or the need for a subject-specific template. The algorithm outperforms existing methods both in terms of segmentation accuracy and computational cost.

The approach is validated on 20 healthy subjects and 4 rehabilitation patients performing rehabilitation movements and a publicly available dataset of a single healthy subject performing full-body exercise motions. The algorithm attains identification accuracies of 98-100%, and segmentation accuracies of 79-90%.

## II. RELATED WORK

Time-series data segmentation and identification for human movement is an active area of research. A common approach is to rely on known motion templates to assist in the identification of motions. However, temporal and spatial variability of motion makes direct comparison of the template with the observation data difficult. Dynamic time warping (DTW) is one example of a template-based method. DTW [9], [10] identifies the temporal variations between the observed motion and the motion template by selectively warping the time scale of an observation sequence to the template. To obtain the warping coefficients, the Euclidean distance from each data point of the template to each data point of the observation is computed, creating a distance matrix. The warping path that leads to the minimum distance is taken. However, DTW becomes intractable with higher dimensionality trajectories, making it unsuitable for on-line use. Poor warping can also lead to singularity issues, where large portions of one motion are warped to small portions of the other. The severity of the singularity issue can be mitigated by constraining the warping path, either by using a fixed band [9] or computing an optimal band dynamically [11].

Keogh and Pazzani [12] note that classic DTW considers only the position data, and does not account for higher level features. They proposed the derivative DTW (DDTW), using the derivative of the position instead of position data. Using only the Euclidean distance, DTW will map two points of identical value together, even if one point is part of a falling trend and the other is part of a rising trend, which is undesirable. DDTW will capture these details that DTW ignores. Comparison between DTW and DDTW shows that the DTW tends to be overaggressive in warping, while the DDTW provides a more accurate mapping.

Ilg *et al.* [1] employ DTW in a multi-tier fashion. The observation signal is dimensionally reduced by removing all data points that are not at a velocity zero, as velocity zeros denote turning points in the motion, and thus can be considered as key features of the motion. DTW is performed on this reduced dataset to align these key features. Then, a finer alignment is performed in each of these windows. The algorithm uses DTW to calculate an optimal temporal mapping path between the template and the smaller observation segmented windows. Although both Keogh and Pazzani [12] and Ilg *et al.* [1] outperform standard DTW, they do not reduce the runtime, making the algorithm unsuitable for on-line processing, particularly for high DoF data.

If the motions to be observed are not known *a priori*, then non-template based methods are required. Pomplun and Matarić [13] propose using zero-velocity crossings (ZVCs) as segment points. ZVCs identify points where the velocity value changes sign, indicating that a joint has changed movement directions. Although a fast on-line algorithm, ZVC tends to over-segment, particularly with noisy data or as the number of DoFs increases. Since ZVC methods do not consider motion templates, it is difficult to tell which crossing points can be safely ignored. In addition, the ZVC algorithm does not provide a method for motion identification. Pomplun suggests that a distance metric can be used, but these metrics can be sensitive to spatial and temporal variations, and may not provide reliable movement identification. Fod *et al.* [14] extend the basic ZVC approach, considering both Pomplun's velocity threshold, as well as declaring a single segment point when multiple DoFs exhibit ZVCs within a short time span. This method works only with certain types of movements, since not all motions would be characterized by multiple ZVCs. For example, slow movement at the start or end of segments would prevent a ZVC declaration at the correct location, even though the movement has effectively ceased. Lieberman and Breazeal [15] also propose an extension to ZVC, demonstrating that the velocity threshold method can be complimented by other signal types, such as tactile contact.

Koenig and Matarić [16] propose a probabilistic approach for segmentation, formulating changes in signal variance as a cost function for determining segment points. A moving window is used to examine the input data. When the variance within the window is high for several DoFs, it implies that the current windowed data is in the progress of changing into another primitive, and thus is an appropriate location to segment. This allows for a quick and systematic method for segment point declaration. However, examining only variance may not provide sufficient information. A participant may perform a motion that inherently has large variance, leading to over-segmentation.

Kohlmorgen and Lemm [17] use a similar approach, using probability distributions as the segmentation cost functions. The probability density function (PDF) of the windowed observation data is calculated, for each incrementing window, over the length of the observation. The PDFs are used to train a hidden Markov model (HMM), which can then be traversed via Viterbi algorithm [18] to generate the most likely state sequences from the PDFs. These state transitions mark the segment points. Kohlmorgen and Lemm develop an on-line variant of the Viterbi algorithm, allowing the segmentation algorithm to execute on-line.

On-line template construction has also been proposed. Kulić *et al.* [19] extend Kohlmorgen's algorithm by clus-

tering together previously segmented sequences to generate new templates in real-time. Once a segment window has been identified, the segment is modeled as a HMM. The Kullback-Leibler distance between the observed HMM and existing models is calculated. If the distance is small, then the observation HMM is merged into the corresponding existing HMM. If not, it is added to the template collection, and used to improve the segmentation. The algorithm was verified on an 18-minute full-body motion sequence, and shows good segmentation accuracy, but also suffers from false positives due to the algorithm over-segmenting motion sequences into smaller subsequences that were considered to be a single segment by the manual segmentation.

An alternative approach is to assume that the observed data evolves according to an underlying deterministic model, but has been contaminated with time warping and additive noise. Probabilistic methods can be used to approximate both the parameters of the underlying model and find the segmentation locations. For example, Chiappa and Peters [20] derive the underlying uncontaminated signal by looking at the Bayesian likelihood that an sequence of observation data is from some underlying action, as well as the warping needed to transform the observation sequence to the action. This is estimated by an expectation-maximization routine. They showed that a routine that provided maximum *a posteriori* estimates for the template provided the best segmentation match. This algorithm was tested on table tennis motions, and was found to be in strong agreement with manual visual segmentation. However, this approach requires that the entire sequence is available for action fitting, making it unsuitable for on-line applications.

In the rehabilitation setting, the patient's exercises are prescribed by the therapist, and thus predefined templates can be utilized. Templates encoded as HMMs can be used as to assess the similarity between a template and an observation sequence. Bashir *et al.* [21] crops observation data based on zero-crossing occurrences in the curvature data. The Cartesian curvature was used as it is a function of the observation data and its first and second derivative, and therefore incorporates high-level features inherently. The cropped data is reduced using principal component analysis (PCA). Motions that have similar PCA coefficients are clustered together via *k*-means spectral clustering, and are used to train HMMs. Observation segmentation was performed by reducing the observation data to PCA coefficient vectors, and comparing them to the HMMs to calculate likelihood. However, utilizing PCA coefficients instead of the original data can be difficult, as similar motions may have similar PCA profiles, making them difficult to differentiate. Using curvature to segment may also lead to over-segmentation issues such as those found in the ZVC algorithms.

Lv and Nevatia [2] use HMM templates as classifiers in an AdaBoost algorithm. Features such as joint angles and joint Cartesian coordinates were used. HMM templates are generated for $M$ action types and $N$ features, resulting in $M \times N$ number of classifiers. Multi-class AdaBoost was employed to weigh the discriminative powers of each of the HMM classifiers. Segmentation was determined by dynamic programming. The observation data is split into two, with the first window starting at some minimum length $l_{min}$, and increased per iteration. Both ends of the data are inserted through the AdaBoost classifiers, and the window configuration that resulted in the highest likelihood was selected. The algorithm is run multiple times, with the starting point of the first window advancing at each run. The algorithm was verified on various upper-body motions, and was shown to be robust even when a large number of features are examined. However, this method requires a large computational cost for both training and segmenting, due to the usage of dynamic programming, and thus cannot be used on-line.

Keogh *et al.* [22] propose a 2-tier on-line segmentation algorithm. They employ a sliding window and mean square error for coarse segment estimation. Following coarse segmentation, a second window and additional distance metrics were used to determine the actual segment. Amft *et al.* [23] extend this to a 3-tier system, by adding HMMs as an identification tool for 4-DoF arm motion identification. They report an average accuracy of 72% (94%, ignoring false positives and negatives) for 4 motion primitives. The algorithm of Amft *et al.* [23], which is most similar to this paper, uses Euclidean distances to estimate segment candidate locations. Euclidean distance may not be appropriate in this situation due to temporal variations in human motion.

Previous work in the field has suggested many approaches for segmentation and identification. DTW provides an accurate method of segmentation that is robust against temporal variations, but is too computationally expensive to be employed on-line. ZVC provides a light-weight and intuitive solution, but tends to oversegment and generates large numbers of false positives. HMM and other probabilistic approaches report high segmentation accuracy, but either oversegment, require large computation time, or require the full observation sequence to be available before segmentation is possible. The proposed approach addresses these weaknesses and provides a solution that is on-line and robust against temporal and spatial variations, with the capability of rejecting false positives.

## III. PROPOSED APPROACH

In order to meet the time constraints of a real-time system, the segmentation and identification algorithm should be as computationally efficient as possible. To achieve on-line segmentation, we propose the feature-guided HMM, a fast algorithm that employs a 2-tier template matching approach to motion segmentation and identification.

The observation data is first scanned for characteristic features, consisting of velocity peaks and ZVCs, to estimate the locations of segment candidates. These segment candidates serve as framing windows for HMM-based template matching in the second stage. This two-stage approach reduces the number of times that the more computationally expensive but accurate HMM algorithm is required to run, thus reducing the overall computational costs of the algorithm. The velocity feature templates are constructed by isolating

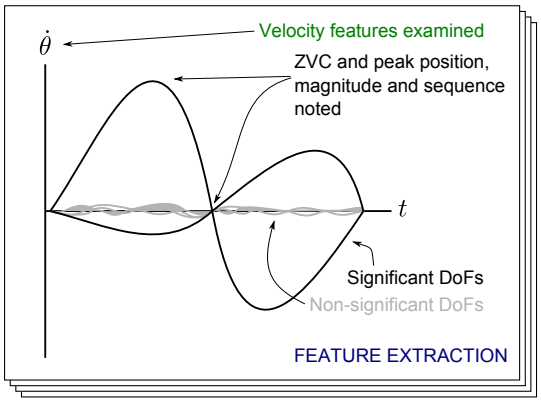


Fig. 1. Training procedure for the feature extraction. The training algorithm extracts significant DoFs and velocity features (ZVC, velocity peaks) from the provided exemplars.

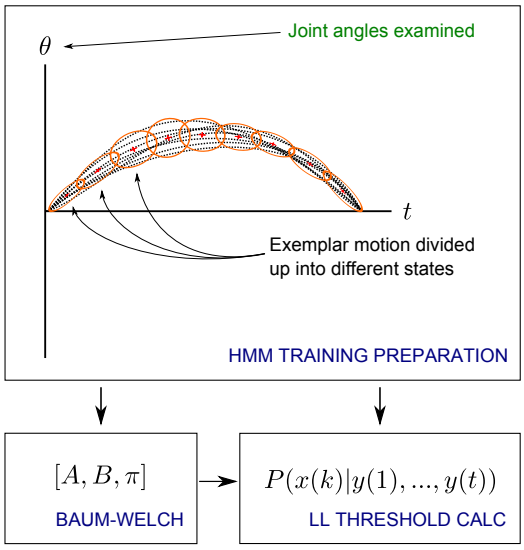


Fig. 2. Training procedure for the HMM training. The training algorithm uses means (cross) and variances (oval) of equally spaced windows as the initial values for the Baum-Welch HMM training algorithm. The exemplar motions are not actually normalized to the same length, but the window width used is adjusted for each exemplar, and thus achieves similar effects. Normalized exemplars are shown here for ease of view. The LL threshold is calculated from running the exemplars through the trained HMM.

significant DoFs, based on DoFs that exhibit large velocity changes, then scanning the available exemplars for ZVCs and velocity peaks, and proposing a feature combination that is shared by the majority of the exemplars. HMM templates are constructed by passing the exemplars into the Baum-Welch algorithm. An overview of the algorithm is shown in Figs. 1 and 2.

This section begins with a brief review of the hidden Markov model and highlights a key concern with HMM, namely window sizing. The rest of the section provides details on the proposed algorithm and how it addresses the HMM window sizing problem. Throughout this paper, *primitive* is used to refer to a motion type or label. *Template* refers to the model of the primitive, and includes both a feature model and an HMM. *Exemplar* refers to the training data that is used to derive the template.

### A. Hidden Markov Model

The hidden Markov model [18] is a stochastic model where the process being modeled is represented by an evolving unobservable state. This underlying state is inferred by the probabilistic relationship between the hidden state and its corresponding observable output. The state has the Markov property, *i.e.*, the next state depends only on the current state. HMM stores its model information as a set of three variables:

- The initial state distribution ($\pi$) represents the probability that an observation sequence begins in a given state. This parameter is important in a fully connected or a cyclic HMM (a given state can transition into any other state). In a left-right HMM (a given state can transition only to itself, or advance to the next state), this variable is less significant, as for the left-right HMM, it is typically assumed that each observation sequence begins in the initial state.
- The state transition probability matrix ($A$) represents the probability that a given state will transition into another state, or remain in the current state. If the probability that the state will remain in itself $a_{ii}$ is high, this implies that the motion is slow-moving.
- The observation probability ($B$) represents the observation model for each state. For continuous variables, Gaussian or mixture of Gaussian distribution models are typically used.

For the proposed algorithm, a left-right HMM is used, since it is expected that in a given template, the motion will progress in a sequential pattern. Therefore $\pi$ is set to $[1; 0; 0; ...]$. The observation data corresponds to joint angles, while the hidden states represent key poses in the primitive. $A$ represents the stochastic first order dynamic model governing the transition between key poses. Since the state data is continuous, $B$ is represented by a set of multivariate Gaussians, where each Gaussian mean represents the pose of the associated hidden state, and the covariance matrix represents the variance and covariance.

Given a set of exemplars, the Baum-Welch algorithm [18] is utilized to train the template model. A given HMM's ability to represent, identify and reproduce a primitive is dependent on its model initialization and the number of states used. A model with a small number of states may not carry enough resolution to fully model the primitive. A model with a large number of states has heavy computational costs and may cause the HMM to become over fitted [18]. Testing with the collected dataset has shown that 6 to 10 states results in similar levels of segmentation accuracy, similar to previous findings for human motion primitive modeling [24]. An 8 state system was selected as the optimal compromise between the time needed to perform template training and segmentation accuracy.

To assess the similarity between a trained model and a new observation sequence, the forward algorithm [18] is applied. The forward algorithm calculates the likelihood that the ob-

servation data could have been generated by the model. The forward algorithm requires $O(N^2T)$ calculations, where $N$ is the number of states and $T$ is the observation length [18]. In order to compare the continuous time series data against the template, a candidate section must be selected. A simple sliding window of fixed length $W$ is not optimal, since intra- and interpersonal differences may result in a different time taken to complete each motion. A second alternative is to test a set of windows of variable size, $W + nW_{ext}$, where $n$ is a multiplier on the fixed-window series, and $W_{ext}$ is the length of addition to the observation window added. Both windowing methods are computationally expensive as they require the forward algorithm to be applied $T - W$ times for a single fixed window, or $(T - W) \times n$ for multiple fixed-windows for each HMM template.

The forward algorithm also implicitly penalizes longer observations. Given the HMM model $\lambda$, the probability $P(O|\lambda)$ that a given observation sequence $O$ of length $T$ was generated by the model is computed as follows:

$$Pr(l|\lambda) = \pi_1 a_{1,2} a_{2,3} \cdots a_{T-1,T} \quad (1)$$

$$Pr(O|l,\lambda) = b_1(O_1) b_2(O_2) \cdots b_T(O_T) \quad (2)$$

$$Pr(O|\lambda) = Pr(O|l,\lambda) Pr(l|\lambda) \quad (3)$$

where $l$ is a given state sequence, and $a$ and $b$ are the probability of $l$ and $O$ occurring, given $\lambda$. Since $a$ and $b$ are both probability values (hence a value between 0 and 1), the longer the state sequence, the smaller the probability becomes [18]. To account for this, the likelihood measure can be normalized by the length of observation data, or the observation data can be resampled so that all data passed into the forward algorithm are of the same length. The latter approach is adopted herein.

### *B. Feature Searching and Template Training*

Instead of using fixed window sizing, this paper proposes a feature searching method, to provide an initial estimate of possible windowing edges. The exemplar motions are scanned for key features, such as velocity peaks or ZVCs, as a way to approximate a potential interest region in the observed data. The forward algorithm can then be performed to assess the likelihood that this region was generated by one of the known motion templates, thus significantly reducing the number of times the forward algorithm needs to be executed.

It is assumed that ZVCs start and terminate any given primitive [13]. In this algorithm, a ZVC is defined as one of the following: 1) the velocity crosses from positive to negative, 2) the velocity crosses from negative into positive, or 3) the absolute mean value of the velocity over a small window is below a threshold [14]. Using velocity features allows the algorithm to locate the general shape of the motion of interest more robustly then algorithms that rely on distance measures [22].

Human motions, particularly typical rehabilitation exercises, are characterized by periodic velocity patterns, such as flexion and extension, making velocity peaks and zero crossings a salient indicating feature. A typical feature template of a rehabilitation motion would consists of a ZVC, then a positive or negative peak, then another ZVC, the opposite peak, and a final ZVC. The feature extraction and matching is based on the assumption that such a velocity profile sequence can be extracted and used in the template exemplars and observations. The template training procedure is summarized in Fig. 3.

*1) HMM template training (Fig. 3.5-6):* The exemplars are used to train HMMs with the Baum-Welch algorithm [18], creating an HMM for each primitive. An 8 state left-right model is used. The Gaussian observation functions are initialized by *k*-means clustering. Log-likelihood (LL) threshold values ($T_R$), used to recognize template motions in the observation data, are derived by taking the average of the LL values of the training exemplars evaluated on the HMM, shifted by a scaling factor.

*2) Significant DoFs (Fig. 3.7):* In order to improve the proposed algorithm's robustness in the presence of high dimensionality data, a DoF feature selection routine is implemented. Rehabilitation motion tends to be focused on improving the range of motion of the injured joints, thus it can be assumed that the joints undergoing the largest range of motion are the significant ones. The significant DoFs of a given template are selected by calculating the standard deviations of the template joint velocities and grouping them via *k*-means clustering, with $k = 2$. The DoFs that are in the cluster with the highest centroid are assumed to be significant for that particular template. To further reduce the feature space, the significant DoFs are checked for correlation. If any of the DoFs are found to be over 90% correlated, the redundant DoFs are removed.

*3) Velocity aggregation (Fig. 3.8):* The velocities of the significant DoFs for a given template are multiplied together, to create a scalar estimate of the overall velocity and estimate the ZVCs and velocity peaks. With this approach, the subsequent algorithm steps are the same regardless of the dimensionality of the original motion sequence, allowing the algorithm to handle both single-DoF and multi-DoF motions in the same way. It is also simpler to threshold on a single aggregated velocity (AV) signal instead of on several independent DoFs.

*4) Velocity feature extraction (Fig. 3.9-16):* Although typical rehabilitation movements exhibit a two-peak feature, the proposed algorithm does not automatically assume that all motions are two-peaked. Rather, the template training sequence searches for ZVCs in the AV of each exemplar, and examines the peak magnitude between each ZVC. If the velocity peaks between any two ZVCs are small, defined as a percentage of the maximum or minimum peak over the whole exemplar, that peak is rejected as a feature. Otherwise, the velocity peak sign, magnitude ($v_p$) and the time point when it occurred is noted. The sequence of ZVCs and velocity peak signs are utilized as the primary feature used for feature matching. The peak time is used to calculate peak-to-peak lengths ($t_{pp}$). $v_p$ and $t_{pp}$ are used to reject significantly smaller motions from triggering a feature match in the template matching section.

```
for all Motion Templates do
    for all Exemplar data do
        Filter data via Butterworth filter
    end for
    Train HMM via Baum-Welch with exemplar data
    Calculate LL threshold with exemplar data
    Calculate significant DoFs via k-means with exemplar velocities
    Compute AV from significant DoFs
    for all Exemplar data AV sequences do
        Locate all ZVCs in AV sequence
        Check peak values between ZVCs
        Discard small peaks
        Note all peak signs, magnitudes and times
        Peak signs sequence stored for feature matching
    end for
    Select majority peak signs sequence
end for
```

Fig. 3. Template training algorithm pseudocode

Each of the exemplar motions will have its features extracted in this manner, and the template characteristics, that is, its sequence of ZVCs and velocity peak signs over each of the exemplars are compared. If the majority of the exemplars have identical template characteristics, then that characteristic sequence is used. If an agreement cannot be made, a template training failure is reported, as this suggests that the exemplars consist of differing motions, and that the exemplars should be reviewed for consistency.

### C. Feature Guided Segmentation

Once the templates are prepared, the algorithm can be used to segment observation data. This process is outlined in graphical form in Fig. 5; pseudocode is provided in Fig. 4.

*1) Segmentation preparation (Fig. 4.2-3):* During the online segmentation phase, a small sliding window is passed over the observation data, noting the local peak values and ZVCs of each of the DoFs. An AV stream is computed for each template, based on the template's significant DoFs.

*2) Feature searching (Fig. 4.2-3):* A ZVC is declared if the AV makes a zero-crossing, or if is very low for several timesteps [14]. Local peak values are tracked by an internal buffer. If the current window has a peak value higher than the stored peak value, the peak value in the buffer is updated accordingly. To avoid noise spikes in the velocity data from affecting the template matching, the peak buffer value is attenuated if it does not contribute to a match after several seconds, to prevent a large spike in the velocity from preventing feature matches.

If a given AV observes a sequence of ZVCs and peaks that matches its corresponding template, then the algorithm has located a potential segment point. Using features to estimate the start and end time of a segment candidate also accounts for temporal variations. This way, the general shape is always matched, regardless of the time it took the subject

```
for all = 1 → T do
    Filter observation data via Butterworth filter
    Create AV streams for each template
    Search for ZVC and peaks in each AV stream
    Discard small velocity peaks
    Attenuate old peaks
    if Detected features == exemplar features then
        Resample windowed data
        Determine LL via forward algorithm
        Highest LL above LL threshold is segment
    end if
end for
```

Fig. 4. Segmentation algorithm pseudocode

to perform the motion. The velocity magnitudes and peak-to-peak distance must exceed $v_p$ and $t_{pp}$ respectively in order for a potential segment point to be declared. This prevents noise, such as when the subject is stationary, from triggering the feature match.

Several ZVCs before and after the located feature peaks are marked as possible segmentation bounds, and all possible candidate windows are compared to the HMM templates via the forward algorithm. ZVCs that are very close together, and are close to small feature peaks are likely due to noise or tremors, thus only ZVCs that are near significantly sized peaks are kept.

*3) HMM matching (Fig. 4.8-10):* The template and window edge combination that results in the highest likelihood value over the threshold $T_R$ is declared a segment. All window edge combinations are resampled so they are of equal length, to prevent the forward algorithm from favouring shorter sequences. Following HMM template matching, the recorded peak magnitude and ZVCs are reset, and the feature search resumes at the next timestep.

## IV. METHODS

To validate and analyze the performance of the proposed algorithm, two types of movements were considered: 1) lower-body rehabilitation movements, and 2) whole body movements. The verification data came from three sources: 1) the University of Waterloo (UW), where 20 healthy subjects performed rehabilitation movements (Section IV-A.1), 2) the Toronto Rehabilitation Institute (TRI), where 4 rehabilitation subjects performed rehabilitation movements (Section IV-A.2), and 3) a publicly available dataset from the University of Tokyo (UT), where 1 healthy subject performed full-body movements (Section IV-B.1) [19].

### A. Lower Body Rehabilitation Movements

Both the UW and the TRI datasets were recorded with a set of SHIMMER IMUs [25], mounted on the hip, knee and ankle. These devices recorded accelerometer and gyroscope data of thigh and calf movement during the participants' exercises. The sampling frequency was 128 Hz. The data was translated to joint angles via an extended Kalman filter (EKF) [26], with adaptive noise parameters and anthropometric joint

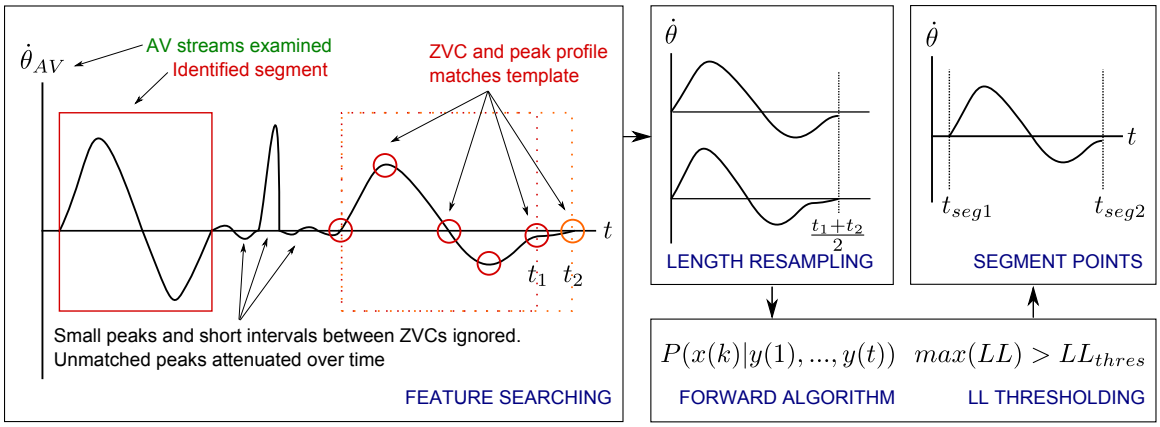


Fig. 5. Segmentation algorithm. The observation velocity streams are converted into AV streams based on each template's significant DoFs. Velocity features are accepted or rejected based on the peak size or peak-to-peak interval data collected from the exemplars. Some subjects gradually slow to a stop, so low velocity for a period of time is also considered as a ZVC. Several ZVCs are collected to form windows. The windows are resampled to be equal length. Forward algorithm is used to assess the LL, which is used to determine which window is a segment.

limits to reduce drift [27]. The human leg was modeled as an articulated chain of rigid bodies, where each rigid body corresponds to a leg limb. A 5-DoF model was employed. For motions where the torso was stationary, the hip was modeled as a 3-DoF joint and the knee modeled as a 2-DoF joint. For motions where the foot was stationary, the ankle was modeled as a 3-DoF joint and the knee was a 2-DoF joint. The RMS error of the EKF system is approximately 6.5° [27]. Prior to segmentation, the estimated joint angles were low pass filtered with a 4 Hz Butterworth filter for the UW dataset and 1 Hz Butterworth filter for the TRI dataset.

For the UW dataset, motion capture data was collected simultaneously, providing Cartesian space data for manual segmentation. However, a motion capture system was not available at TRI, thus Cartesian space data was generated from the EKF joint estimates. For both datasets, ground truth manual segmentation was determined by a single human observer using video playback of the Cartesian data. The manual segments were denoted based on the observer's perception of when a repetition had ended and when the next repetition began, based on movement speed and body position.

*1) University of Waterloo (UW) dataset:* The algorithm was tested on 20 healthy subjects from UW, performing five types of lower-body physiotherapy exercises: knee extensions while seated (KEF), sit-to-stand (STS), squats (SQD), knee/hip flexion while supine (SKH) and hip flexions while supine (SHF). These exercises were selected as they are commonly employed in the physiotherapy clinic for lower-body rehabilitation [28]. The average age of the participants was 22 years old. No participant had any lower back or leg injuries in the past six months. For each participant, 2 sets of 10 repetitions were collected, with each set consisting of a single exercise type. The experiment was approved by the University of Waterloo Research Ethics Board, and signed consent was obtained from all participants.

*2) Toronto Rehabilitation Institute (TRI) dataset:* To assess the generalizability of the proposed algorithm to rehabilitation patients, motion data from 4 total joint replacement patients undergoing post-operative physiotherapy were collected at TRI. The patients performed knee extension exercises while seated. The patients were tracked from the first day of admission until discharge, and had daily rehabilitation sessions. The experiment was approved by the University of Waterloo Research Ethics Board and the University Health Network Research Ethics Board, and signed consent was obtained from all participants.

### B. Whole Body Movements

*1) University of Tokyo (UT) dataset:* To investigate performance on full body motions, the algorithm was also applied to a publicly available motion dataset from Kulić *et al.* [19], originally collected at UT. The dataset contains 751 full-body exercise motions, spanning nearly 17 minutes of motion capture data. The data set consists of a single healthy demonstrator performing a random sequence of 45 different movement primitives. On average, each movement was executed 20 times. The most common 8 motions were selected for segmentation: bowing (BAD), both arm raise to 90° (BAR90) or 180° (BAR180), left arm raise (LAR90), right arm raise to 90° (RAR90) or 180° (RAR180) and squatting (SQD). The data was provided as joint angles of a 20-DoF kinematic model, collected at 100 Hz.

### C. Analysis Procedures

All processing and algorithmic implementation were carried out in MATLAB 7.12. The HMM functions were implemented with Murphy's HMM MATLAB Toolbox [29]. 8-state left-right HMMs were used, trained using the Baum-Welch algorithm. $\pi$ was initialized with $[1; 0; 0; ...]$, as typical for left-right HMMs. The initial $\mu$ was calculated by separating the templates into 8 sequential equally sized segments, and calculating the mean for each DoF within each segment. The initial $\Sigma$ for each state was set to a diagonal matrix of 10, and allowed to converge after several iterations, as some $\Sigma$ values suggested by separating the data into 8 segments

and calculating the variance resulted in numerical issues with the training algorithm.

An algorithmic segmentation point was declared correct if it falls within $\pm t_{error}$ of a manual segment point. It is a false positive if an algorithmic segment point was declared when there is not one. It is a false negative if a segment point was not declared when there should be one. Each segment's two segment points are awarded points separately. That is, a completely correct segment could receive +2 correct, whereas a partially correct segment could receive +1 correct and +1 false negative.

## V. EXPERIMENTAL RESULTS

For the UW dataset, two scenarios are investigated: 1) individual templates (Section V-A), where templates are trained for each subject using the training data, and then tested on unseen data of the same subject; and 2) generic templates (Section V-B), where the template is constructed from data of 3 or 5 participants, and then tested on the remaining 15 participants. For the TRI dataset (Section V-C), templates generated from the healthy participant data as described in Section V-B are used on the rehabilitation data. For the UT dataset (Section V-D), templates from the single participant were generated from parts of the observation sequence, and tested on the full length of the sequence, segmenting both training and novel data. A comparison to other algorithms using the UW dataset (Sections V-E and V-F) was also performed.

### *A. Segmentation using Self-templates for the UW Dataset*

The data streams used for the feature selection were the 5 joint angles obtained from an EKF joint estimator [27]. For each subject, the first of the two collected motion sets was used for the template training, then both motion sets were used separately for testing the proposed algorithm. In this set of experiments, the templates used were subject specific, so the templates generated from one subject's motions were used to segment only their own motion.

Figs. 6 (sit-to-stand) and 7 (squats) show various motions segmented by the algorithm. The boxes indicate the algorithmic segmentation points, while the circles indicate manual segmentation points. The top plot shows angle data; the bottom plot shows angular velocity. The dark lines are DoFs denoted as significant across all the templates, whereas the lighter lines are not significant.

Using only the joint angles, the segmentation accuracy for the feature-guided HMM with $t_{error}$ = 0.2$s$ was found to be 76% of the segments for data not used during training. If $t_{error}$ = 0.3$s$, the accuracy increases to 84%. If segmentation accuracy is ignored and only identification accuracy is examined, the algorithm reports 95% accuracy. However, utilizing only joint angle data sometimes leads the identification process to confuse two similar actions. For example, squats and sit-to-stand both involve a large change in the knee sagittal angle and a small change in the ankle sagittal angle. Using joint data only, the squat position is sometimes confused with the seated position, causing misidentification.

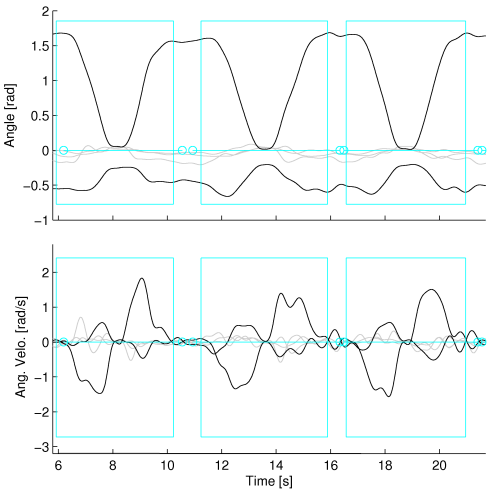


Fig. 6. Joint angles (top) and angular velocity (bottom) of a subject performing sit-to-stand, from the UW dataset. The circles denote the manual segments, whereas the rectangles denote the algorithmic segments. In this figure, only one of the six segment points is declared a correct segment under $t_{error}$ = 0.2$s$, even though the algorithm located all the motions. Reaction speed of the human observer is a common source of the mismatch between the automatic and the manual segmentation, as the manual segments sometimes lag behind the time when the participant ceases motion. This can be seen most strongly between the second and third segment. Examining the space between the first and second segment, it can be noted that the participant does not actually stop moving between repetitions, making the location of the true segment point ambiguous, especially if the viewing angle is not optimal.

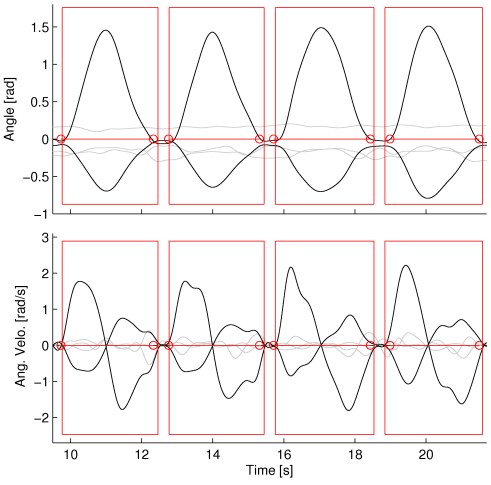


Fig. 7. Joint angles (top) and angular velocity (bottom) of a subject performing squats, from the UW dataset. The circles denote the manual segments, whereas the rectangles denote the algorithmic segments. Despite significant motion in the other DoFs, the algorithm successfully segments the motion under examination.

TABLE I

SEGMENTATION AND IDENTIFICATION RESULTS OF THE PROPOSED ALGORITHM ON THE UW DATASET WITH JOINT ANGLES AND ACCELEROMETER DATA. THE MOTIONS ARE KNEE EXTENSION (KEF), SIT-TO-STAND (STS), SQUATS (SQD), KNEE-HIP FLEXION (SKH) AND HIP FLEXION (SHF), WITH THE ALGORITHM'S PERFORMANCE WITHIN A GIVEN ERROR BOUND $t_{error}$, AND THE NUMBER OF SEGMENT POINT ESTIMATES THAT ARE CORRECT (C), FALSE POSITIVE (FP) AND NEGATIVE (FN), AS A PERCENTAGE [%] OF THE TOTAL NUMBER OF GROUND TRUTH SEGMENT POINTS. SEPARATE RESULTS ARE REPORTED FOR DATA USED FOR TRAINING (TR) AND DATA THAT HAVE NOT BEEN USED FOR TRAINING (UTR).

| | | Id. | $t_{error}$ = 0.2$s$ | | | $t_{error}$ = 0.3$s$ | | |
|---|---|---|---|---|---|---|---|---|
| Data | Motion | C | C | FP | FN | C | FP | FN |
| TR | KEF | 100 | 94 | 0 | 5 | 96 | 0 | 4 |
| | STS | 100 | 66 | 7 | 34 | 82 | 2 | 18 |
| | SQD | 100 | 83 | 0 | 18 | 92 | 0 | 8 |
| | SKH | 100 | 82 | 0 | 18 | 91 | 0 | 9 |
| | SHF | 100 | 87 | 1 | 13 | 93 | 0 | 7 |
| | **Total w/o STS** | **100** | **86** | **0** | **14** | **93** | **0** | **7** |
| | **Total** | **100** | **82** | **2** | **18** | **91** | **1** | **9** |
| UTR | KEF | 100 | 95 | 0 | 5 | 96 | 0 | 4 |
| | STS | 100 | 52 | 16 | 48 | 77 | 5 | 23 |
| | SQD | 100 | 80 | 0 | 20 | 88 | 0 | 12 |
| | SKH | 100 | 78 | 0 | 22 | 86 | 0 | 14 |
| | SHF | 100 | 79 | 3 | 21 | 86 | 0 | 14 |
| | **Total w/o STS** | **100** | **83** | **1** | **17** | **89** | **0** | **11** |
| | **Total** | **100** | **77** | **4** | **23** | **87** | **1** | **13** |

Adding the acceleration signal introduces strong additional orientation-based data into the HMM vectors, reducing this misidentification. These results, shown in Table I, show that including these accelerometer terms raises the segmentation accuracy, particularly for the squat and sit-to-stand motions, resulting in an overall segmentation accuracy of 87% from 84%. These results also demonstrate the capability of the proposed algorithm to handle a variety of input time series data, including both joint angle and Cartesian acceleration measurements. Table I also shows that the identification accuracy also increases to 100%.

### *B. Segmentation using Inter-personal Templates for the UW Dataset*

In a clinical setting, it is impractical to have templates from each patient. The movement patterns of a rehabilitation patient will change as the patient's condition improves from post-surgery recovery to discharged, so any template and algorithm used for segmentation must be able to handle large intra- and inter-personal variations. A second set of experiments was carried out to assess the algorithm's ability to segment if a single common template was used for all observed data.

The participant data were randomly separated into groups of 3 or 5, and templates were generated from these groups. The accelerometer data were rotated before HMM training to align the mounting orientation of the sensors between participants. All other parameters were identical to the ones used in Section V-A.

TABLE III

SEGMENTATION AND IDENTIFICATION RESULTS OF SEGMENTATION PERFORMED ON THE TRI DATASET, UNDER DIFFERENT $t_{error}$ VALUES, WITH HEALTHY INTER-SUBJECT TEMPLATES. THE MOTION EXAMINED IS KNEE FLEXION. THE NUMBER OF SEGMENT POINTS THAT ARE CORRECT (C), FALSE POSITIVE (FP) AND NEGATIVE (FN), AS A PERCENTAGE OF THE TOTAL NUMBER OF GROUND TRUTH SEGMENT POINTS [%], ARE REPORTED.

| | Id. | $t_{error}$ = 0.2$s$ | | | $t_{error}$ = 0.3$s$ | | |
|---|---|---|---|---|---|---|---|
| Patient | C | C | FP | FN | C | FP | FN |
| 1 | 100 | 74 | 5 | 26 | 87 | 3 | 13 |
| 2 | 100 | 66 | 9 | 34 | 84 | 7 | 16 |
| 3 | 84 | 34 | 38 | 66 | 43 | 19 | 57 |
| 4 | 99 | 64 | 8 | 34 | 81 | 1 | 18 |
| **Total** | **98** | **64** | **11** | **35** | **79** | **6** | **21** |

Table II shows the mean and standard deviation of the accuracy over all the template groups. The data used for template training were not used for the segmentation testing. The algorithm demonstrates good robustness when using participant-independent rather than participant-specific templates, dropping approximately 5% for overall accuracy for $t_{error}$ = 0.2$s$ and 6% for overall accuracy for $t_{error}$ = 0.3$s$ compared to the single-template results from Section V-A. Identification accuracy drops by 1%. Similar to the results in Section V-A, the sit-to-stand results were noticeably worse. With exception of the sit-to-stand, the 5-template set outperforms the 3-template sets in all the motions. This is to be expected, since the 5-template set includes a wider variety of exemplar motions, and thus is more able to account for inter-participant variability.

### *C. Segmentation using Inter-personal Templates for the TRI Dataset*

The 5-person template set generated from the healthy participant data was applied to the TRI dataset and is shown in Table III. Since only knee extension motions were examined, only the knee extension template was used. The velocity thresholds for the feature detection was rescaled based on the data of each individual patient. The velocity thresholds are normally set to be a percentage of the maximum velocity observed in the exemplars. However, healthy participants exhibits higher velocities than rehabilitation patients, so for each rehabilitation patient, the maximum velocity of the observation data was used to rescale the thresholds. Motions that were poorly recovered from EKF, or consisted of failed attempts to perform the movement due to patient pain or unsteadiness were removed from consideration. The algorithm successfully identified 98% of the primitives. At $t_{error}$ = 0.2$s$, the segmentation accuracy is 64%. At $t_{error}$ = 0.3$s$, the segmentation accuracy increases to 79%.

### *D. Segmentation using Self-templates for the UT Dataset*

The UW and TRI datasets examined only consisted of lower body rehabilitation in sequences where each motion set contained the same motion type. The UT dataset provides a

TABLE II

SEGMENTATION AND IDENTIFICATION RESULTS ON THE UW DATASET, WITH IDENTIFICATION ACCURACY PERCENTAGE [%], AS WELL AS SEGMENTATION PERFORMANCE UNDER DIFFERENT $t_{error}$ VALUES, WITH INTER-SUBJECT TEMPLATES. THE MOTIONS ARE KNEE FLEXION (KEF), SIT-TO-STAND (STS), SQUATS (SQD), KNEE-HIP FLEXION (SKH) AND HIP FLEXION (SHF).

| | Identification | | | | $t_{error}$ = 0.2$s$ | | | | $t_{error}$ = 0.3$s$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 3-template set | | 5-template set | | 3-template set | | 5-template set | | 3-template set | | 5-template set | |
| Motion | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| KEF | 99.1 | 1.9 | 99.4 | 0.9 | 85.3 | 2.0 | 86.6 | 1.6 | 86.7 | 2.1 | 88.0 | 1.8 |
| STS | 99.4 | 1.2 | 98.3 | 1.9 | 53.1 | 4.1 | 53.2 | 8.8 | 70.7 | 6.5 | 69.8 | 11.3 |
| SQD | 99.7 | 0.2 | 98.8 | 0.9 | 73.6 | 2.4 | 74.6 | 3.4 | 81.2 | 2.8 | 82.6 | 3.2 |
| SKH | 99.2 | 1.2 | 99.0 | 1.2 | 74.3 | 8.8 | 74.4 | 11.9 | 82.4 | 7.6 | 82.4 | 12.4 |
| SHF | 100.0 | 0.1 | 100.0 | 0.0 | 79.5 | 2.1 | 83.8 | 1.1 | 85.7 | 2.7 | 90.6 | 1.1 |
| **Total w/o STS** | **99.5** | **0.5** | **99.3** | **0.5** | **78.9** | **1.2** | **81.0** | **1.5** | **84.3** | **1.7** | **86.6** | **2.2** |
| **Total** | **99.5** | **0.4** | **99.1** | **0.8** | **73.1** | **1.5** | **74.6** | **3.0** | **81.2** | **2.0** | **82.8** | **3.5** |

whole-body motion dataset where the demonstrated movements are in arbitrary order, demonstrating that different motion types can be recognized simultaneously.

Template training was performed by selecting 8 exemplars from an excerpt of the observation sequence, then the algorithm was applied to the whole observation. The segmentation results can be found in Table IV, with the results of the segmentation accuracy for the data used for training separated from the data not used for training. At $t_{error}$ = 0.2$s$, the algorithm reports a segmentation accuracy of 79%. It increases to 84% if the error bound is expanded to 0.3$s$. Identification accuracy is at 95%. These results demonstrate that the algorithm can be applied to full body movements and to observation sequences that contain movements in arbitrary order.

### *E. Accuracy Comparison to Other Algorithms using UW Dataset*

In order to compare the proposed approach to existing work, a ZVC method, described in [14] was also implemented. Segmentation points are declared when the velocity crosses zero, thus a segment window is formed with two consecutive ZVCs. Segment windows that are too small are rejected. To reduce the amount of spurious ZVCs, only ZVCs that occurred on the significant DoFs were considered. Since ZVC produces segment points at both the end of flexion and the end of extension, every two consecutive ZVC segment windows were combined.

A fixed-sliding window HMM [18] was also implemented. HMM construction for the fixed sliding window was identical to the feature guided HMM. The fixed window length was computed from the lengths of the exemplar templates. Segmentation points were declared on local maximas of the likelihood, as long as it is above some likelihood threshold.

Lastly, a DTW algorithm [9] was also implemented. An arbitrary exemplar is selected to be the template motion. A mapping matrix, used to time-align two time series signals, is constructed by taking the Euclidean distance between each time step. An unconstrained path was used, since motion data often contains pauses, which is less suited for a greedy algorithm [30]. The warping path width was constrained to 8 timestamps to prevent singularities. A segment candidate is considered a primitive candidate as long as the Euclidean distance between a given template and the observation data was maintained below some empirically-derived threshold. The threshold is determined by taking the distance between all the exemplar motions against the template exemplar, and taking the average distance. If the observation data is of a different length then that of the template, a penalty distance is added, proportional to the difference in length. Initial tests with a bottom-up DTW [10] resulted in a segmentation accuracy of 50%, with an average runtime of 700 seconds, indicating that basic DTW is not suitable for this application. To reduce the DTW computation time and improve accuracy, a modified version of DTW was implemented. The feature-searching component (Fig. 4.2-6) from the proposed algorithm was used to determine the potential segment points, and

TABLE IV

SEGMENTATION AND IDENTIFICATION RESULTS FOR THE PROPOSED ALGORITHM ON THE UT DATASET [19], UNDER DIFFERENT $t_{error}$ VALUES. THE MOTIONS ARE BOWING (BAD), BOTH ARM RAISE TO 90° (BAR90) OR TO 180° (BAR180), LEFT ARM RAISE (LAR90), RIGHT ARM RAISE TO 90° (RAR90) OR TO 180° (RAR180) AND SQUATTING (SQD). THE TABLE REPORTS THE NUMBER OF SEGMENT POINT ESTIMATES THAT ARE CORRECT (C), FALSE POSITIVE (FP) AND NEGATIVE (FN), AS A PERCENTAGE OF THE TOTAL NUMBER OF GROUND TRUTH SEGMENT POINTS [%]. SEPARATE RESULTS ARE REPORTED FOR DATA USED FOR TRAINING (TR) AND DATA THAT HAVE NOT BEEN USED FOR TRAINING (UTR).

| | | Id. | $t_{error}$ = 0.2$s$ | | | $t_{error}$ = 0.3$s$ | | |
|---|---|---|---|---|---|---|---|---|
| Data | Motion | C | C | FP | FN | C | FP | FN |
| TR | BAD | 100 | 86 | 0 | 14 | 86 | 0 | 14 |
| | BAR180 | 92 | 77 | 8 | 23 | 88 | 8 | 12 |
| | BAR90 | 94 | 83 | 0 | 17 | 89 | 0 | 11 |
| | LAR90 | 100 | 84 | 16 | 16 | 97 | 16 | 3 |
| | RAR180 | 91 | 80 | 0 | 20 | 86 | 0 | 14 |
| | RAR90 | 82 | 82 | 9 | 18 | 86 | 9 | 14 |
| | SQD | 95 | 95 | 0 | 5 | 95 | 0 | 5 |
| | **Total** | **94** | **83** | **5** | **17** | **90** | **5** | **10** |
| UTR | BAD | 88 | 69 | 0 | 31 | 69 | 0 | 31 |
| | BAR180 | 89 | 75 | 0 | 25 | 86 | 0 | 14 |
| | BAR90 | 100 | 88 | 0 | 13 | 94 | 0 | 6 |
| | LAR90 | 100 | 100 | 0 | 0 | 100 | 0 | 0 |
| | RAR180 | 93 | 75 | 0 | 25 | 84 | 0 | 16 |
| | RAR90 | 100 | 86 | 0 | 14 | 91 | 0 | 9 |
| | SQD | 100 | 70 | 10 | 30 | 70 | 10 | 30 |
| | **Total** | **95** | **79** | **1** | **21** | **84** | **1** | **16** |

TABLE V

SEGMENTATION AND IDENTIFICATION RESULTS COMPARING THE PROPOSED (FEATURE HMM) TO OTHER ALGORITHMS (ZVC [14], FIXED HMM [18] AND FEATURE DTW [10]) USING THE UW DATASET, WITH IDENTIFICATION ACCURACY, AS WELL AS THE ALGORITHM'S PERFORMANCE WITHIN A GIVEN ERROR BOUND $t_{error}$, AND THE NUMBER OF SEGMENT POINT ESTIMATES THAT ARE CORRECT (C), FALSE POSITIVE (FP) AND NEGATIVE (FN), AS A PERCENTAGE OF THE TOTAL NUMBER OF GROUND TRUTH SEGMENT POINTS [%]. SEPARATE RESULTS ARE REPORTED FOR THE DATA USED FOR TRAINING (TR) AND DATA THAT HAVE NOT BEEN USED FOR TRAINING (UTR).

| | | Id. | $t_{error}$ = 0.2$s$ | | | $t_{error}$ = 0.3$s$ | | |
|---|---|---|---|---|---|---|---|---|
| Data | Algorithm | C | C | FP | FN | C | FP | FN |
| TR | ZVC | - | 61 | 26 | 38 | 69 | 21 | 31 |
| | Fixed HMM | 99.7 | 24 | 46 | 76 | 35 | 32 | 65 |
| | Feat. DTW | 100 | 81 | 2 | 19 | 89 | 1 | 11 |
| | Feat. HMM | 100 | 82 | 2 | 18 | 91 | 1 | 9 |
| UTR | ZVC | - | 66 | 21 | 34 | 74 | 17 | 26 |
| | Fixed HMM | 99.5 | 20 | 47 | 80 | 28 | 36 | 72 |
| | Feat. DTW | 99.9 | 76 | 5 | 24 | 85 | 2 | 15 |
| | Feat. HMM | 100 | 77 | 4 | 23 | 87 | 1 | 13 |

the HMM component in the proposed algorithm was replaced with DTW. That is, potential segment points were located by considering the sequence of velocity peaks and zeros in the observation data, while rejecting short ZVC intervals or small velocity peaks. When a sequence of velocity features matches that of a given template, a segment potential is declared and assessed by the DTW. This also allows the HMM and the DTW to be compared in a more straight forward fashion.

The 5 joint angles from the EKF and the accelerometer signal were used for template training and observation input for the fixed-window HMM and the DTW. Only the joint angle data were used for the ZVC algorithm as the accelerometer data would introduce noisy ZVCs and degrade the segmentation quality. Table V shows the feature-guided HMM's error metrics compared against ZVC, fixed-window HMM and feature-guided DTW, at different $t_{error}$ levels. The ZVC algorithm does not have a training component, but the data are presented separately for ease of comparison. Fixed-window HMM and ZVC individually do not perform as well as the combined proposed algorithm, but the proposed algorithm performs similarly to the feature-guided DTW.

### *F. Timing Comparison to Other Algorithms using UW Dataset*

Table VI shows the timing results for each of the examined algorithms. The exemplar length is the average length of each observation sequence. The template construction time is the average time taken to construct all relevant template data for a given subject. The segmentation time is the average time required to segment one set of observation data.

Due to its simplicity, the ZVC algorithm requires no template training time and very little segmentation time. However, as noted in the previous section, this algorithm is inaccurate. Fixed-window HMM uses the Baum-Welch algorithm to train the HMMs based on exemplar data, and

TABLE VI

TIMING RESULTS COMPARING THE PROPOSED (FEATURE HMM) TO OTHER ALGORITHMS (ZVC [14], FIXED HMM [18] AND FEATURE DTW [10]) USING THE UW DATASET, IN [S]. AVERAGE EXEMPLAR LENGTH IS $37.14\ s \pm 11.03\ s$.

| | Template construction | | Segmentation time | |
|---|---|---|---|---|
| Algorithm | Mean | SD | Mean | SD |
| ZVC | - | - | 0.13 | 0.04 |
| Fixed HMM | 37.92 | 8.51 | 54.27 | 33.53 |
| Feature DTW | 70.33 | 30.76 | 76.58 | 69.02 |
| Feature HMM | 38.25 | 9.59 | 6.72 | 2.56 |

requires a significant amount of training time. Its segmentation time is also very long, as it needs to run the forward algorithm numerous times at each time step, once for each template available. The feature-guided HMM requires the most training time, though the feature extraction component takes very little time additional time. With the feature-guiding, the proposed algorithm is able to more intelligently determine when to apply the forward algorithm, and decrease the segmentation runtime significantly, when compared to the fixed-window HMM. Even with added feature matching, DTW still requires considerably longer to execute when compared to HMM and typically require more time than the duration of the temporal sequence. Although the segmentation accuracy of the DTW is comparable to that of the HMM, the high computational cost of the DTW makes it impractical for any real-time applications.

The feature-guided HMM takes, on average, 6.7 seconds to segment an average of 10 repetitions, or 0.6 seconds per repetition.

## VI. DISCUSSION

### *A. Segmentation Accuracy*

The improvement in accuracy from 77% to 87% from expanding $t_{error}$ from 0.2$s$ to 0.3$s$ in Table I implies that a number of the algorithmic segments are just outside of the manual segment error bounds. Several different factors contribute to this result. Since there are no standardized guidelines on how a motion sequence should be segmented, manual segmentation is typically left up to the perception of the human observer. Although manual segment points are typically used as ground truth, there may be inconsistencies between observers or even between exemplars for the same observer. It can be difficult to visually determine when a segment has started or ended, due to stray motion or tremors at segment ends, or due to poor camera angle during video replay, causing manual segments to be declared before or after the actual end of motion. The observer may also be segmenting on DoFs that are not the most significant. At times, the algorithm overestimates the bounds of the segments, as the motion may have been perceived to have ended due to slow velocity before the actual ZVC.

Examining these results shows that the sit-to-stand motion performs particularly poorly in terms of segmentation accuracy. A detailed examination of the sit-to-stand results shows that the segmentation algorithm produces results comparable

to the other motions, but a mismatch between the locations of the manual and algorithmic segmentations causes the algorithm to report a lower accuracy (see Fig. 6). We believe this is due to issues with the viewing angle during manual segmentation, especially if the participant does not perform a complete stop between repetitions. Comparing $t_{error}$ = 0.2$s$ to $t_{error}$ = 0.3$s$, the sit-to-stand accuracy reported for sit-to-stand for both the dataset used for training and the dataset not used for training in Table I rises to from 66% and 52% to 82% and 77% respectively. If sit-to-stand is not included, the total accuracy for these two tables rises from 91% and 87% to 93% and 89% for $t_{error}$ = 0.3$s$, respectively.

It is uncertain if a segmentation system providing 79-83% accuracy for generic templates is sufficient for rehabilitation exercise analysis. Such systems are not yet in widespread usage. Interviews with physiotherapists have not yielded much insight into accuracy requirements because these tools are uncommon, and the benefits of having automatic segmentation systems are not fully realized. Physiotherapists have noted that a method to automatically count the number of repetitions completed and time-to-completion for each motion is beneficial for diagnosis. The high identification accuracy of the proposed algorithm can readily provide accurate count estimates, but further improvements can be made to increase the segmentation accuracy.

Since automated segmentation methods are new in the physiotherapy clinic, it is desirable to produce a system that is not only fast and accurate, but also easy to use, to encourage usage by non-specialists such as physiotherapists. It is for this reason that *a priori* knowledge of the exercises was not used for the training templates, and that template profiles are determined automatically. This would allow physiotherapists to provide templates through simple demonstrations or patient recordings. Template quality would likely improve if prior knowledge (*e.g.*, likely locations of ZVCs or indications of key joints) was provided *a priori*.

As template training is an off-line step, inter-personal and generic templates would be preferable over patient-specific templates. To investigate the generalizability of the templates, the same template set obtained from healthy participants was used for both populations. Table II reports identification accuracy of 99% on healthy participants whereas Table III reports a identification accuracy of 98%. This suggests that the proposed algorithm successfully identifies data from both healthy and rehabilitation subjects. However, segmentation accuracy decreased from 75% in Table II for healthy subjects to 64% in Table III for rehabilitation subjects using the more strict 0.2$s$ error bound, and from 83% to 79% using the 0.3$s$ error bound. Given the large differences in motion variability between the two populations, these results appear promising. It is important to note that the TRI sample is small and thus may be difficult to generalize. Additional validation is required on a larger population of patients with a larger number of movements. The TRI dataset is also challenging as it includes patient motions from the first day of admission (*i.e.* shortly after surgery) until discharge, so the TRI dataset exhibits higher variability and less smoothness in the motion, when compared to the UW dataset. Templates created from patient data may improve the segmentation results, but this is difficult to test due to the low number of patient movement data collected.

Several limitations of the algorithm prevent it from being applicable to general human movement analysis. The feature-guided HMM assumes that segment points only occur on ZVCs. For rehabilitation motions, where the exercises tend to consist of flexion and extension motions, ZVCs are a reasonable assumption, but may not apply to more complicated motions. Another limitation is the feature-based thresholds. Though the thresholds described in Fig. 4 are automatically generated from the exemplars, several patient movements consisted of velocities that dropped below the thresholds, thus the feature detection did not locate the motion without some form of adjustment. A method to adjust these thresholds on-line, based on observation data, would help to tailor the generic templates to suit the movement profiles of individual patients.

The computational costs of the feature-guided HMM are small enough to allow for segmentation and identification during on-line observation, as the result can be provided 0.6 seconds following the completion of the exercise. The time taken for a repetition is 3.7 seconds on average. For rehabilitation patients, the pause between exercises is typically larger than this, allowing the system to provide feedback before the next repetition is started or during initiation of the subsequent repetition.

### *B. Comparison to Other Segmentation Algorithms*

Although feature-guided HMM segments along ZVC points, its ability to reject spurious crossings compared to the ZVC algorithm greatly improves the segmentation accuracy and reduces the number of false positives. The ZVC algorithm often groups a long sequence of slow moving motions with the first half of the motion of interest, causing the rest of the observation sequence to be segmented improperly. If a segment was redefined to separate the flexion from the extension motion, the ZVC algorithm would report better performance. A single threshold to reject short sequences is also difficult to use, as each participant moves at a different speed. No identification results are reported for the ZVC algorithm as it does not provide any motion identifications.

Fixed-window HMM also performed poorly. Its inability to change its window size, which is the average length of the templates, means that even though it is good at identifying the underlying motion, it is not able to produce accurate segmentation bounds. It does provide accurate identifications, and emphasizing the need for a windowing technique.

It does not appear that DTW warping irregularities is a significant problem for this data set, as feature-guided DTW performs comparably to the feature-guided HMM in both segmentation and identifying. Having a multi-tier segmentation algorithm and a way to reduce the search space for potential segment points greatly increases the accuracy of the segmentation algorithm. Due to variations in human motion, it is critical that the algorithm employed can handle spatial

and temporal intra- and interpersonal variances between the motion template and observations. HMMs model spatial variations explicitly as state variance, and temporal variations within the state transition matrix, allowing it to identify the observation motion, even if it is carried out faster or slower than the exemplar template. ZVC does not account for these variabilities at all.

## VII. CONCLUSIONS AND FUTURE WORKS

This paper proposes an on-line motion segmentation algorithm that can segment human motion data. The algorithm is shown to be able to segment data with high accuracy, and is capable of handling any time series input stream, including joint angles or Cartesian linear acceleration data. The ability to segment on-line enables immediate feedback to the physiotherapist and rehabilitation participant, allowing the participant to correct incorrect motions to maximize rehabilitation efficacy.

The current algorithm has been tested with a healthy participant population and a limited patient population. In the future, we plan to test the algorithm with more data from patient populations, and to continue work in improving the segmentation accuracy when a single set of templates are used for multiple participants. We also plan to collaborate with physiotherapists to determine the requirements for segmentation accuracy in clinical practice.

## VIII. ACKNOWLEDGMENT

The authors would like to thank the participants at the University of Waterloo, as well as the physiotherapists and patients of the Toronto Rehabilitation Institute for providing the time and data utilized for this work.

## REFERENCES

[1] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese, "On the representation, learning and transfer of spatio-temporal movement characteristics," *International Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 613–636, 2004.

[2] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 359–372.

[3] K. Lorincz, B. R. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh, "Mercury: A wearable sensor network platform for high-fidelity motion analysis," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, CA, 2009, pp. 183–196.

[4] D. Roetenberg, P. J. Slycke, and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *IEEE Transactions on Biomedical Engineering*, vol. 54, pp. 883–90, 2007.

[5] H. Zhou, H. Hu, N. D. Harris, and J. Hammerton, "Applications of wearable inertial sensors in estimation of upper limb movements," *Biomedical Signal Processing and Control*, vol. 1, pp. 22–32, 2006.

[6] M. G. Pandy, "Computer modeling and simulation of human movement," *Annual Review of Biomedical Engineering*, vol. 3, pp. 245–273, 2001.

[7] J. F. S. Lin and D. Kulić, "Automatic human motion segmentation and identification using feature guided hmm for physical rehabilitation exercises," in *Workshop on Robotics for Neurology and Rehabilitation, IEEE International Conference on Intelligent Robots and Systems*, San Francisco, CA, 2011, pp. 33–36.

[8] ——, "Segmenting human motion for automated rehabilitation exercise analysis," in *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2012, pp. 2881–2884.

[9] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[10] E. J. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, pp. 349–371, 2003.

[11] C. A. Ratanamahatana and E. Keogh, "Making time-series classification more accurate using learned constraints," in *Proceedings of the SIAM International Conference on Data Mining*, 2004, pp. 11–22.

[12] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 1st SIAM International Conference on Data Mining*, Chicago, IL, 2001, pp. 1–11.

[13] M. Pomplun and M. Matarić, "Evaluation metrics and results of human arm movement imitation," in *Proceedings of 1st IEEE/RAS International Conference on Humanoid Robotics*, Cambridge, MA, 2000.

[14] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[15] J. Lieberman and C. Breazeal, "Improvements on action parsing and action interpolation for learning through demonstration," in *Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots*, 2004, pp. 342–365.

[16] N. Koenig and M. J. Matarić, "Behaviour-based segmentation of demonstrated tasks," in *Proceedings of the International Conference on Development and Learning*, 2006.

[17] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 14, 2002, pp. 793–800.

[18] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[19] D. Kulić, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.

[20] S. Chiappa and J. Peters, "Movement extraction by detecting dynamics switches and repetitions," in *Proceedings of the Advances in Neural Information Processing Systems*, 2010, pp. 388–396.

[21] F. Bashir, W. Qu, A. Khokhar, and D. Schonfeld, "Hmm-based motion recognition system using segmented pca," in *Proceedings of the IEEE International Conference on Image Processings*, 2005, pp. 1288–1291.

[22] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proceedings of the IEEE International Conference on Data Mining*, 2001, pp. 289–296.

[23] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Proceedings of the IEEE International Symposium on Wearable Computers*, 2005, pp. 160–163.

[24] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[25] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer: A wireless sensor platform for noninvasive biomedical research," *IEEE Sensors Journal*, vol. 10, pp. 1527–1534, 2010.

[26] G. Welch and G. Bishop, "An introduction to kalman filters," University of North Carolina at Chapel Hill, Tech. Rep., 2006.

[27] J. F. S. Lin and D. Kulić, "Human pose recovery using wireless inertial measurement units," *Physiological Measurement*, vol. 33, no. 12, pp. 2099–2115, 2012.

[28] Toronto Rehabilitation Institute, "Hip and knee exercises."

[29] K. Murphy, "Bayes net toolbox for matlab," code.google.com/p/bnt/, 1998, last accessed on 2012/04/27. [Online]. Available: code.google.com/p/bnt/

[30] R. Turetsky and D. Ellis, "Ground-truth transcriptions of real music from force-aligned midi syntheses," in *Proceedings of the 4th International Symposium on Music Information Retrieval*, 2003, pp. 135–141.