Pose Estimation and Segmentation for Rehabilitation

by

Vladimir Joukov

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2015

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Parts of this thesis have been adapted from works ([34, 33, 43]) I authored and co-authored with Jonathan Lin who is a PhD student under the supervision of Professor Dana Kulić at the Adaptive Systems Laboratory at Waterloo University, Dr. Michelle Karg who at the time was a postdoctoral fellow in the Adaptive Systems Laboratory, Dr. Vincent Bonnet who is a postdoctoral fellow at Tokyo University of Agriculture and Technology in Japan, and Gentiane Venture who is an associate professor at Tokyo University of Agriculture and Technology.

Chapter 4 is adapted from the work titled "Online tracking of the lower body joint angles using IMUs for gait rehabilitation" published in IEEE Engineering in Medicine and Biology Conference 2014 [34] which was completed under the supervision of Dr. Michelle Karg and Professor Dana Kulić who contributed with ideas, discussions, and manuscript review.

The methods and findings presented in chapter 5 are adapted from the work titled "Rhythmic EKF for Pose Estimation during Gait" accepted for publication in IEEE-RAS International Conference on Humanoid Robots 2015 [33]. Professor Dana Kulić and Dr. Michelle Karg contributed with ideas and discussions, research and development progress supervision, and manuscript review. Professor Gentiane Venture and Dr. Vincent Bonnet suggested related works and reviewed the manuscript.

Chapter 6 is adapted from work titled "Human motion segmentation by data point classification" published in IEEE Engineering in Medicine and Biology Conference 2014 [43]. This work was initially started as a course project to satisfy the requirements of the STAT 841 graduate course. The development was completed with Jonathan Lin in equal parts and adapted into the publication with guidance from Professor Dana Kulić.

Abstract

The global population is getting older and the aging demographic is increasing demands on health-care industry. This will drive the demand for post stroke, joint replacement, and chronic disease management rehabilitation. Currently physiotherapists rely on mostly subjective and observational tools for patient assessment and progress tracking. This thesis proposes methods to enable the use of non-intrusive, small, wearable, wireless sensors to estimate the pose of the lower body during rehabilitation and extract objective performance measures useful for therapists.

Two different kinematic models of the human lower body are introduced. The first approach expresses the body position and orientation in the world frame using three prismatic and revolute joints, while the second switches the model's base between the right and the left ankle during gait. An Extended Kalman Filter (EKF) is set up to estimate the joint angles, velocities, and accelerations of the models using measurements from inertial measurement units. The state update model assumes constant joint acceleration and is linear. Measurement prediction, relating the joint positions, velocities and accelerations to the measured angular velocity and linear acceleration at each IMU, is done using forward kinematics, using one of the two proposed kinematic models. The approach is validated on healthy participant gait using motion capture studio data for ground truth comparison. The prismatic and revolute model achieves better Cartesian position accuracy in the swing leg due to a shorter kinematic chain, while the switching base model improves the stance leg Cartesian estimate and does not allow measurement noise to accumulate as drift in global position, knee joint angle root mean squared errors (RMSE) of 6.1° and 5.6° are attained respectively by the models.

Next the Rhythmic Extended Kalman Filter (R-EKF) algorithm is developed to improve pose estimation. It learns a model of rhythmic movement over time based on harmonic Fourier series and removes the constant acceleration assumption. The estimated phase and frequency of the motion also allow the proposed approach to segment the motion into repetitions and extract useful features such as gait symmetry, step length, and mean joint movement and variance. The algorithm is shown to outperform the EKF in simulation, on healthy participant data, and stroke patient monthly assessments. For the healthy participant marching dataset, the R-EKF improves joint acceleration and velocity estimates over regular EKF by 40% and 37% respectively, estimates joint angles with 2.4° RMSE, and segments the motion into repetitions with 96% accuracy.

While the proposed R-EKF effectively segments rhythmic rehabilitation movement such as gait, not all rehabilitation motions are rhythmic or may have uneven delays between repetitions by regimen design or due to fatigue. For such motions a time-series

segmentation as data point classification algorithm is proposed. Common dimensionality reduction and classification techniques are applied to estimated joint angle data to classify each time-step as a segment or non-segment point. The algorithm is tested on five common rehabilitation exercises performed by healthy participants and achieves a segmentation accuracy of 82%.

Acknowledgments

I would like to thank my supervisor, Professor Dana Kulić, for the support and guidance throughout my time as a student. Thank you for introducing me to research while I was still in undergraduate studies, for all the encouragement and advice over the last two years, and for hours spent reviewing my last minute writing. Thank you for the exciting research projects, industrial collaborations, and travel opportunities. Your dedication to your students is truly astonishing.

I would also like to thank Jonathan Lin and Michelle Karg, you were always available for discussions and without your ideas this project would not be possible. Working with you has been a real pleasure.

A big thank you to all the members of adaptive systems lab. Our research meetings have greatly expanded my knowledge on countless topics. Thank you also for always being the first to volunteer in user studies.

The data collected at Grand River Hospital was critical in completion of this research. I would like to thank Karen Guha for her expertise and help in the process.

Finally, I would like to thank my family and friends for their support and encouragement.

Dedication

To my parents and Yana.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Physiotherapy is a type of rehabilitation that aims to restore a patient's quality of life after an injury, surgery, or stroke by improving their mobility. Through prescribed exercises and specialized equipment, physiotherapy helps the patient to regain their muscle strength, range of motion, and natural movement. Injuries or illnesses requiring physiotherapeutical treatment often afflict the elderly. In many developed countries the elderly population is rapidly growing. In Canada the number of seniors has increased by 27% since 2001 and one in four citizens will be over the age of 65 by 2036 [54]. Demand for services related to care for the elderly, including physiotherapy, is expected to rapidly grow [32]. Joint replacement surgeries are primarily performed on elderly patients with joint degradation due to osteoarthritis or joint injury due to a fall [22]. The risk of stroke doubles for each decade after the age of 55 and nearly three quarters of patients are over 65 [72]. Given the increased demand physiotherapy practice is changing in significant ways. As the number of patients grows, therapists will not be able to work one on one with a patient and instead will supervise a group. Preventative and self-managed care is also expected to become more popular; patients will be expected to follow exercise regiments on their own time to manage chronic diseases, reduce hospital visits, and improve quality of life [32].

Typically when a new patient enters physiotherapy the first session is an assessment of the patient's movement abilities. This includes a combination of questionnaires and visual observation of the patient's movement, such as the Berg Balance test [7]. Next the therapist designs a customized exercise regimen for the patient; a large part of the regimen will be assigned to be performed on the patient's own. During each clinical session, the therapist observes and physically assists the patient to ensure all exercises are completed correctly. Subsequent assessments are done monthly to track the patient's

progress and the exercise regimen is modified accordingly. Unfortunately only rudimentary tools are available to the therapists for assessment and monitoring of patients. Currently physiotherapists rely on observation gait analysis which has been shown to have high inter-rater variability even between experts [14] and have fairly low agreement with motion capture kinematics data [69]. Another common technique is goniometry [52], which measures the total range of motion of a specific joint, however it cannot be applied while the patient is performing and exercise and does not show the over all motion of the patient. For wrist displacement, goniometry accuracy compared to motion capture is approximately 7 degrees [20]. Lavernia *et al.* [41] evaluated the accuracy of therapist observation and goniometry measurements for knee range of motion assessment using x-ray imaging. Their analysis shows that on average physiotherapists tend to underestimate the range of motion by 9 degrees relying purely on observation and by 7 degrees when using a goniometer. Furthermore few tools are available for patients outside the clinic. Without feedback patients may perform exercises incorrectly and prolong the rehabilitation period. Thus there is a need for an automatic assessment system that continuously tracks patient movement and is able to provide real time feedback to both the therapist and the patient. This system must be at least as accurate as goniometry, run in real-time for visual feedback, and segment exercises into repetitions to allow extraction of useful quantitative assessment measures.

1.1 Thesis Contributions

On-line Pose Estimation during ambulatory movement from wearable IMU sensors: The ability to walk is fundamental for post-surgery and post-stroke patients to regain independent living. Previous approaches for gait rehabilitation can only estimate the foot orientation and position and not full lower body pose [58, 50] and use off line post processing methods to extract joint angles [78]. Existing kinematic model based approaches are not suitable for gait as they assume a stationary base [44, 21]. This thesis builds upon the earlier fixed-base approach [44] and proposes two modeling approaches to model a patient's lower body using branched kinematics with a moving base and an online pose estimation algorithm which accurately estimates the patient's lower body pose (joint angles, velocities, and accelerations) using the kinematic model and non-intrusive wearable inertial measurement units (IMU).

Online gait motion model learning for improved pose estimation and segmentation: Apart from online pose estimation it is important to learn the motion model of the patient to segment gait into cycles and extract objective measures. Previous works

[28, 25, 19] assume that joint angles have been already estimated and then learn the model. There is no approach that combines pose estimation with motion model learning. We combine an online periodic motion learning method [56] with the extended Kalman filter to allow the pose estimation algorithm to learn the motion model online and use the model to improve pose estimation for periodic exercises such as gait. It is also accurately segments rhythmic exercises into repetitions. From the estimated pose and segmentation, objective measures for the physiotherapists including range of knee motion, symmetry between legs, stride length, and exercise execution time are extracted. These can greatly improve the therapists' ability to asses and monitor patients.

Motion Segmentation as time-series classification: For non rhythmic motions a segmentation as time-series classification algorithm is developed. This approach allows well developed classifier algorithms to be applied to segment time-series data. Previous motion segmentation methods with no *a priori* knowledge of the motion use time-series features to declare segment points, they tend to over segment, and cannot handle circular movements. Template based approaches learn a per exercise template and rely on it to segment the exercise or detect exercise change. Using dimensionality reduction the proposed algorithm does not require training for each exercise separately and generalizes between motions and subjects.

1.2 Thesis Outline

Chapter 2 provides the overview of existing work focused on automated rehabilitation. First, existing pose estimation algorithms using wearable sensors are discussed. Next, various generative motion model learning approaches are presented. Finally, works on classification of time-series data into exercise type and segmentation of estimated pose into exercise repetitions is presented.

Chapter 3 provides the background information on the algorithms and mathematical tools used in this thesis. The operational principles of inertial measurement units and a calibration algorithm are presented. A brief introduction to model state estimation using the Kalman and extended Kalman filters is provided. Robotic modeling of kinematic chains is introduced. Finally various dimensionality reduction and classification algorithms are presented.

Chapter 4 presents two approaches to model the human lower body as a kinematic chain. One approach utilizes three prismatic and three revolute joints to represent the global position of the lower body in space, the other allows switching between the right

and left ankles as the patient changes their stance leg during gait. The Extended Kalman filter (EKF) is set up to estimate joint angle positions, velocities, and accelerations using wearable sensors placed at the waist, knees, and ankles. An approach to detect heel strikes during gait for correct ankle switching is discussed. The pose estimation algorithm is compared to motion capture and the performance using both kinematic models is compared to ground truth motion capture data.

Chapter 5 introduces the Rhythmic Extended Kalman Filter (Rhythmic-EKF), which can learn a periodic motion and over time improve the pose estimation over EKF. By estimating the phase and frequency of the motion, Rhythmic-EKF is also able to segment periodic motions into repetitions. Rhythmic-EKF is shown to outperform EKF for pose estimation in simulation, healthy participants marching on the spot, and on stroke patient data collected during a 6 minute walk test. Rhythmic-EKF is also shown to successfully extract valuable information for patient progress monitoring. The approach is validated in simulation, on healthy participant data, and stroke patient rehabilitation over 5 months. It is shown to track joint angles to 2.4° and extract measurements such as stride length and mean and variance of movement.

Chapter 6 formulates time-series segmentation as a classification problem. This allows for use of well developed dimensionality reduction and classification algorithms to segment non rhythmic motions. The proposed segmentation algorithm is validated on a dataset of 20 healthy participants performing 5 common rehabilitation exercises.

Chapter 7 summarizes the findings and outlines directions for future work.

Chapter 2

Related Work

Many research groups have considered improving physiotherapy through the use of technology. This chapter overviews previous works on pose estimation, motion modeling, and motion segmentation. A variety of different sensing technologies have been proposed for pose estimation in the rehabilitation setting. Motion models are useful for comparing the patient's movement with a healthy exemplar and progress tracking. Segmenting the estimated pose into repetitions is necessary to extract useful measures for the therapists.

2.1 Pose Estimation

This section overviews common motion measurement techniques and pose estimation algorithms focusing on the use of wearable inertial measurement units (IMU).

A variety of technologies are available for pose estimation. In the film industry and biomechanics research, multi-camera based systems are popular, the cameras track multiple reflective markers attached to the actor and can very accurately estimate their position. Inverse kinematics is commonly done in post processing to convert marker Cartesian positions into joint angles. However these systems require line of sight for each camera and cannot be used in a clinical setting due to multi room environments, and occlusions caused by training equipment, other patients, or therapists. Due to the high cost and extensive calibration requirements of the cameras they are also not suitable for home-based rehabilitation.

Depth camera pose estimation, typically using the Microsoft Kinect camera [64], has also been proposed for rehabilitation purposes. Huang used the Kinect to detect whether the patient's movement reached a rehabilitation standard target and provided visual exemplars of the rehabilitation movement to patients. Clinical results show that the system decreased work load for the therapists and increased the patient's motivation to participate in rehab [27]. Lange *et al.* developed a game based balance rehabilitation system using the Kinect. The player's avatar travels through a mine collecting gems and placing them into a cart. By collecting gems in specific patterns common rehabilitation motions are achieved [39]. Kinect based systems are useful for assessing the general trends of the motion but are not sufficient to extract accurate objective measures useful for therapists since they work well only when the patient is directly facing the camera and struggle with self-occlusion by other body parts during complex motions [53]. Xu *et al.* [73] analyzed Kinect's joint angle estimation accuracy for gait on a treadmill and showed it can only achieve a RMSE of 28.5° and 11.8° in the hip and knee joints respectively. To improve the accuracy Bonnet used Kinect's image data to track the Cartesian location of custom markers placed on the shoulder, elbow, and hand. Then the depth data of the markers and an optimization algorithm was used to compute the joint angles. While accuracy improved the system remains non-portable and suffers from occlusions caused by equipment and other patients or therapists.

Zhou *et al.* compared different technologies proposed for pose estimation in a rehabilitation setting. Inertial based sensors were shown to be highly accurate, compact, and cost effective [77]. Bergman analyzed preferences of clinicians and patients to determine design criteria for wearable non-invasive rehabilitation sensors [9]. Patients focus on the size and aesthetics of the sensor. They prefer small devices with sleek designs that do not affect normal daily behavior and are simple to use. Clinicians would like a system that provides them with information about patient progress, monitors the patient at home reducing hospital visits, and is simple to use. Wearable wireless inertial measurement units fit the desired design criteria perfectly. With advancements in microelectromechanical system (MEMS) manufacturing they can be made very small to not restrict the patient's movement, can wirelessly stream data in real time to provide feedback, and work in any environment. The remainder of this section reviews works focused on IMU based pose estimation.

2.1.1 Accelerometer Based Pose Estimation

In many rehabilitation exercises such as stair ascent and gait the healthy motion is performed primarily in a 2D plane. By assuming that the motion is limited to a 2D plane

and is slow enough to ignore centripetal acceleration, the pose estimation problem is greatly simplified. Then accelerometers can be used as inclonometers and measure the incline of each link based on the gravity vector. Bergmann [8] used these assumptions and was able to accurately estimate the incline of the thigh and shank during stair ascent. Their algorithm performed with an average root mean square error (RMSE) of 4 degrees for planar motions. Low *et al.* proposed using wearable accelerometers to track forearm flexion [46]. The kinematic model consists of a planar shoulder and elbow joint and each angle is directly measured assuming that acceleration other than gravity is negligible. They focus on the design of the wireless sensors and do not provide an estimate of system measurement accuracy.

Using accelerometers as inclinometers is very limited since only movements in a plane perpendicular to gravity can be tracked; even in the plane movements where centripetal acceleration is significant cannot be handled. It is possible to perform integration and double integration of the accelerometer signal to estimate velocity and position respectively, however this results in poor accuracy with low cost MEMS sensors due to noise.

2.1.2 Inertial Sensor Pose Estimation

To track fast motions it is possible to add gyroscopes which measure the angular velocity of the limbs. However gyroscopes tend to have a constant bias and integrating the measurement accumulates error over time [11].

Many previous works have proposed using IMUs to track position and orientation of rigid bodies independently, ignoring human kinematics. Sabatini segmented the gait pattern into phases using angular velocity and computed the initial orientation of each link during the motionless phase between heel strike and toe off using accelerometer data and used forward-backward strap-down integration to compute the limb orientation and position during the swing phase [50]. This approach requires a separate algorithm to segment the patient's gait into phases and does not run in real time since both the previous and next foot initial orientation must be known to perform strap down integration. Bonnet *et al.* used a weighted Fourier linear combiner to model gyroscope measurements and integrate the Fourier series into a pose estimate of a single IMU attached to the trunk. Ignoring the zero frequency Fourier series weights allows for drift free integration. This method however does not perform well without a priori knowledge of the motion until the weights are learned and requires the motion to be quasi-periodic.

An at home knee replacement surgery rehabilitation system was developed by Ayoade and Baillie using an IMU sensor attached at the knee and ankle [2]. The system provides color coded correctness of movement feedback to the patient by segmenting healthy movement into three regions and counts repetitions. The accuracy of pose estimation was not tested. Their study shows that IMU systems are accepted by users and improve rehabilitation progress results after 6 weeks of rehabilitation.

Rampp *et al.* [58] focused on estimating only the stride length and time during gait. They segmented each stride into events based on the gyroscope z axis data to compute stride time. Next the gyroscope signal was integrated to obtain the orientation of the ankle during each stride. Using the orientation the gravity vector is calculated and subtracted from accelerometer measurements, drift is corrected using foot stance times. Finally the accelerometer is double integrated to estimate stride length. While the method can estimate stride length with an accuracy of 8cm which can be useful to the therapists, it only provides the ankle orientation and not the entire lower body pose.

Zhou [78] also employed strap down integration to compute the rotation of each link of the human arm. They subtracted the gravity vector from accelerometer measurements and applied double integration. To deal with drift they used Lagrangian optimization with kinematic constraints. The optimization successfully deals with accelerometer drift due to double integration but does not reduce the gyro drift and requires computationally expensive polynomial regression.

Schwartz et al. [62] approached the problem from a machine learning perspective. They used a motion capture studio to record joint angles and IMU sensor data while performing preset activities and applied Gaussian Process Regression to learn the general mapping between IMU sensor data and joint angles for each activity and individual separately. To determine what mapping to use when new sensor data is obtained they used a multi-class support vector machine to learn the classification of activities using sensor data. This method performs with an average joint angle error of 5.6 degrees. Unfortunately it does not generalize to the whole space of human motions. Also it cannot generalize to new subjects since the mappings for each activity must be learned per subject using a motion capture studio.

Many of the methods described above treat each link as a separate rigid body and compute their orientations independently without considering human kinematic constraints. The orientation of each rigid body is converted to joint angles in a post processing step that requires optimization and may lead to solutions which are not physically attainable. Lin and Kulić used a kinematic model and a Extended Kalman Filter (EKF) to estimate arbitrary 3D leg motion [44]. The states of the EKF are the joint positions,

velocities, and accelerations, while the measurement vector consists of the accelerometer and gyroscope data from sensors attached above the knee and ankle. The EKF inherently deals with sensor noise and produces joint estimates directly, which can be used for real time measurement and feedback. To battle gyroscope drift potential fields were applied if the joint angle estimate left a defined range. Unscented Kalman filter was used by El-Gohary and McNames to estimate the shoulder and elbow joint angles however they did not implement kinematic constraints [21]. Both methods assume that the base of the kinematic model is fixed and cannot handle motions such as gait.

2.2 Motion Modeling

Accurately modeling the estimated motion is very useful for patient assessment and progress tracking. Comparing a learned model of the patient's movement with a healthy exemplar can be used as a measure of the patient's performance. The motion model can also be used to improve pose estimation performance. It is possible to identify the performed motion by comparing it with multiple known models. Segmentation into repetitions can be performed with a motion model that incorporates temporal features. Motion modeling has been explored in the field of robotics and human-computer interaction (HCI). Robotics researchers typically aim to create generative models so human like motion can be reproduced on a robot; this is usually done by learning the motion from multiple human demonstrations. HCI focuses on using motion models for recognition and segmentation; primarily discriminative models are used. This section reviews generative models, section 2.3 discusses using motion models for segmentation purposes.

To handle temporal and spatial changes in the motion Ijspeert *et al.* used dynamic movement primitives (DMP) to model goal directed behavior with a non-linear system. The dynamics of the motion trajectory are represented as a linear spring damper system with a learned non-linear forcing term. This approach is useful due to scaling and invariance properties of the system. Scaling the time constant and forcing terms does not effect the shape of the trajectory but changes the speed and spatial scale. The forcing term can also be used for motion identification by comparing the learned weights of known motions with a previously unseen motion [28]. A rhythmic version of DMP was also developed based on a limit cycle system and online learning approaches have also been proposed [61].

A combination of an online rhythmic learning model based on Fourier series and DMP has been proposed by Gams [25]. The performed rhythmic motion is modeled

as a sum of multiple adaptive frequency phase oscillators and the lowest non-zero frequency oscillator is used as the phase in learning the rhythmic DMP. The DMP invariance properties are later used to reproduce scaled versions of the motion. Petric [56] noted that using multiple oscillators with different frequencies makes the adaptation process more difficult and learning the DMP is redundant in the system. Instead he used a harmonic Fourier series leaving only a single frequency and phase to learn the motion. The weights of the Fourier series can later be used for motion identification and the estimated phase for segmentation.

It is also possible to decompose the motion into smaller probabilistic primitives. Calinon *et al.* [19] projected joint angles and end effector position into a lower dimensional space using principal component analysis and used Gaussian mixture models to learn the motion in the latent space. To include temporal variations in the movement the temporal information was included as an extra dimension in the mixture model. Gaussian mixture regression was used to construct a generalized motion in the latent space and projected into joint space. Billard used hidden Markov models (HMM) to imitate motions [10]. HMM represents the movement as a set of states and probabilities of transitioning between them which allows for better temporal variability modeling than Gaussian mixture models. After training on a number of examples the HMM can produce a generalized version of the motion using the Viterbi algorithm. Kulić and Nakamura [38] proposed an on-line learning method based on a tree like structure of HMM movement primitives. Time-series data is segmented into potential movement primitives, the movement primitives are compared to existing HMM nodes. If the potential primitive is similar to an existing HMM node, that node is reinforced. Otherwise a new HMM node is added. A higher order HMM models the grouping of motion primitives into movement patterns.

2.3 Classification and Segmentation

A fully automated rehabilitation system must make sure the patient performs assigned exercises, count the number of repetitions of each, and extract objective measures of performance. To achieve this the system needs to recognize the exercise and segment it into repetitions. Classification is the process of identifying the performed motion from a set of known labels and segmentation is the process of finding the start and end points of each repetition of the motion from time series data as well as the rest time in between. Classification and segmentation of motions from joint angles remains a difficult problem. A full body human model can have a large number of joint angles

and only a few may be active for a particular motion leading to multiple dimensions of the feature vector not providing useful information. Furthermore motions between participants often have high variance and thus it is difficult to develop algorithms that generalize to unseen individuals. Even a single participant's motion can greatly vary over time due to fatigue, an injury, or movement improvement.

Classification of motion has been a primary focus of gesture recognition which allows people to interact with computers in a more human like and intuitive fashion. Many previous works have focused on identifying handwriting gestures [57, 4, 16]. Recently, wearable devices are becoming more popular as an interaction method. Thalmic Labs developed an armband that recognizes 7 gestures based on EMG sensors [60]. MIT Media Lab created a framework for gesture recognition using IMU sensors [6]. Lara *et al.* [40] surveyed human activity recognition methods based on wearable sensors. Most systems first extract features from the raw time-series sensor data, redundant or irrelevant features are removed using feature selection techniques. Next a model is trained using the remaining features and machine learning methods. To recognize the activity from new sensor data, features are extracted and the closest matching model is found. Decision trees [1], k-nearest neighbour [47] and neural networks [45] have all been used to label the underlying motion. It is possible to use these approaches to perform motion segmentation but only when a transition from one motion to another occurs, they cannot segment time-series data into repetitions of the same exercise.

Segmentation of time-series data in exercise repetitions can be done using time-series features or motion model primitives. If no *a priori* knowledge about the motion is available time-series features can be used. A variance based algorithm has been proposed by Koenig and Mataric to segment time-series feature sequence [35]. Segment points were declared at local maximums of variance above a threshold. A zero velocity crossing (ZVC) suggests a movement direction change and is common at motion repetitions. ZVC has been used by Fod *et al.* to declare segment points when ZVC occurred in multiple joints within a short time-span. Although fast and easy to implement, these algorithms tend to oversegment and work only for certain types of movement. A smooth circular motion cannot be accurately segmented using variance or ZVC.

Algorithms that rely on more advanced template based techniques such as the HMM [3, 12] or dynamic time warping [29] have also been used, but they are computationally expensive, and often require a template to be generated per exercise type in order to perform segmentation. In order to reduce the complexity of the task, efforts have gone towards reducing the observation dimensionality before using the HMM [66] with some success. Several of these algorithms are sufficiently compact such that they can be utilized on-line [37, 42].

Classification techniques such as support vector machines (SVM) have also been considered. The SVM has been applied to detect large jumps in the singular value decomposition (SVD) of the observation to perform segmentation [76]. Several different SVMs can be trained to identify data as belonging to a particular class or not, and segment when the label changes [48]. Other techniques, such as boosting, look at joint angle features [49] and accelerometer spectral data [36] to propose segmentation points.

2.4 Summary

Multiple pose estimation methods have been proposed using wearable inertial measurement units for rehabilitation. Assuming the motion is slow and planar, accelerometers can be used to measure the incline of each limb. Adding gyroscopes allows pose estimation for faster motions but gyroscope bias may lead to accumulation of error. Many approaches treat each link as a separate rigid body and employ computationally expensive optimization to estimate the pose as a post processing step. Previous kinematic model based methods assume an immobile base making them useful only for exercises where the patient is stationary. In chapter 4 of this thesis a kinematic model based pose estimation algorithm is proposed which can handle a moving base and thus can be used during gait rehabilitation.

Extracting a generative motion model from the estimated pose trajectory allows motion identification, segmentation, and comparison with healthy movement. To handle spatial and temporal variability, methods based on dynamical movement primitives, Fourier series, and stochastic models have been proposed. The approaches require exemplars of the motion for learning, there is no method available that can both perform pose estimation and motion model learning simultaneously. In chapter 5 of this thesis the kinematic model based pose estimation algorithm is extended to learn a rhythmic motion model online and use the learned model to improve pose estimation, segment the time-series data into repetitions, and extract patient performance measures.

Segmentation of motion into repetitions of an exercise is an important step in extracting useful measures for therapists and providing feedback for patients. Time-series feature based algorithms tend to over-segment and template based methods require a per motion template and do not generalize to novel movements. The algorithm proposed in chapter 5 successfully models and segments rhythmic motions but cannot perform segmentation on a regimen consisting of multiple exercises. Few algorithms have been validated to generalize to previously unseen motions. Chapter 6 proposes a segmentation by time-series point classification algorithm that projects the time-series

data into a lower dimensional space and is capable of generalization between motions and participants.

Chapter 3

Background

This chapter overviews the concepts and algorithms employed in the thesis. It provides an overview inertial measurement unit operation and calibration needed for data collection and algorithm validation. The Kalman filter and its non-linear extension, used for joint angle, velocity, and acceleration estimation in chapters 4 and 5, are presented. Robot kinematics, which are applied to model the human lower body in chapter 4, are reviewed. Addaptive oscillators, which are used to learn the rhythmic model of movement in chapter 5, are introduced. Lastly, dimensionality reduction, classification, and aggregation algorithms used in chapter 6 are presented.

3.1 Inertial Measurement Unit

An inertial measurement unit (IMU) is an electronic device that uses a combination of sensors to measure its motion. They were developed for aircraft and vehicle navigation and are typically used in autopilots of airplanes, ground vehicles, and space crafts. Recent advances in microelectromechanical system (MEMS) fabrication led to miniaturization and cost reduction of IMUs. They are now finding applications outside of navigation and are small enough to be used as wearable sensors. Usually an IMU consists of several separate three dimensional sensors in a single package:

Gyroscope Measures angular velocity

Accelerometer Measures linear acceleration including gravity

Magnetometer Measures the magnetic field passing through the IMU

3.1.1 Gyroscope

Originally gyroscopes consisted of a heavy spinning flywheel rotating about an axis mounted on two rings. While the flywheel is spinning it will remain upright due to conservation of angular momentum. These mechanical devices were useful in ship navigation to find the horizon during bad visibility.

MEMS gyroscopes do not have a flywheel but instead use the Coriolis effect of vibrating masses which dictate that a vibrating object will tend to continue vibrating in the same plane even as its support rotates. To measure angular velocity two vibrating masses are oscillated in opposite directions, when the support is rotated Coriolis forces act on the masses in opposite directions. The displacement between the vibrating structures is detected as a change in capacitance proportional to angular velocity.

3.1.2 Accelerometer

The accelerometer measures linear acceleration using a known test mass. Consider a mass supported by a spring, when the mass accelerates in the direction of the spring it will compress proportionally to the applied force. Knowing the mass, the spring constant, and measuring the compression, Newtons second law can be used to compute experienced acceleration. Suspending the mass by a spring in each axis of motion allows linear acceleration to be measured in three dimensions. The MEMS accelerometer works on this principle, a test mass forms a differential capacitor with the surrounding plates and as it moves the change of capacitance proportional to linear acceleration is measured.

3.1.3 Magnetometer

Traditional magnetometers used a suspended charged object to detect the strength and direction of the earth's magnetic field for navigation. MEMS magnetometers operate based on the Hall effect; when electrons move through a conducting plate which is placed in a magnetic field they experience a Lorentz force and are deflected. This in turn will create a measurable potential difference across the plate proportional to the magnetic field. Using three conducting plates and assuming only the earth's magnetic field is present, three dimensional global orientation can be measured.

3.1.4 IMU Calibration

Our IMUs are composed of MEMS accelerometer (ADXL345) and gyroscope (L3G4200D). These devices report the measured capacitance changes as integer values. The purpose of calibration is to convert these integer values into meaningful measurements of m/s^2 and $rads/s$ in the same coordinate frame. For this biases, scaling, axis alignment, and cross sensor alignment parameters must be determined. There are commercial solutions available, such as the XSens Suit, that are pre-calibrated and are sold with the parameters stored in non-volatile memory, these solutions tend to be very expensive and the calibration parameters may change with temperature and time.

The proposed calibration relies on a motion capture studio and three motion capture markers placed on the sensor. The three markers allow us to define a coordinate frame for the IMU and compute the true orientation and angular velocity. Figure 3.1 illustrates the necessity of redefining the coordinate frame for the IMU due to misalignment of the gyroscope and the accelerometer. Using the true orientation, measured by motion capture, during static positions the accelerometer misalignment, scale, and bias are computed. The gyroscope is calibrated based on the integral of the true angular velocity. Since validation of IMU based pose estimation algorithms requires comparison with motion capture pose estimation, defining the axes of the sensor with markers greatly helps with sensor placement during experiments.

We place three motion capture markers on the sensor to define the IMU frame and mount the IMU on a tripod. Data is collected while the IMU is moved through a set of distinct static orientations. Using the markers the IMU orientations are calculated. The x axis of the IMU is defined using two of the markers for each timestep i.

$$x_i = \frac{M2_i - \frac{M3_i - M2_i}{2}}{\|M2_i - \frac{M3_i - M2_i}{2}\|} \tag{3.1}$$

Computing a normal vector to the plane defined by the three markers provides the z axis and the cross product between the two axes provides y.

$$z_i = \frac{x_i \times (M1 - \frac{M3_i - M2_i}{2})}{\|x_i \times (M1_i - \frac{M3_i - M2_i}{2})\|} \tag{3.2}$$

$$y_i = z_i \times x_i \tag{3.3}$$

Using the three axes the rotation matrix ${}^{i}R_{w}^{IMU}$ from world frame into IMU frame is

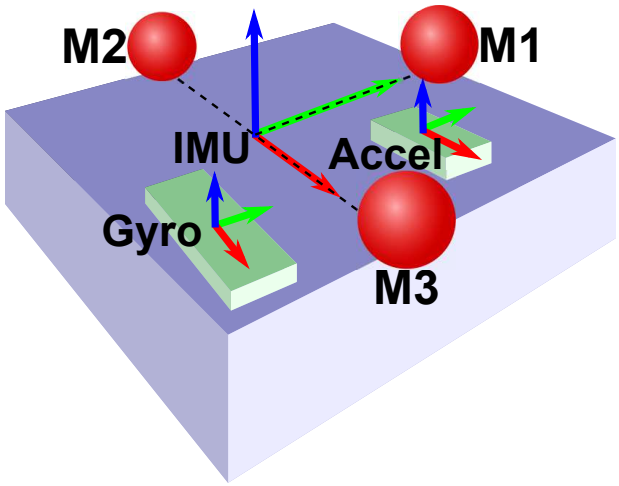


Figure 3.1: Desired IMU frame defined with three motion capture markers. Since gyroscope and accelerometer are separate chips on the IMU they will not be aligned with the desired frame or each other.

obtained for each time-step.

$$ {}^{i}R_{w}^{IMU} = \begin{bmatrix} x_i^T \\ y_i^T \\ z_i^T \end{bmatrix} \tag{3.4} $$

Next a static detector described in [67] is applied to extract static intervals for accelerometer calibration. During the static intervals centripetal acceleration is not present and the calibrated accelerometer output should be ${}^{i}R_{w}^{IMU} g$. Least squares is used to compute the accelerometer calibration matrix.

$$ {}^{i}R_{w}^{IMU} g = T_a \begin{bmatrix} a_i' \\ 1 \end{bmatrix} \tag{3.5} $$

where a_i' is the IMU accelerometer output, and T_a is a 4 by 3 calibration matrix. Figure 3.2 shows the performance of accelerometer calibration.

To calibrate the gyroscope, true angular velocity experienced by the IMU is computed from the motion capture markers. Consider the IMU as a rigid body in space, then in the IMU frame each attached marker is experiencing a linear velocity $v_{imu,i}$ and and angular velocity $w_{imu,i}$. The motion capture system provides us with the positions

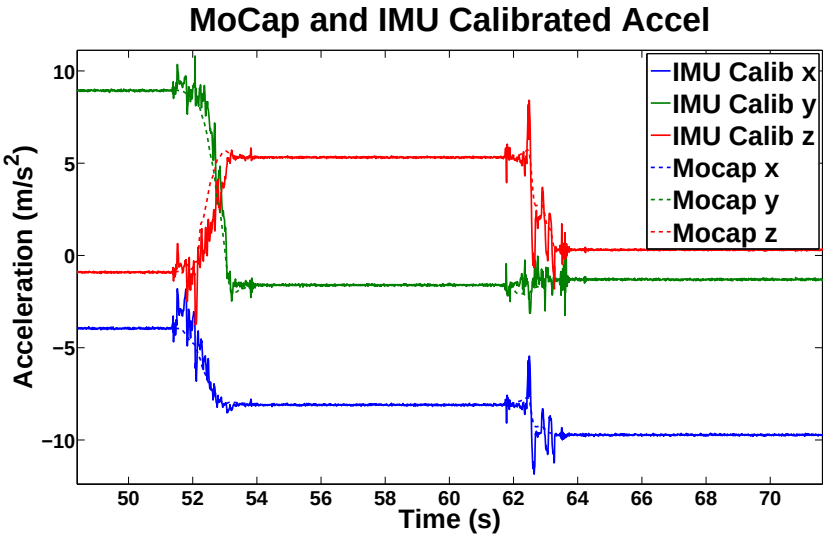


Figure 3.2: Calibrated IMU accelerometer and acceleration computed from motion capture during static intervals.

$(M1_i, M2_i, M3_i)$ and, through numerical differentiation, linear velocities $(\dot{M1}_i, \dot{M2}_i, \dot{M3}_i)$ of the markers in the world frame. For each marker the linear velocity can be expressed in terms of v_{imu} and w_{imu} in the world frame.

$$\dot{M1}_i = ({}^iR_w^{IMU})^T v_{imu,i} + M1_i \times (({}^iR_w^{IMU})^T w_{imu,i}) \tag{3.6}$$

$$\dot{M2}_i = ({}^iR_w^{IMU})^T v_{imu,i} + M2_i \times (({}^iR_w^{IMU})^T w_{imu,i}) \tag{3.7}$$

$$\dot{M3}_i = ({}^iR_w^{IMU})^T v_{imu,i} + M3_i \times (({}^iR_w^{IMU})^T w_{imu,i}) \tag{3.8}$$

Expressing the cross product in matrix form allows us to solve for w_{imu} in a least squares sense.

$$\begin{bmatrix} S(M1_i) - S(M2_i) \\ S(M2_i) - S(M3_i) \\ S(M3_i) - S(M2_i) \end{bmatrix} (({}^iR_w^{IMU})^T w_{imu,i}) = \begin{bmatrix} \dot{M1}_i - \dot{M2}_i \\ \dot{M2}_i - \dot{M3}_i \\ \dot{M3}_i - \dot{M2}_i \end{bmatrix} \tag{3.9}$$

Where S is the cross product matrix operator. During the static intervals the gyroscope should read the zero vector thus the average reading over all static intervals represents the bias B_g. The cumulative numeric integral of unbiased gyroscope readings and cumulative numeric integral of the IMU angular velocities calculated from motion capture

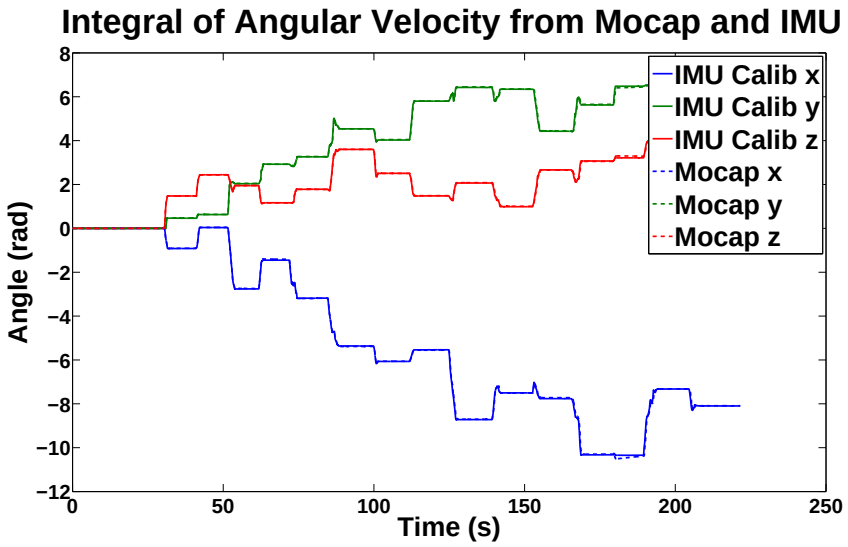


Figure 3.3: Numeric integral of calibrated gyroscope and angular velocity computed from motion capture.

are used to compute the gyroscope calibration matrix in a least squares sense.

$$\sum_{n=1}^{i} w_{imu,n} = T_g \sum_{n=1}^{i} (w'_{imu,n} - B_g) \tag{3.10}$$

Where $w'_{imu,i}$ are the integer values from the sensor and T_g is the desired calibration matrix. The numeric integral instead of original values is used to ensure small temporal misalignment between the motion capture and sensor angular velocities will not greatly reduce calibration accuracy. Figure 3.3 shows the performance of gyroscope calibration.

3.2 Kalman Filter

The Kalman Filter [71] is a popular sensor fusion technique which provides an unbiased estimate the state of a system from noisy observations. For a linear model it is shown to be an optimal filter under the assumption that both measurement and process noise are zero-mean Gaussian.

Consider a linear system of the following form:

$$s_t = As_{t-1} + w_{t-1} \tag{3.11}$$
$$z_t = Cs_t + v_t \tag{3.12}$$

At each time step t, s_t is the n-dimensional state vector which represents the true state of the system, the matrix A is the process matrix describing the transition from the previous state s_{t-1} to the current state s_t. The process update is assumed to be affected by zero mean Gaussian noise w_t with covariance Q_t. The measurement vector made at each time step is z_t and the matrix C is a transformation from state space into measurement space. Any noise in the measurement v_t is modeled as zero mean Gaussian with covariance R_t. The goal is to make an unbiased estimate of the state $\hat{s}_t$ at every time step to minimize the error covariance matrix P_t.

$$P_t = E[(s_t - \hat{s}_t)(s_t - \hat{s}_t)^T] \tag{3.13}$$

For a linear system with Gaussian process and measurement noise the Kalman filter finds the optimal gain K_t used to combine priori state estimate $\hat{s}_t^-$ made using equation (3.11) and the current measurement z_t for optimal state estimate $\hat{s}_t$ in a closed form. The state estimate is then calculated as

$$\hat{s} = \hat{s}_t^- + K_t(z_t - C\hat{s}_t^-) \tag{3.14}$$

where $z_t - C\hat{s}_t^-$ is the residual error between the actual and predicted measurement. Kalman filter equations can be separated into two steps, prediction and update. During the prediction step, the priori state and error covariance estimates are made. These estimates are then modified in the update step using the measurement. Figure 3.4 summarizes the two steps.

state prediction equations	measurement update equations
$\hat{s}_t^- = A\hat{s}_{t-1}$	$K_t = P_t^- C^T (CP_t^- C^T + R_t)^{-1}$
$P_t^- = AP_{t-1}A^T + Q_t$	$\hat{s}_t = \hat{s}_t^- + K_t(z_t - C\hat{s}_t^-)$
	$P_t = (I - K_t C)P_t^-$

Figure 3.4: Kalman filter state prediction and update.

3.2.1 Extended Kalman Filter

Many real world systems are non-linear and have the form

$$s_t = f(s_{t-1}) + w_{t-1} \tag{3.15}$$
$$z_t = h(s_t) + v_t. \tag{3.16}$$

thus the Kalman filter cannot be applied directly. The Extended Kalman Filter (EKF) linearizes the equations about the operating point, approximating the system as

$$s_t \approx \tilde{s}_t + A(s_t - \tilde{s}_t) + w_{t-1} \tag{3.17}$$
$$z_t \approx \tilde{z}_t + C(s_t - \tilde{s}_t) + v_t \tag{3.18}$$

where $A = \frac{\partial f}{\partial s_t}$ and $C = \frac{\partial h}{\partial s_t}$ are the Jacobians of the state update and measurement equations with respect to the state, $\tilde{s} = f(s_{t-1}, 0)$ and $\tilde{z} = h(s_t, 0)$, the noiseless state and measurement estimates. The Kalman gain is computed but is no longer optimal due loss of higher order terms during linearization. The equations are again separated into prediction and update steps and are summarized in figure 3.4.

state prediction	measurement update
$\hat{s}_t^- = f(s_{t-1}, w = 0)$	$P_y = C_t P_t^- C_t^T + V_t R_t V_t^T$
$P_t^- = A P_{t-1} A^T + W_t Q_{t-1} W_t^T$	$K_t = P_t^- C^T (P_y)^{-1}$
	$\hat{s}_t = \hat{s}_t^- + K_t(z_t - h(\hat{s}_t^-, 0))$
	$P_t = (I - K_t C_t) P_t^-$

Figure 3.5: Extended Kalman filter state prediction and update.

3.3 Robotic Modeling

A robot manipulator is a chain of rigid links connected by prismatic or revolute joints. Starting at the base (link 0) of the manipulator joint i connects link $i-1$ to link i. With a coordinate frame attached at each link the joint actuation defines a transformation between the frames of link $i-1$ and i, a revolute joint i performs a rotation of θ_i rads about z_{i-1} axis and a prismatic joint i causes displacement of d_i meters along the z axis. Each joint has a single degree of freedom (DOF) and more complex structures such as a spherical wrist can be expressed as a combination of single DOF joints and intermediate links. Figure 3.6 shows a simple elbow manipulator with three revolute joints.

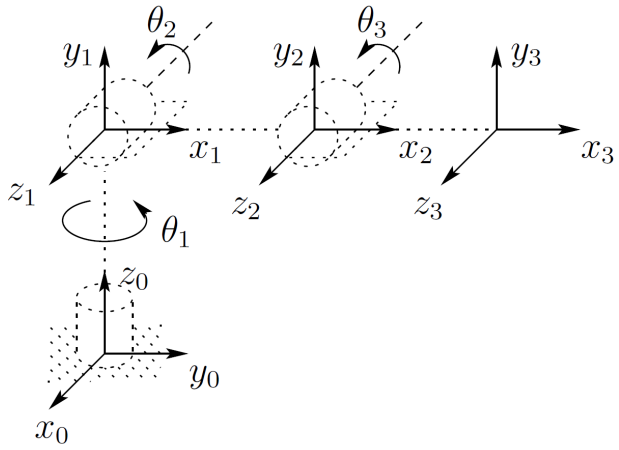


Figure 3.6: Simple elbow manipulator. The end effector frame (x_3, y_3, z_3) is defined by the actuation of three joints $\theta_1, \theta_2, \theta_3$. [65]

3.3.1 Forward Position

The manipulator must perform tasks in Cartesian space but is controlled through joint actuation, thus it is necessary to compute the position and orientation of the end effector using actuation values of the joints. A common approach to defining the geometry of a manipulator is the Denavit-Hartenberg convention which defines the transformation from one link to the next using four parameters: link length a_i,link twist α_i, link offset d_i, and joint angle θ_i. The four variables are used to define the transformation matrix from link $i-1$ to link i as

$$T_{i-1,i} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i)cos(\theta_i) & sin(\theta_i)sin(\alpha_i) & a_i cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i)cos(\alpha_i) & -cos(\theta_i)sin(\alpha_i) & a_i sin(\theta_i) \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

Then the transformation from the base of the robot to the end effector $T_{0,ee}$ for an n linked robot can be computed as

$$T_{0,ee} = T_{0,1} T_{1,2} \cdots T_{n-1,n} \tag{3.20}$$

$$R_{0,ee} = T_{0,ee}(1:3, 1:3) \tag{3.21}$$

$$x_e e = T_{0,ee}(1:3, 4). \tag{3.22}$$

Orientation and translation of the end effector are represented by the rotation matrix $R_{0,ee}$ and translation x_{ee} components of the transformation.

3.3.2 Forward Velocity and Acceleration

It is also possible to derive the angular and linear components of velocity and acceleration of the end effector in terms of the joint velocities $\dot{q}_i$ and accelerations $\ddot{q}_i$. Considering the angular velocity of the previous frame w_{i-1}^{i-1} and the joint velocity $\dot{q}_i$ caused by actuation of the revolute joint i, the angular velocity of the next frame can be computed as

$$w_i^i = R_{i-1,i}^T w_{i-1}^{i-1} + R_{i-1,i}^T \dot{q}_i \tag{3.23}$$

where $R_{i-1,i}$ is the rotation matrix from frame $i-1$ to frame i. Taking the derivative of (3.23) the angular acceleration is obtained:

$$\alpha_i^i = R_{i-1,i}^T \alpha_{i-1}^{i-1} + R_{i-1,i}^T \ddot{q}_i + w_i \times (R_{i-1,i}^T \dot{q}_i) \tag{3.24}$$

The linear velocity can be calculated using the cross product of angular velocity and the displacement vector r_i from the previous frame to the next for a revolute joint

$$\dot{x}_i = R_{i-1}^T \dot{x}_{i-1} + w_i^i \times r_i \tag{3.25}$$

and directly using joint velocity for prismatic joints.

$$\dot{x}_i = R_{i-1}^T \dot{x}_{i-1} + R_{i-1,i}^T \dot{d}_i \tag{3.26}$$

Taking the derivative and adding gravity in the current frame, the linear acceleration in the current frame for a revolute joint i is given:

$$\ddot{x}_i = R_{i-1}^T \ddot{x}_{i,i-1} + \alpha_i^i \times r_i + w_i^i \times w_i^i \times r_i + R_{0,i} g \tag{3.27}$$

and

$$\ddot{x}_i = R_{i-1}^T \ddot{x}_{i,i-1} + R_{i-1,i}^T \ddot{d}_i + R_{0,i} g \tag{3.28}$$

given a prismatic joint i. Where $R_{0,i}$ is the rotation from the world frame to the current frame and g is the gravity vector.

3.4 Adaptive Oscillators

Consider the Hopf oscillator with a periodic forcing term G governed by the following differential equations:

$$\dot{x} = (\mu - r^2)x - f y + \epsilon G(t) \tag{3.29}$$

$$\dot{x} = (\mu - r^2)y - f x \tag{3.30}$$

where $\mu > 0$ and radius $r = \sqrt{x^2 + y^2}$ define the amplitude of the oscillation and f the frequency and ϵ is the perturbation scaling factor. The system can be represented in polar coordinates as

$$\dot{r} = (\mu - r^2)r + \epsilon G(t)cos(\Omega) \tag{3.31}$$

$$\dot{\Omega} = f - \frac{\epsilon}{r} G(t)sin(\Omega) \tag{3.32}$$

Small force perturbations of $\epsilon > 0$ do not change the general behavior of the stable limit cycle and can only effect its phase, more specifically the perturbations perpendicular to the limit cycle will be dampened out and tangential perturbations can greatly effect the phase. The tangential component of the perturbation is $\frac{\epsilon}{r} G(t)sin(\Omega)$. Using the tangential component of the perturbation as frequency dynamics,

$$\dot{f} = -\epsilon G(t)sin(\Omega), \tag{3.33}$$

allows the oscillator to synchronize to the frequency of the forcing term. This approach can be used for any type of limit cycle and guarantees convergence to the forcing term frequency [59].

3.5 Classification

This section provides an overview of employed classification and dimensionality reduction techniques employed in this thesis. Consider a set of n dimensional feature vectors X_i and their corresponding known labels Y_i such that $Y_i = f(X_i)$ for some unknown function f. The goal of classification algorithms is to estimate f given a training set of known feature vectors and label pairs. Often it is possible to increase performance of classification by aggregating the predicted labels of multiple classifiers. Chapter 6 proposes an approach to segment human motion into exercise repetitions by applying these methods to time-series joint angle data.

3.5.1 Dimensionality Reduction

Often not all n dimensions provide information to better estimate the label, dimensions with high noise and low correlation to the label may be detrimental to classification algorithms [23]. Using the n dimensional data-set may also be too computationally expensive due to its size. The purpose of dimensionality reduction is to project the n dimensional feature vector into a lower dimensional space where the vectors corresponding to different labels are more easily separable.

Principal Component Analysis (PCA)

PCA transforms the input data into a set of orthogonal vectors, termed *principal components* (PC), such that the first component represents the vector along which the input data experiences largest variability, and subsequent PCs containing smaller amounts of variance [70].

It is assumed that the first few vectors in the PCs contain the dimensions of significance, and the remaining PCs result from noise and error. However, the optimal number of PCs, m, may change based on the features utilized. To determine the optimal m the elbow approach is commonly taken. The elbow is determined by ordering the fraction of total variance in the data represented by each PC, then selecting top m components that contains some amount of variance.

Fisher's discriminant analysis (FDA)

PCA does not account for class labels in its operation, and only accounts for variance. This means that PCA could project separable data in a way that the data becomes inseparable. FDA incorporates the class labels and maximizes the separation between the classes in the projection [30]. Similar to PCA, some testing and tuning is required to determine a suitable m value. It can be shown that there will be at most $c-1$ positive eigenvalues where c is the number of class labels. Because of this, FDA is not well suited for two class problems since it will reduce the data to a one dimensional space and too much information is lost.

Kernel supervised principle component analysis (ksPCA)

PCA does not consider the class labels when projecting into lower dimensional space and FDA projects onto a limited number of dimensions. ksPCA overcomes both of these difficulties; it maximizes the dependence between projected feature vectors and their corresponding class labels by maximizing the Hilbert-Schmidt independence criterion which is a measure of the dependence between two random variables. Furthermore, a kernel can be applied to both the feature vector and the class label which allows the algorithm to project non linearly separable data into a lower dimensions in a way which makes it separable [5]. Because this is a supervised method, it is possible to project motion data into an underlying dimension in which segment points are in distinct clusters, allowing this algorithm to generalize to new motions not present in the training set.

3.5.2 Classifiers

Once the dimension reduction method is applied, a classifier is selected. The classifier is trained on the training data $X_{training}$, and applied to the testing data $X_{testing}$.

k-Nearest Neighbour (*k*-NN)

k-NN is a simple algorithm that classifies a given data point based on the labels of its closest *k* neighbours [30]. Various different distance metrics can be used. For simplicity, the Euclidean distance is employed, regardless of any dimensionality reduction that may have occurred. The *k* choice is significant, and should be an chosen to be an odd value to prevent voting ties between classes.

Quadratic Discriminant Analysis (QDA)

QDA, and the linear discriminant analysis (LDA), assumes that the $X_{training}$ can be represented by a Gaussian, and is obtained by equating the Gaussian representation of the two classes to each other, to obtain the class boundary [30].

Radial Basis Function (RBF)

RBF refers to a function whose output is dependent on the distance between the input datapoint and the origin. A RBF classifier, or a RBF network, is a mono-layer neural network that employs RBFs as the activation function.

Classification is done by examining the output of the RBF network to see which of the two-class outputs is higher. The higher value suggests that there is a higher probability that the observation belongs to that class.

Support Vector Machine (SVM)

SVM is a binary linear classifier. The SVM model generates a decision boundary by maximizing the distance to the closest samples from the two classes, and classifies new data points based on their position relative to this decision boundary. If the data is not linearly separable soft-margin SVM or kernel SVM can be used [17]. In soft-margin SVM mis-classifications are permitted within some distance (margin) of the decision hyper-plane. Kernel SVM applies a kernel transformation to the data in hopes it is linearly separable in a higher dimensional space.

Artificial neural networks (ANN)

ANN describe a type of classifier that models layers of interconnected neurons. These neurons are activated by some given input, weighted, and propagated to subsequent neurons [31].

The optimal typography is application specific. Tuning is required to determine the appropriate number of hidden layers, neuron count, and the number of training epoch, to prevent over-fitting. Classification ANN is designed to have as many outputs as classes, for a previously unseen feature vector the outputs can be interpreted as probabilities of the feature vector belonging to each class.

3.5.3 Aggregation

Several classifiers can be combined under an aggregation scheme to improve the performance of the classifier.

Boosting

AdaBoost is a systematic method of taking misclassified training data and emphasizing them in a subsequent training step [24]. However, the AdaBoost algorithm is formulated to work with classifiers that can accept weighted data points, which does not apply to all classifiers. Instead, a resampling scheme, based on the data weights, is employed [63]. An observation point is labeled based on a weighted majority vote from the ensemble classifiers.

Bagging

Bagging attempts to reduce overfitting by training several classifiers with only a subset of the full training data. An observation point is labeled based on the average of the results reported by the ensemble classifiers [13].

Chapter 4

Lower Body Modeling and Pose Estimation

This chapter proposes two different approaches to model the human lower body using branched kinematics. The first modeling approach represents the movement of the base in the world frame using 3 prismatic and 3 revolute joints, while the second switches the base to alternating ankles as the demonstrator walks. The two different models are visualized in figure 4.1. The extended Kalman filter is set up to estimate the joint angles from wearable inertial measurement units placed on the waist, knees, and ankles. The algorithm is validated on gait data collected from healthy participants. [1].

4.1 Prismatic-Revolute Base Model

A transformation from the world frame into the kinematic model's base frame can be expressed as a translation and rotation in each of the world axes *(x,y,z)*. Similarly, both linear and angular velocity and acceleration of the base frame can be represented as velocity and acceleration in the world axis. Thus, it is possible to model a moving base kinematic chain as a fixed base kinematic chain starting with 3 prismatic and 3 revolute joints in the world frame. To fully encompass lower body motion we consider the pelvis as the base, the hips and knees are modeled using three and one revolute joints respectively. This model allows each leg to act independently and thus errors in joint angle estimation in one leg should not affect the other. However it is not bound

[1] This work has been published in IEEE Engineering in Medicine and Biology Conference 2014 [34]

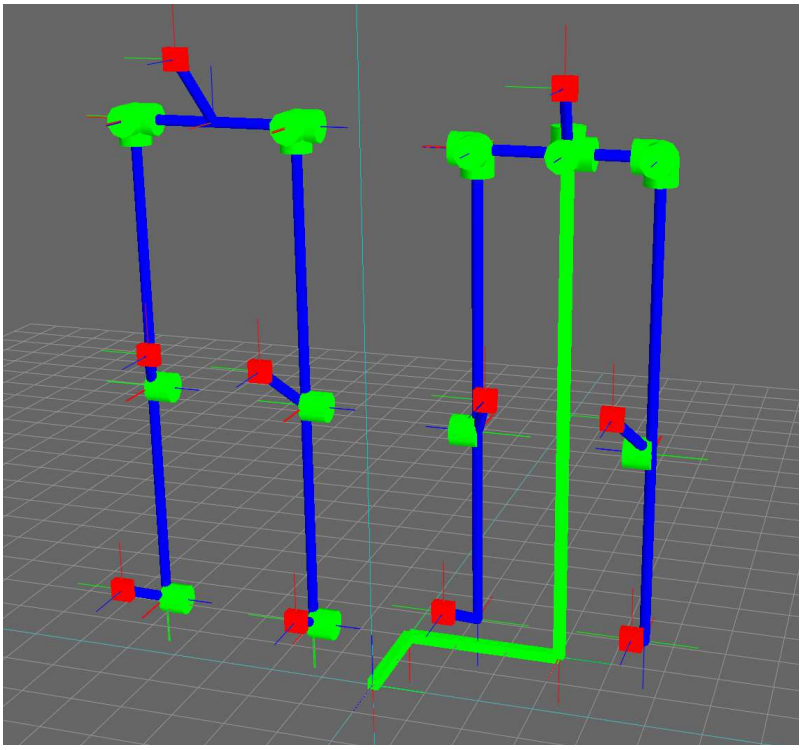


Figure 4.1: Two proposed kinematic models for the human lower body. Blue line segments, red boxes, green cylinders, and green line segments, represent links, sensors, revolute joints, and prismatic joints, respectively. Switching base model (left) with single DOF at the ankles and knees and three DOF at the hips. Prismatic-Revolute model (right), the transformation from world frame and the base frame is described using 3 prismatic and 3 revolute joints, the hips are modeled with 3 DOF and the knees with 1 DOF.

to the observable region, accumulating position error in the first three prismatic joints will cause the position of the base to diverge.

4.2 Switching Base Model

A different approach to overcome the limitation of a stationary base is inspired by the characteristic of gait where one leg is the support leg and the other leg is the swing leg. The center of the ankle of the support leg can be considered as the base and the base switches to the other leg when ground contact of the recent swing leg is detected. Adding a single revolute joint at the ankle to the knee joint and the three hip joints allows us to capture human forward gait motion. This model has no prismatic joints and thus errors in acceleration and velocity do not accumulate as transnational drift. It also has fewer degrees of freedom resulting in less uncertainty during pose estimation.

4.3 Extended Kalman Filter for Kinematic Chain

To fully capture participants motion during gait exercises the position q, velocity $\dot{q}$, and acceleration $\ddot{q}$ of the joint angles must be estimated. Thus the EKF state vector is defined as $[q, \dot{q}, \ddot{q}]$ and the measurement vector includes the IMU's accelerometer and gyroscope sensor readings. Position, velocity, and acceleration of the joint angles can be described as a smooth function, where their values at the next time-step are predicted by integrating the velocity and acceleration terms [44]. Any change in the acceleration is assumed to be part of the noise:

$$q_t = q_{t-1} + \dot{q}\Delta t + \ddot{q}\Delta t^2/2 \tag{4.1}$$
$$\dot{q}_t = \dot{q}_{t-1} + \ddot{q}\Delta t \tag{4.2}$$
$$\ddot{q}_t = \ddot{q}_{t-1}, \tag{4.3}$$

where Δt is the time difference between each measurement. Since the state change equations are already linear the A matrix is formulated as follows.

$$A = \begin{matrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{matrix} \tag{4.4}$$

By considering the sensors as end effectors of the kinematic chain, the measurement prediction is done using forward kinematics 3.3 and the derivative of the measurement equation with respect to the state becomes a combination of standard Jacobians used in robotics [15]. The Jacobians for gyroscope and accelerometer measures are:

4.3.1 Gyroscope

A gyroscope sensor measures the rate of rotation around its internal frame. Rate of rotation for an end effector in a kinematic chain can be expressed as $w = J_w\dot{q}$, where J_w is the angular Jacobian in the end effector frame. Thus the observation Jacobian in the EKF is

$$J_{gyro} = \begin{bmatrix} 0 & \mathbf{J_w} & 0 \end{bmatrix}. \tag{4.5}$$

4.3.2 Accelerometer

An accelerometer measures the acceleration in the sensor's frame including gravity. Acceleration applied to the sensor from the kinematic chain is a combination of centripetal acceleration due to the velocity of revolute joints $\dot{q}$ and tangential acceleration due to the joint accelerations $\ddot{q}$. We can express the combined acceleration by using the velocity Jacobian, its derivative, and the gravity vector in the sensor frame:

$$\mathbf{a} = \mathbf{R_0^{sens}}\mathbf{g} + \dot{\mathbf{J}}_\mathbf{v}\dot{\mathbf{q}} + \mathbf{J_v}\ddot{\mathbf{q}}, \tag{4.6}$$

where $\mathbf{R_0^{sens}}$ is the rotation matrix from the world frame into sensor frame and $\mathbf{g} = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}$ is the gravity vector. Taking the derivative of (4.6) with respect to the state variables results in an observation Jacobian for the accelerometer

$$\mathbf{J_{accel}} = \begin{bmatrix} \frac{\mathbf{dR_0^{sens}}}{\mathbf{dq}}\mathbf{g} & \dot{\mathbf{J}}_\mathbf{v} & \mathbf{J_v} \end{bmatrix}, \tag{4.7}$$

where $\frac{\mathbf{dR_0^{sens}}}{\mathbf{dq}}$ can be expressed with respect to each joint $\mathbf{q_i}$ through skew symmetric matrices and the rotation matrix to the joint frame $\mathbf{R_0^i}$:

$$\frac{\mathbf{dR_0^{sens}}}{\mathbf{dq_i}} = \mathbf{R_0^i S(k) R_i^{sens}}. \tag{4.8}$$

For the Prismatic-Revolute model, EKF can be applied directly by adding the position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$ of the three prismatic and three revolute joints describing the motion of the base to the state vector. For the Switching Base model the state vector remains as the position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$ of the joints in the kinematic model. However when the base is switched (i.e. when a heel strike is detected) the kinematic chain reverses, because of this the covariance estimate $\mathbf{P}$ of the EKF becomes invalid. To overcome this problem, we keep track of two covariance matrices: $\mathbf{P}_{\mathbf{left}}$ for when the left ankle is the base of the kinematic chain and $\mathbf{P}_{\mathbf{right}}$ for the right. When a step is detected, the error covariance $\mathbf{P}$ is switched to the appropriate matrix.

This approach requires an additional heel strike detection algorithm to trigger the base switching. A large spike is observable in the accelerometer sensor above the ankle that measures vertical accelerations. Using these spikes, steps are detected as proposed in [75]. The signal is filtered using a fourth order Butterworth band-pass filter with cutoff frequencies of 3Hz and 40Hz. Next, it is differentiated to emphasize the location of the heel strikes and squared to remove directional information. A moving window integrator finally smooths the signal and regions above a threshold are indicated as detected heel strikes [75]. Figure 4.2 shows the detected heel strikes, the kinematic model base is switched at the starting point of the strike region.

4.4 Experimental Results

To test the accuracy of the proposed algorithm, the estimated pose was compared to motion capture data.

4.4.1 Data Collection

Gait data of 5 healthy participants, ages 19 to 25, was collected. The SHIMMER IMUs were used to collect gyroscope and accelerator measurements. These sensors sample the on-board Freescale MMA7361L 3D accelerometer and two InvenSense 2D IDG-500 gyroscopes to obtain acceleration and rate of rotation about the 3 internal axes [18]. Calibration software from Shimmer Research based on Parvis and Ferraris [55] was used to calibrate the sensors. Motion capture markers were placed on the ankles, knees, and hips as well as on each of the IMUs. Due to the limited size of the motion capture studio it was not possible to capture continuous gait, thus each participant completed two

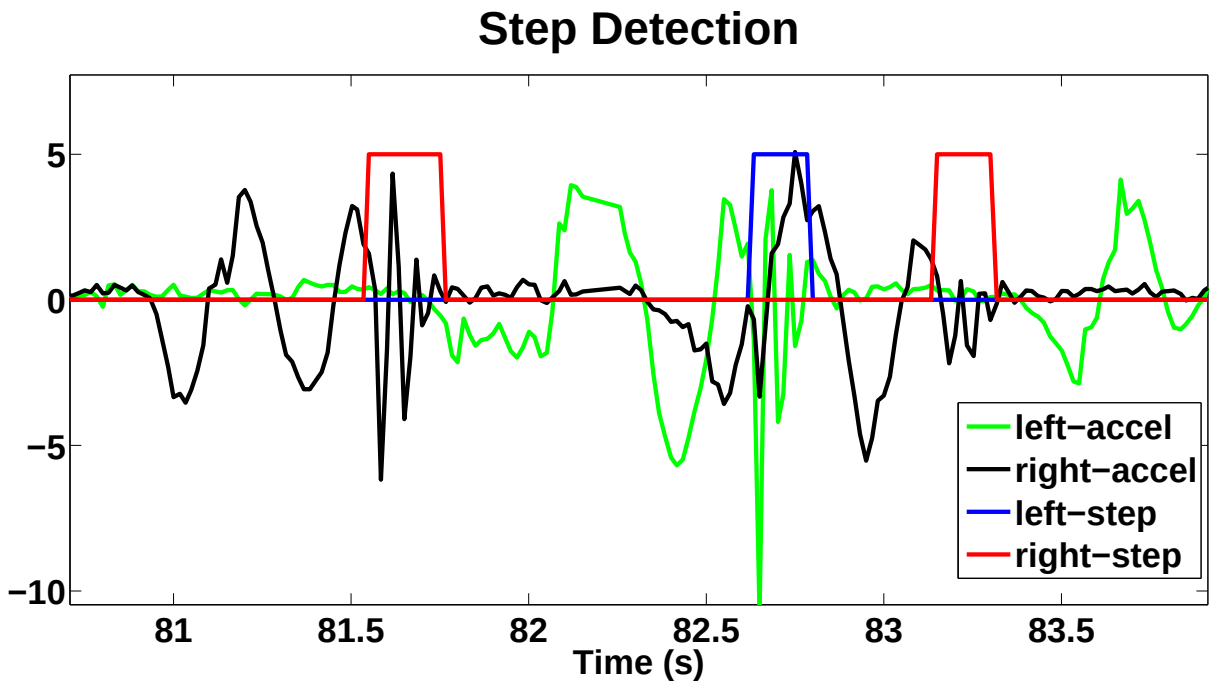


Figure 4.2: Step detection algorithm applied to gait of a healthy participant. The red and blue lines represent regions of detected heel strike and the green and black are right and left normalized acceleration data respectively.

full gait cycles 10 times. The experiment was approved by the University of Waterloo Research Ethics Board, and signed consent was obtained from all participants.

4.4.2 Preprocessing

To successfully run the EKF algorithm we need an accurate lower body model of the participant. Using the average distance between motion capture markers during the exercise, we calculate the hip to hip, hip to knee, and knee to ankle lengths as well as the IMU offsets for the kinematic model. Due to human physiology, when each IMU is attached using Velcro straps it experiences slight rotation away from the link. We find the 100 sample window with the least accelerometer variance while the participant is standing still and calculate the rotation matrix between the IMU's gravity vector and world gravity vector g. This rotation matrix is used to align the IMU's frame with the link frame in the kinematic model. Since gyroscopes are known to have drift present in their measurement we also use the window of least variance to compute the gyroscope offset for each of the sensors. It is also important to appropriately select the covariance for both the process **Q** and measurement noise **R** for best EKF performance. While the sensor noise can be measured, there is no way to measure the process noise and those parameters are selected by experience. In our experiments, the process noise at each joint was set as $0.01, 0.1, 1$ for position, velocity, and acceleration respectively. The measurement noise was set to $5 rad/s, 250 m/s^2$ for each axis of the gyroscope and accelerometer respectively.

4.4.3 Performance

To compare the joint angles estimated by the proposed approach with the mocap data, the knee joint angles are computed from motion capture data by looking at the cross product of the vectors formed by the offsets between hip to knee $\vec{a}$, and knee to ankle $\vec{b}$ markers.

$$\theta = sin^{-1}(\frac{\vec{a} \times \vec{b}}{||\vec{a}||||\vec{b}||}) \tag{4.9}$$

Figure 4.3 shows the Switching Base model performance at estimating the knee joint angles for a short walk. The RMSE error in knee angle for both approaches is shown in Table I. The root mean squared error for the left and right knee joint angles are 5 and

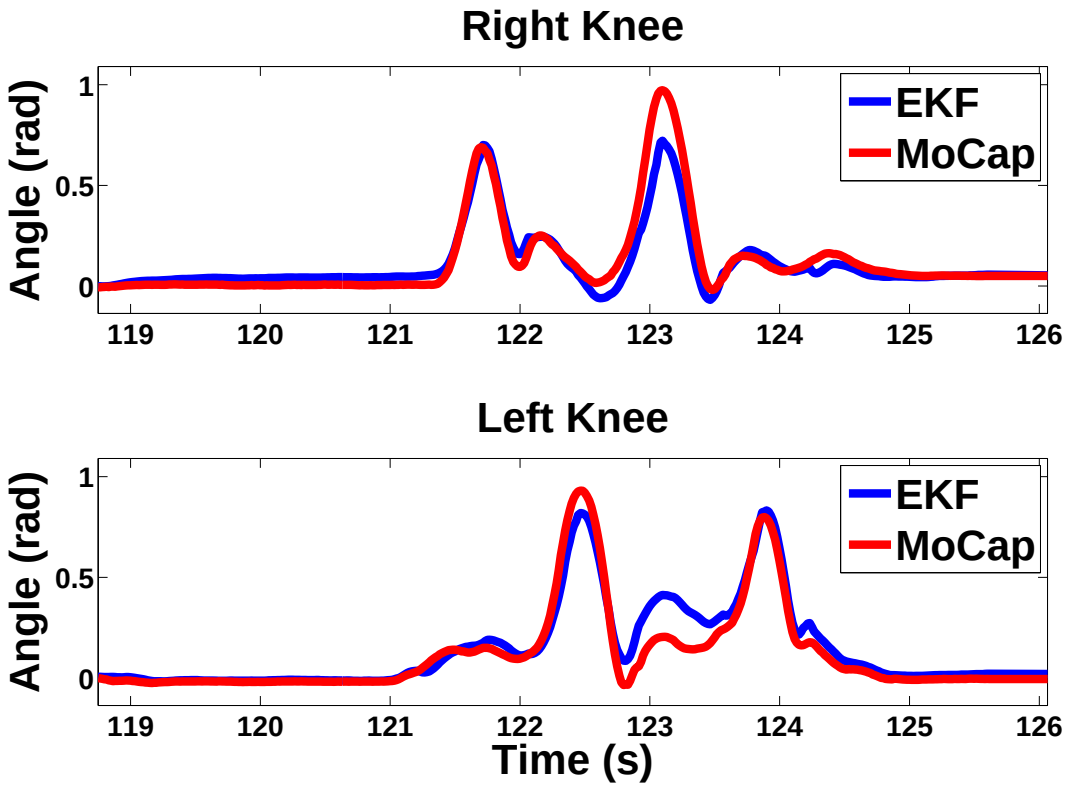


Figure 4.3: Comparison between the knee joint angles over two full gait cycles estimated from the motion capture studio (red) and by the Switching Base model EKF (blue).

6.2 degrees respectively which is an improvement over accuracy of goniometers (7 deg) commonly used by therapists to measure joint range of a single joint at a time. We also compared the Cartesian positions of the ankles and knees. Gait features such as stride length and foot placement can be calculated per step. Thus error was considered as zero at the start of each step cycle. Due to the difference in modeling of the legs (independent in the Prismatic-Revolute model and dependent in the Switching Base model), it was important to consider the support and swing leg performance separately. Because the ankle is bound in the Switching Base model, noise in the IMU measurements cannot greatly affect the support leg. However due to the dependence of the legs the swing leg's position error is increased. The joint angle and position estimation root mean squared errors are summarized in table 4.1. As expected, we see that the Switching Base model estimates the position much better for the support leg and the Prismatic-Revolute model slightly outperforms in estimating the position of the swing leg.

4.4.4 Sources of Error

A major source of error in our experiment can be attributed to the sensors and their attachment to the participant. The IMUs are secured using elastic straps, during a heel strike the strap allows the knee and ankle sensors to continue rotating forward. The EKF estimates this as rotation at the hip joint and causes a characteristic under-step. Other sources of error include modeling the joints at marker locations instead of anatomically correct locations and excluding the ankle to heel distance for the Switching Base model.

The EKF assumes a constant acceleration state evolution model and any changes in acceleration are seen by the EKF as noise. This leads to lag in the acceleration estimate which, through integration, affects the velocity and position estimates. In the next chapter a rhythmic version of EKF is developed that can over time learn the acceleration model and remove this assumption leading to improvement in joint position, velocity, and acceleration estimation.

Table 4.1: Root mean squared error of knee joint angle (Ang) and Cartesian position estimation using the two proposed kinematic models. The Cartesian results are shown for knee (K) and ankle (A) of each leg (left (L) and right (R)) for the support (Su) and swing (Sw) phase

	Switching Base Model			Prismatic-Revolute Model		
L K Ang	5.03±1.27 (deg)			5.64±2.12 (deg)		
R K Ang	6.20±1.48 (deg)			6.46±2.37 (deg)		
	x (cm)	y (cm)	z (cm)	x (cm)	y (cm)	z (cm)
L A Su	1.3±1.0	0.3±0.1	0.6±0.3	10.5±1.7	5.6±1.4	3.1±1.4
L A Sw	9.4±2.5	11.9±2.3	9.5±1.3	9.1±3.7	8.5±2.2	8.9±1.9
R A Su	1.0±0.5	0.3±0.1	0.9±0.4	10.6±2.5	5.4±1.6	4.1±0.5
R A Sw	9.9±3.8	8.2±2.2	5.3±1.2	8.1±3.4	5.3±1.6	6.5±1.6
L K Su	3.4±1.6	1.3±0.4	0.9±0.3	5.5±0.7	3.6±0.7	2.4±1.0
L K Sw	6.7±1.7	10.3±2.6	5.2±0.6	7.2±2.7	6.4±2.8	4.0±1.1
R K Su	3.3±1.1	4.1±1.4	1.4±0.6	5.6±0.7	3.4±1.0	2.6±0.9
R K Sw	6.5±2.6	4.4±1.2	2.3±0.6	5.3±1.9	3.3±1.2	2.0±0.9

Chapter 5

Rhythmic Extended Kalman Filter (Rhythmic-EKF)

This chapter expands the EKF pose estimation algorithm to improve performance for periodic motion such as walking. The proposed approach combines the extended Kalman filter with a canonical dynamical system. The system incrementally learns the rhythmic motion over time, improving the estimate over a regular EKF, and segmenting the motion into repetitions. An estimate of the lower body pose is calculated in real time using data from several inertial measurement unit sensors and the EKF as described in chapter 4. A canonical dynamical system (CDS) is adapted to the underlying motion, improving the performance of the EKF over time by removing the constant acceleration assumption and assuming periodic motion. A virtual yaw sensor is used to eliminate gyroscope drift. The phase and frequency variables estimated by the CDS also allow extraction of useful features from the estimated motion, which can be used to evaluate gait properties and isolate gait segments. Figure 5.1 provides the schematic of the algorithm [1].

5.1 Rhythmic Extended Kalman Filter for Kinematic Chain

In section 4.3 the state vector of the EKF consisted of the position q, velocity $\dot{q}$, and acceleration $\ddot{q}$ of the joint angles while the measurement vector included the IMU sensor

[1] This work has been accepted for publication to IEEE RAS Humanoids Conference 2015 [33]

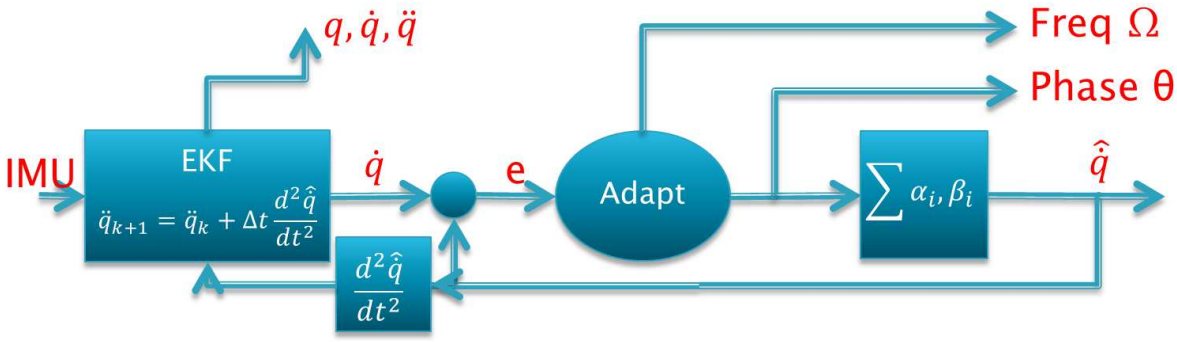


Figure 5.1: Rhythmic-EKF Schematic

readings. The standard approach of assuming any change in the acceleration is part of the noise was used:

$$q_t = q_{t-1} + \dot{q}\Delta t + \ddot{q}\Delta t^2/2 \tag{5.1}$$
$$\dot{q}_t = \dot{q}_{t-1} + \ddot{q}\Delta t \tag{5.2}$$
$$\ddot{q}_t = \ddot{q}_{t-1}, \tag{5.3}$$

where Δt is the time difference between measurements. However, this leads to lag in the acceleration estimate which propagates to the velocity and position estimates. Instead of a constant acceleration we can add higher order terms such as jerk into the model and assume constant jerk.

$$\ddot{q}_t = \ddot{q}_{t-1} + \dddot{q}_{t-1}\Delta t \tag{5.4}$$
$$\dddot{q}_t = \dddot{q}_{t-1} \tag{5.5}$$

Unfortunately IMU sensors cannot measure jerk and thus no extra information is available to improve the acceleration estimate.

Consider a special case of motion estimation where the motion is periodic. This is the case for human gait and any repetitive motions. Then instead of constant jerk we can assume it is a function of the phase Ω and frequency f of the motion.

$$\dddot{q}_t = F(\Omega) \tag{5.6}$$
$$\dot{\Omega} = f \tag{5.7}$$

We propose to learn $F(\Omega)$ during online estimation, to gradually improve the performance of the EKF over time. We use the canonical dynamical system (CDS) to learn the

underlying harmonic Fourier series of the joint velocity and use the second analytical derivative as the jerk $F(\Omega)$. We use the term rhythmic to describe periodic movement following the terminology of rhythmic movement primitives [51].

Canonical Dynamical System

The canonical dynamical system is an adaptive oscillator estimator (reviewed in Chapter 3.4) based on the harmonic Fourier series limit cycle

$$\hat{y} = \sum_{i=1}^{n} \alpha_i cos(i\Omega) + \sum_{i=1}^{n} \beta_i sin(i\Omega) \tag{5.8}$$

where $\hat{y}$ is the state estimate, α_i and β_i are the Fourier coefficients, and Ω is the current phase. A feedback adaptive frequency phase oscillator is used to adapt the coefficients and phase to learn a rhythmic motion online.

$$\dot{\Omega} = f - \zeta e sin(\Omega) \tag{5.9}$$

$$\dot{f} = -\zeta e sin(\Omega) \tag{5.10}$$

$$\dot{\alpha_i} = \eta cos(i\Omega) e \tag{5.11}$$

$$\dot{\beta_i} = \eta sin(i\Omega) e \tag{5.12}$$

Where e is the error between actual and estimated values $e = y - \hat{y}$ and ζ and η are the frequency and coefficient learning rates respectively [56].

To learn the rhythmic Jerk $F(\Omega)$ using this method actual Jerk values are required which are not available from the EKF estimated state. Noting that the Fourier series are infinitely differentiable the CDS can be used to learn the lower order terms such as acceleration or velocity, and then analytically differentiated to compute $F(\Omega)$. Initializing the Fourier coefficients to zero ensures that the initial performance of the Rhythmic-EKF is identical to that of regular EKF. For pose estimation using IMU sensors we choose joint velocity as the term to learn since it is directly measured by gyroscopes, should not contain a bias, and is not assumed to be constant, as is acceleration before the CDS is converged.

5.2 Eliminating Gyroscope Drift During Gait

The accelerometer's constant gravity reference ensures that EKF will accurately correct gyroscope drift coinciding with the world x and y axis. However EKF cannot correct

gyroscope drift around the world z axis using the accelerometer data, this can result in accumulation of error in joint angle estimates for joints which rotate about axes parallel to the direction of gravity. To handle this issue, Lin and Kulic proposed applying potential fields in the EKF joint acceleration state once the joint position exceeds the joint limit [44]. The approach requires the potential field to be tuned for each joint independently and may cause oscillation of position estimate between the joint limits. Instead we use a virtual yaw sensor γ as part of the measurement vector. It requires only a single tuning parameter and effectively combats gyro drift.

We assume that, for the most part, the human is walking in a straight line and thus the yaw angles of their pelvis and each thigh should not drift away from the original orientation. With this in mind we attach a virtual yaw sensor to the pelvis and each thigh, the measurement is set to the starting orientation and is added as part of EKF measurement vector. The measurement noise of the virtual yaw sensor represents the standard deviation of expected motion and is used as a tuning parameter.

Using forward kinematics the yaw measurement prediction is computed as:

$$\gamma = tan^{-1}(\frac{R_{0,i}(2,1)}{R_{0,i}(1,1)} \tag{5.13}$$

and the Jacobian of the yaw sensor is the third row of the velocity Jacobian [15] in the base frame. Figure 5.2 shows the benefits of using a virtual yaw sensor in EKF when the gyroscope measurement is biased. Note that 3D motion is still observable and estimated, but the yaw sensor prevents the angle from drifting with time.

5.3 Experimental Results

This section compares the pose estimation performance of regular (section 4.3) and Rhythmic EKF (section 5.1). First convergence to a rhythmic motion is evaluated in simulation. Next we show that Rhythmic EKF outperforms pose estimation for healthy participants and is able to accurately segment the rhythmic motion into repetitions. Finally, to verify that the algorithms can be used in a clinical setting, the performance on two stroke patients undergoing rehabilitation is considered.

5.3.1 Simulation Results

To test the convergence properties of Rhythmic EKF we simulate a gyroscope and an accelerometer attached at an offset of 0.5m to a single revolute joint rotating about the

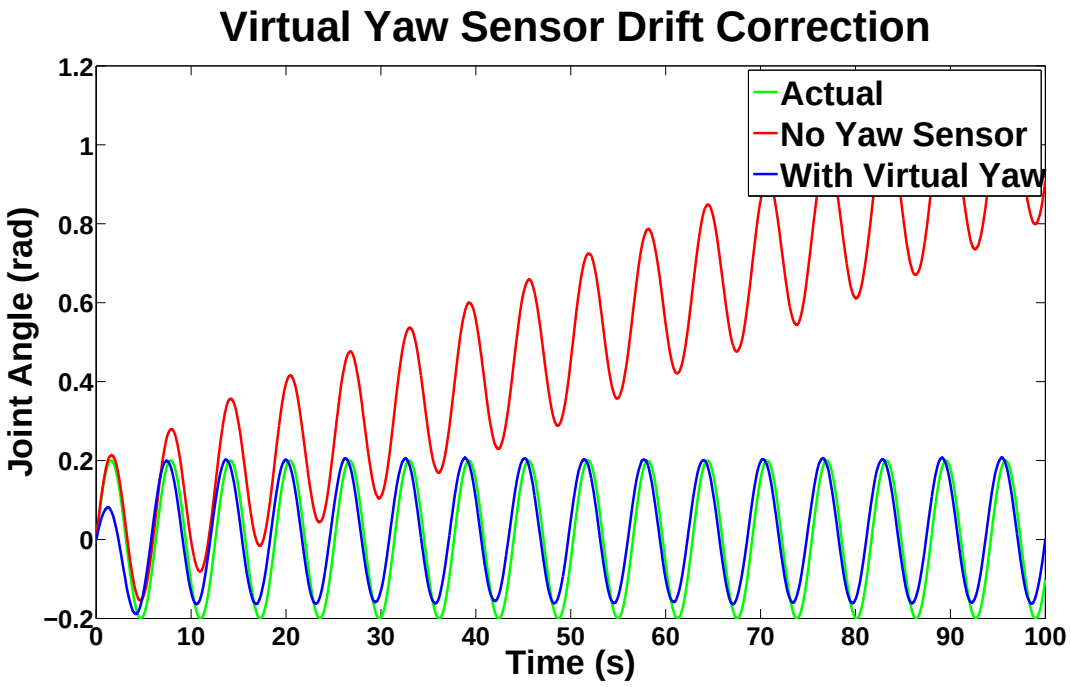


Figure 5.2: Simulated EKF tracking of single joint angle which actuates about the world z axis and is measured by a biased gyroscope. The gyroscope has Gaussian noise with standard deviation of 0.05 rad/s and a constant bias of 0.01 rad/s. Without the virtual yaw sensor, EKF integrates the bias into the joint position estimate and error quickly accumulates. Once a virtual yaw sensor is added to the measurement vector it does not allow the joint position estimate to diverge. The virtual yaw sensor measurement is set to a constant 0 and its noise standard deviation is tuned to 0.1 rad.

world x axis. To simulate human motion the joint is actuated for 100 seconds using a Fourier series with 10 harmonics starting at 4 rads/s and coefficients from a uni-variate distribution. Using forward kinematics, the sensor measurements are computed and random Gaussian noise of $2\frac{m}{s}^2$ and $0.5\frac{rads}{s}$ is added to the accelerometer and gyroscope respectively. Next the EKF's noise parameters are found using Matlab's constrained minimization toolbox, by finding the noise parameters that minimize the joint position estimation error. The Rhythmic-EKF uses the same noise parameters as the EKF, the number of harmonics and well as frequency and coefficient learning rates were chosen experimentally to be 7, 0.7, and 0.2 respectively. We verify that Rhythmic-EKF can successfully adapt the CDS to the rhythmic motion and compare the performance to EKF once convergence is achieved.

Figure 5.3 shows the adaptation process of Rhythmic-EKF to the rhythmic motion. Frequency is accurately tracked after 10 repetitions of the motion and the Fourier coefficients are learned within 20 repetitions. Once convergence is achieved the tracking is improved for all states, i.e., the acceleration, position, and velocity over EKF with optimal noise parameters. Particularly lag, overshoot and undershoot in acceleration and velocity estimates are reduced. Furthermore the predicted phase can easily be used for segmentation into repetitions and frequency can be used to track the speed of the motion. A closer view of one of the velocity peaks is shown in figure 5.4. Since acceleration is no longer assumed constant the predicted velocity does not overshoot or significantly lag the actual value.

EKF's acceleration estimate error increases linearly with Jerk since the model always predicts constant acceleration, Rhythmic-EKF builds an accurate model over time. Thus by plotting Jerk against error in acceleration estimate (figure 5.5) the benefits of Rhythmic-EKF are very clear.

5.3.2 Healthy Participant Results

Three subjects were asked to walk in place for two minutes while wearing 5 IMU sensors streaming at 50Hz and motion capture markers in a motion capture studio with eight Eagle cameras sampling at 100 frames per second. Two sets of data were collected: (1) participants walking at 70 steps per minute using a metronome for timing and (2) walking at their own pace. Three markers were placed on each of the sensors to determine their orientation and translation with respect to the link. Markers were also placed on the subject's left and right ASIS, back, as well as knees and ankles on both medial and lateral sides. These markers were used to estimate the joint centers and link

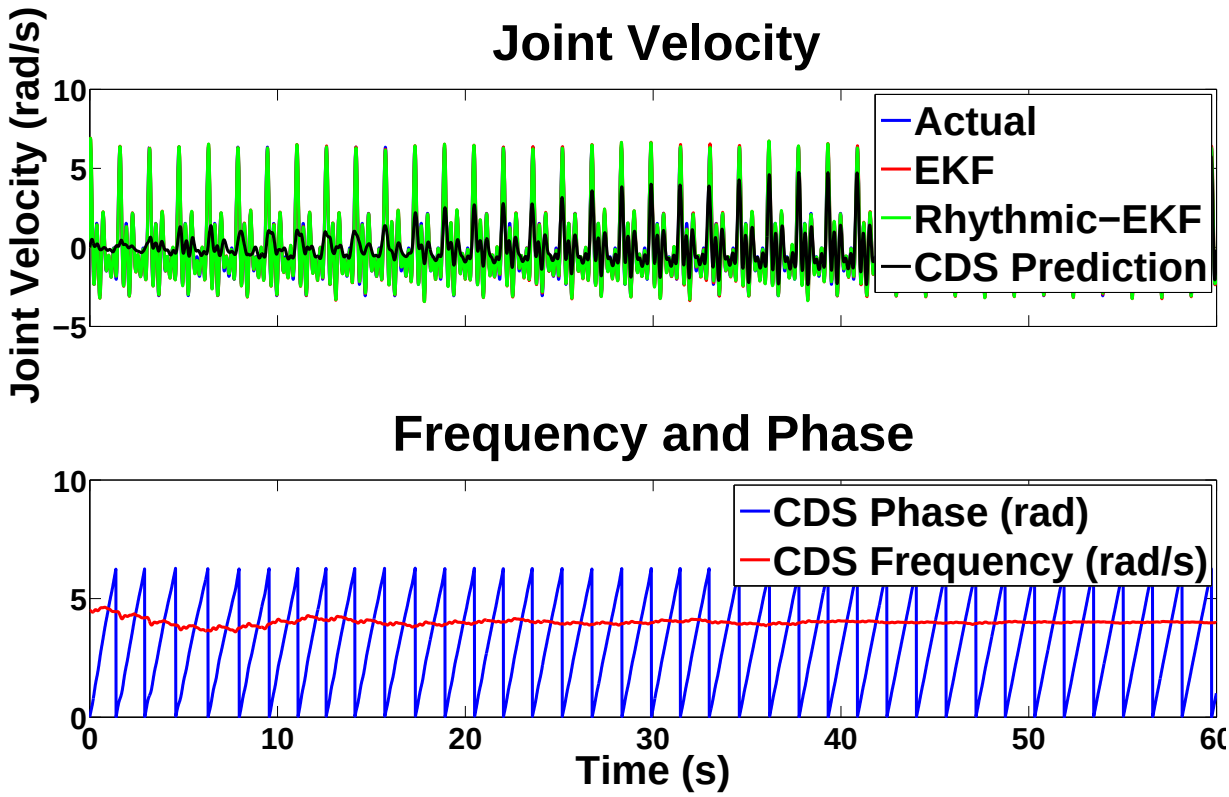


Figure 5.3: Rhythmic-EKF convergence to rhythmic motion. The frequency, phase, and Fourier coefficients quickly adapt to accurately predict the joint velocity.

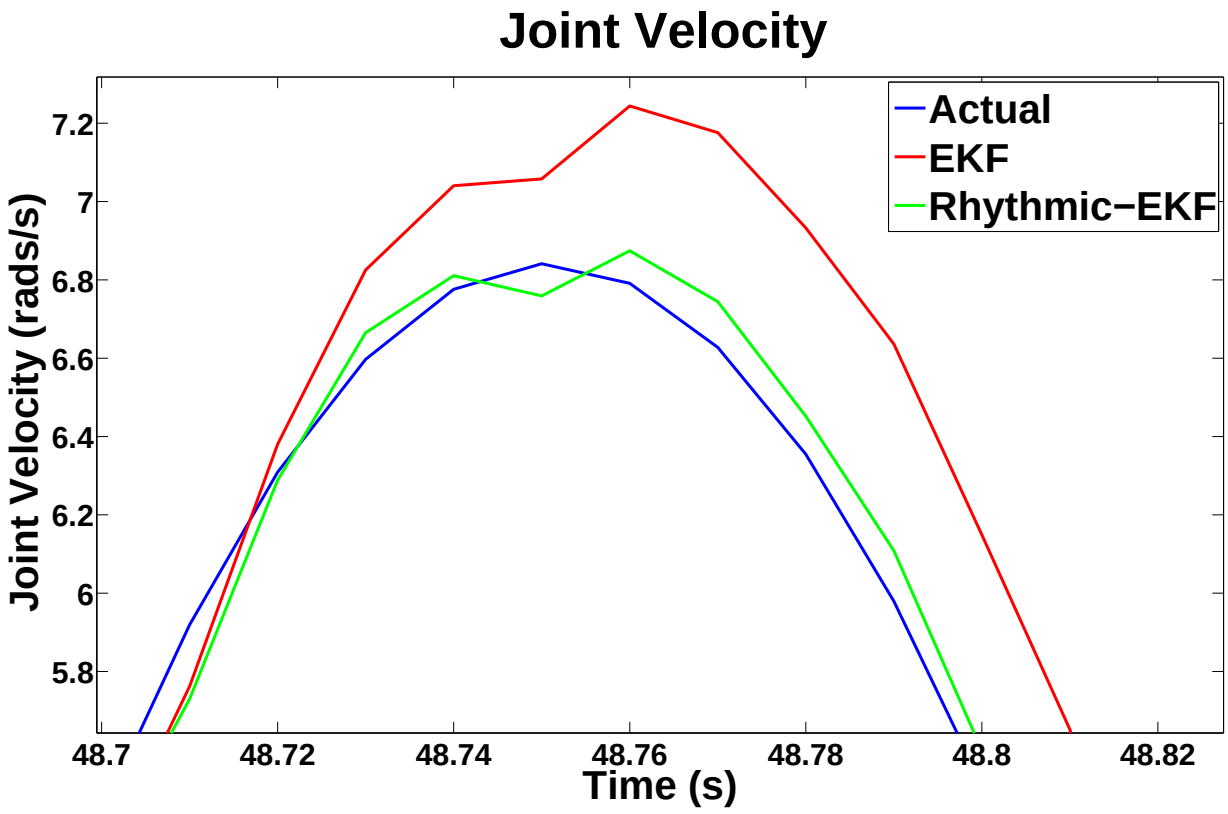


Figure 5.4: Rhythmic-EKF performance during velocity peak compared to that of EKF after convergence is achieved. Since Rhythmic-EKF has an accurate model to predict acceleration the velocity estimate no longer overshoots or lags the actual value significantly.

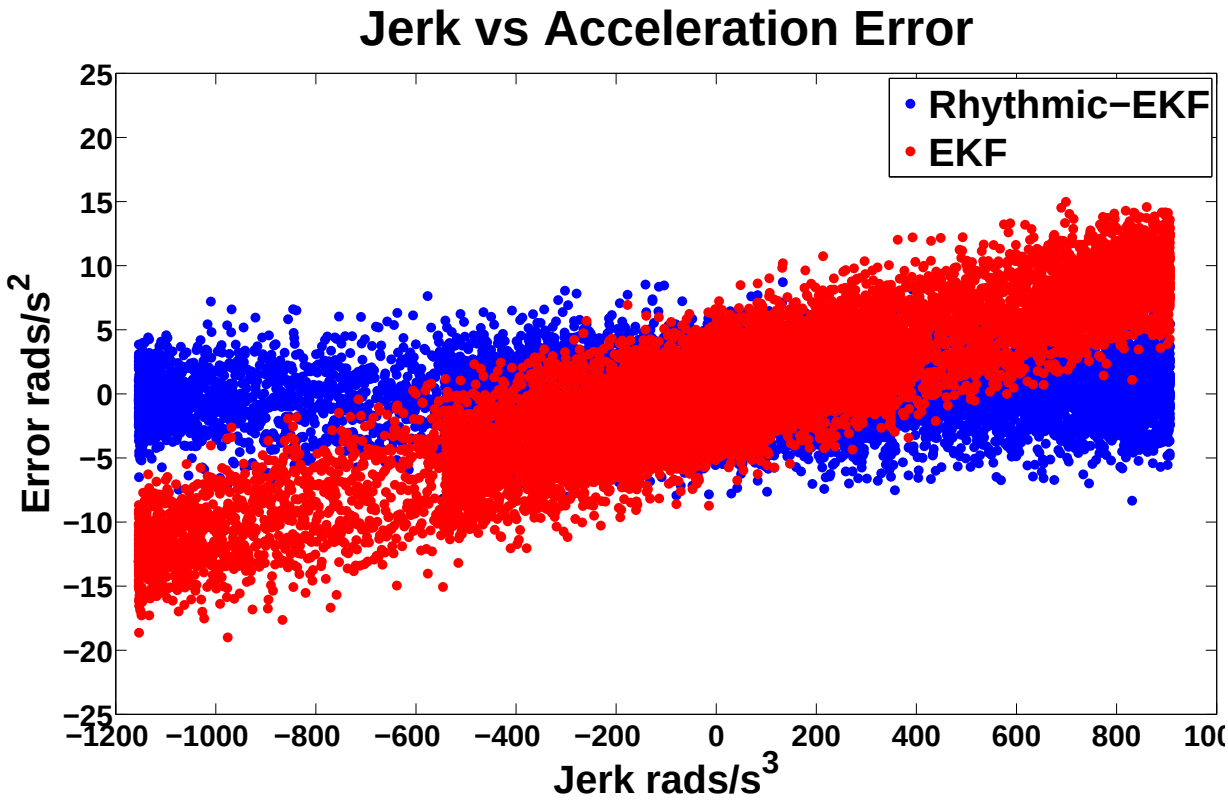


Figure 5.5: Due to the constant acceleration assumption, EKF's estimated acceleration will always lag and the error will linearly increase with Jerk. This leads to an increase in the velocity, and position errors due to integration. After convergence Rhythmic-EKF has an accurate model of the acceleration profile and thus removes the lag and linear dependence on Jerk. This leads to an error distribution around zero.

lengths to generate the kinematic model. The hip joint centers were determined using the pelvis width and length as well as leg length as described in Harrington et al. [26]. The position of the knee and ankle joint centers were obtained by taking the average of their respective medial and lateral marker positions. Data from the sensors and the motion capture was time-aligned in post processing. The study was approved by the University of Waterloo research ethics board.

Joint angles, velocities, and accelerations were obtained from marker data. The proposed Rhythmic-EKF and EKF algorithms were applied to the IMU data. Three virtual yaw sensors were added to the measurement vector at the center of the pelvis and each thigh to combat gyroscope drift. The noise parameters were tuned for best performance of the EKF algorithm for the first participant. Rhythmic-EKF was set to have 7 harmonics and the frequency and coefficient learning rates were set to 0.7 and 0.05 respectively. To phase synchronize all of the joints we use f and Ω estimated for the right knee joint for the entire lower body. Figure 5.6 shows Rhythmic-EKF convergence to human motion over one of the metronome walking trials. Table 5.1 shows the accuracy of the frequency convergence after the first 15 seconds of walking. At initialization the Fourier coefficients of the CDS are zero resulting in zero estimated jerk and Rhythmic-EKF behaving identically to regular EKF. Once the frequency is locked on the Fourier coefficients are incrementally learned and Rhythmic-EKF estimation outperforms that of regular EKF. Figure 5.7 shows the benefit of Rhythmic-EKF at different stages of CDS Fourier coefficient convergence.

Table 5.1: Mean frequency after the initial 15 seconds of walking estimated by the Rhythmic-EKF.

	Actual Freq (Hz)	Rhythmic-EKF Freq (Hz)	% Error
Subj1	1.167	1.170	-0.30
Subj2	1.167	1.179	-1.08
Subj3	1.167	1.169	-0.18

Rhythmic-EKF and EKF are expected to have the same performance during constant acceleration regions. Using data-sets of participants walking at their own pace, we verify that the proposed approach improves estimation over regular EKF in high acceleration regions. Table 5.2 presents the hip and knee joint position, velocity, and acceleration root mean squared error (RMSE) over regions where absolute acceleration exceeds 75% of maximum. Even without noise parameters tuned for Rhythmic-EKF, on average, it improves the velocity and acceleration estimation by 35% and 55% re-

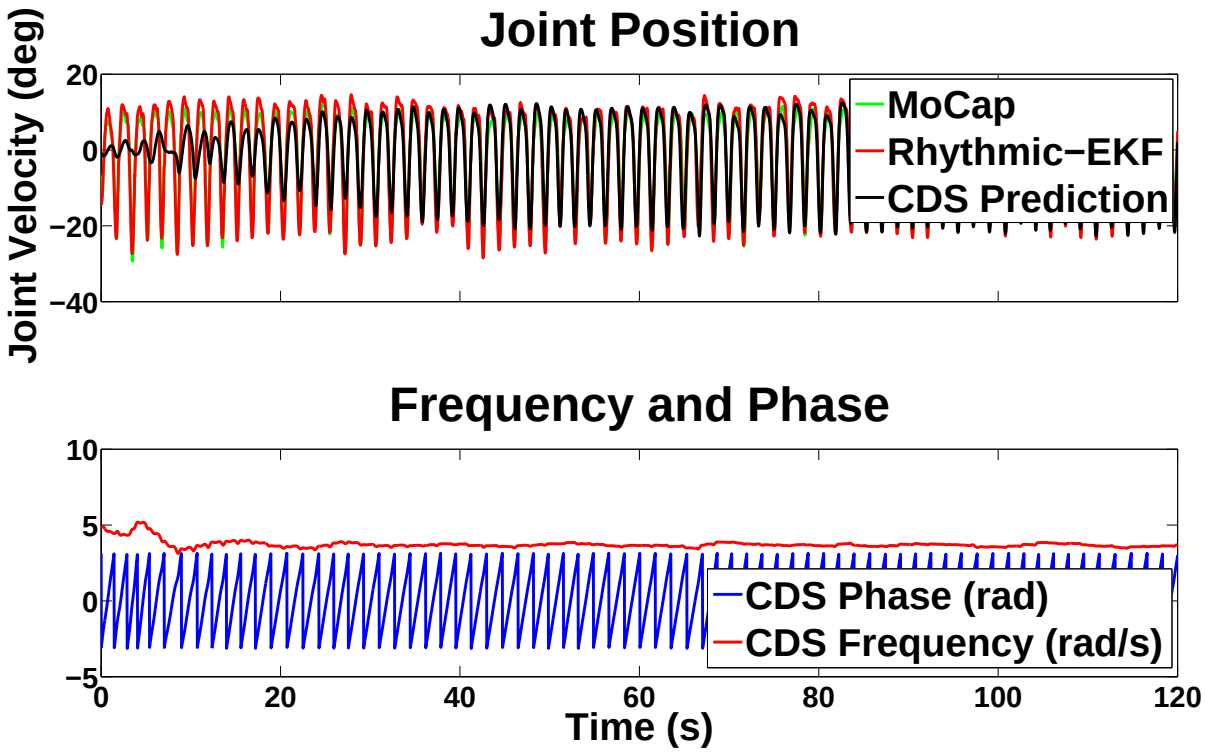


Figure 5.6: Rhythmic-EKF convergence to metronome paced human walking motion. Correct motion frequency is estimated after 5 repetitions. As soon as frequency is locked on the Fourier coefficients begin to converge and start improving the estimate of jerk.

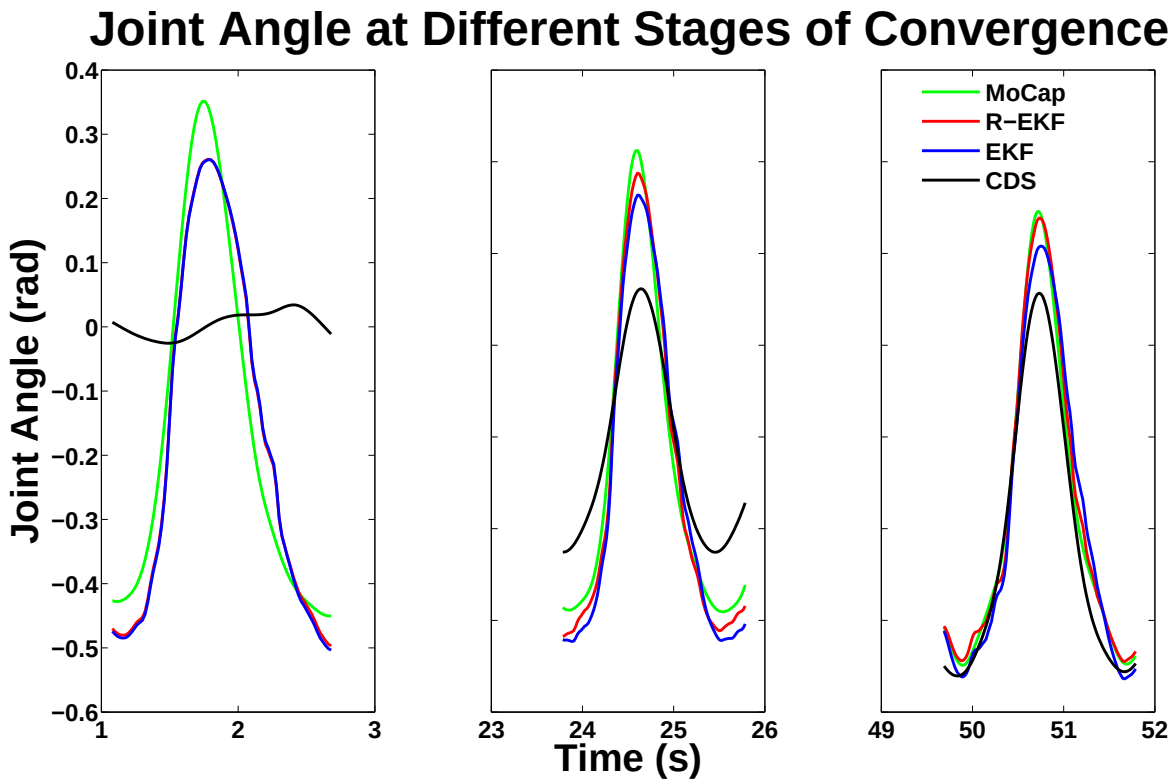


Figure 5.7: At initialization (left) the Fourier coefficients of the CDS are set to zero, thus the estimated jerk is zero and Rhythmic-EKF and EKF perform identically. Even while the coefficients are in the process of being learned (middle) Rhythmic-EKF already has a better model than constant acceleration and performance is improved. Once the coefficients converge to a good model of the motion (right), Rhythmic-EKF significantly outperforms EKF in high acceleration regions.

spectively in high acceleration regions. In low acceleration regions Rhythmic-EKF still outperforms EKF but the difference is not as drastic in acceleration estimation, Table 5.3 presents the RMSE for acceleration regions under 75% of maximum. In all the tables below, the results in bold indicate the best performing system for the corresponding row. Rhythmic-EKF significantly outperforms goniometry (RMSE 7° [20]) and visual observation (RMSE 9° [41]) techniques currently used by physiotherapists with an average RMSE of 2.4° in the hip and knee joint angle estimation.

Table 5.2: Root mean squared error of joint position, velocity, and acceleration for hip and knee joints (averaged over left and right) over top 25% acceleration regions.

	Rhythmic-EKF		**EKF**	
	Joint Position RMSE (deg)			
	Hip	Knee	Hip	Knee
Sub01	2.45	**2.46**	**1.95**	2.64
Sub02	**1.94**	**2.53**	2.14	3.09
Sub03	**3.65**	**2.00**	4.07	2.51
	Joint Velocity RMSE (deg/s)			
	Hip	Knee	Hip	Knee
Sub01	**14.83**	**15.46**	18.88	29.33
Sub02	**6.56**	**23.78**	12.03	35.03
Sub03	**12.78**	**14.90**	18.70	29.38
	Joint Acceleration RMSE (deg/s^2)			
	Hip	Knee	Hip	Knee
Sub01	**120.4**	**312.5**	277.1	587.2
Sub02	**80.4**	**339.3**	193.8	577.4
Sub03	**87.0**	**233.8**	190.5	421.8

The Rhythmic-EKF model also significantly improves measurement prediction which can be useful in control applications. Table 5.4 shows the RMSE between actual and predicted gyroscope and accelerometer measurements for the next time step, after the first 15 seconds of walking for the knee and ankle IMU.

Once Rhythmic-EKF achieves frequency convergence the estimated phase can be used for segmenting the motion as well as temporally aligning the segments. We allow the algorithm to run for 15 seconds before using the phase variable for segmentation. Between the 3 participants a total of 306 steps were taken, there were 11 visibly incorrect segments in the hip joints and 7 in the knee joints leading to an accuracy of 96.4%

Table 5.3: Root mean squared error of joint position, velocity, and acceleration for hip and knee joints (averaged over left and right) over bottom 75% acceleration regions.

	Rhythmic-EKF		**EKF**	
	Joint Position RMSE (deg)			
	Hip	Knee	Hip	Knee
Sub01	1.78	1.97	**1.56**	**1.81**
Sub02	**1.37**	**2.75**	1.69	2.81
Sub03	**3.56**	**1.89**	4.11	1.89
	Joint Velocity RMSE (deg/s)			
	Hip	Knee	Hip	Knee
Sub01	**11.91**	**13.37**	13.89	27.85
Sub02	**6.31**	**16.81**	9.35	30.25
Sub03	**11.21**	**12.77**	15.19	25.42
	Joint Acceleration RMSE (deg/s^2)			
	Hip	Knee	Hip	Knee
Sub01	**103.9**	**155.4**	122.1	270.0
Sub02	**56.7**	**162.5**	80.6	247.8
Sub03	**71.4**	**122.8**	89.1	185.9

Table 5.4: Rhythmic-EKF's and EKF's measurement prediction for the next time step compared to IMU data averaged for left and right sides.

	Rhythmic-EKF		**EKF**	
	Gyroscope RMSE (deg/s)			
	Knee IMU	Ankle IMU	Knee IMU	Ankle IMU
Sub01	**10.08**	**11.55**	17.52	21.23
Sub02	**10.69**	**13.05**	15.51	20.45
Sub03	**10.98**	**10.82**	18.78	23.08
	Accelerometer RMSE (m/s^2)			
	Knee IMU	Ankle IMU	Knee IMU	Ankle IMU
Sub01	**1.18**	**0.96**	1.35	1.18
Sub02	**1.18**	**1.05**	1.30	1.24
Sub03	**1.21**	**1.15**	1.34	1.48

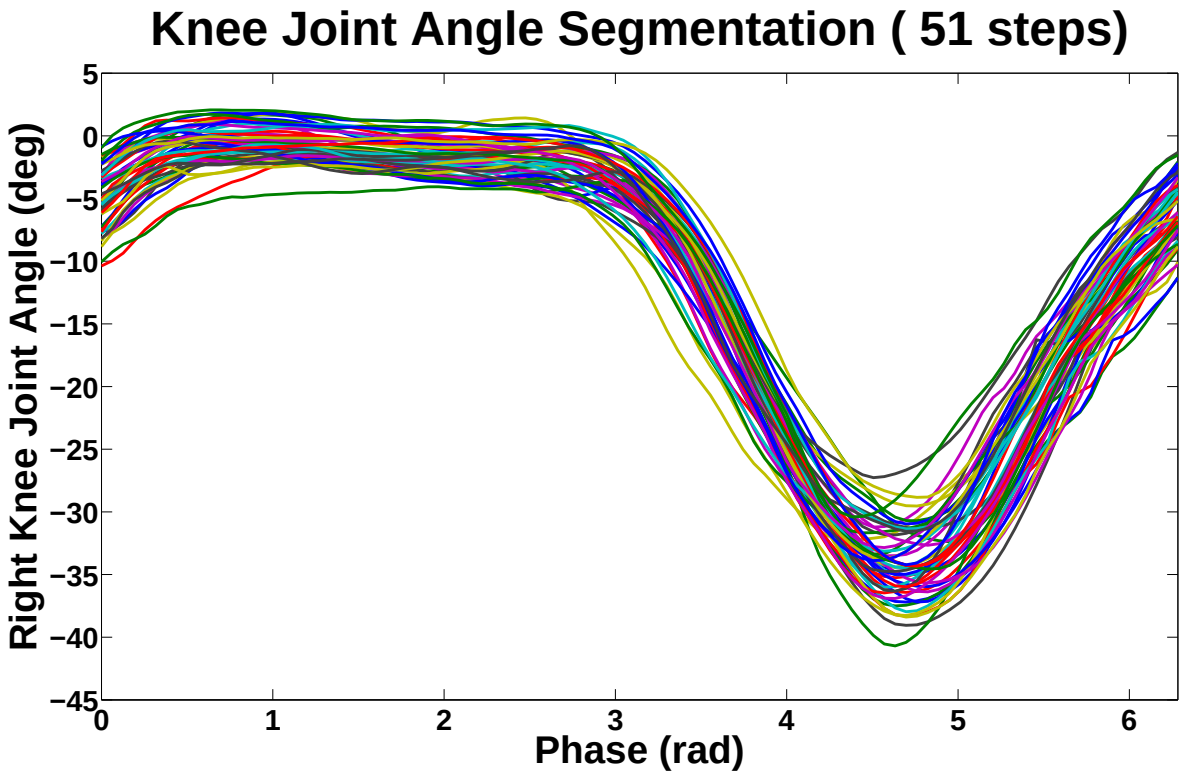


Figure 5.8: Segmentation of estimated joint position using the phase of Rhythmic-EKF. The segmentation begins after 15 seconds of walking to allow Rhythmic-EKF to achieve convergence.

and 97.1% respectively. Figure 5.8 shows the right knee joint angle estimate segmented using the phase variable.

Due to quick convergence of the frequency and phase variables Rhythmic-EKF can handle changing frequencies. A metronome walking trial was collected and the metronome frequency was changed every 30 seconds. The algorithm was able to accurately track the frequency and phase of the motion. The phase was used for segmentation and temporal alignment. Figure 5.9 shows tracking of the frequency and phase and motion segmentation based on the estimated phase.

With the motion segmented and the segments aligned, important measures such as mean and standard deviation of the joint angles can be extracted. Because all of the lower body joints are synchronized to a single phase variable, it is easy to visualize

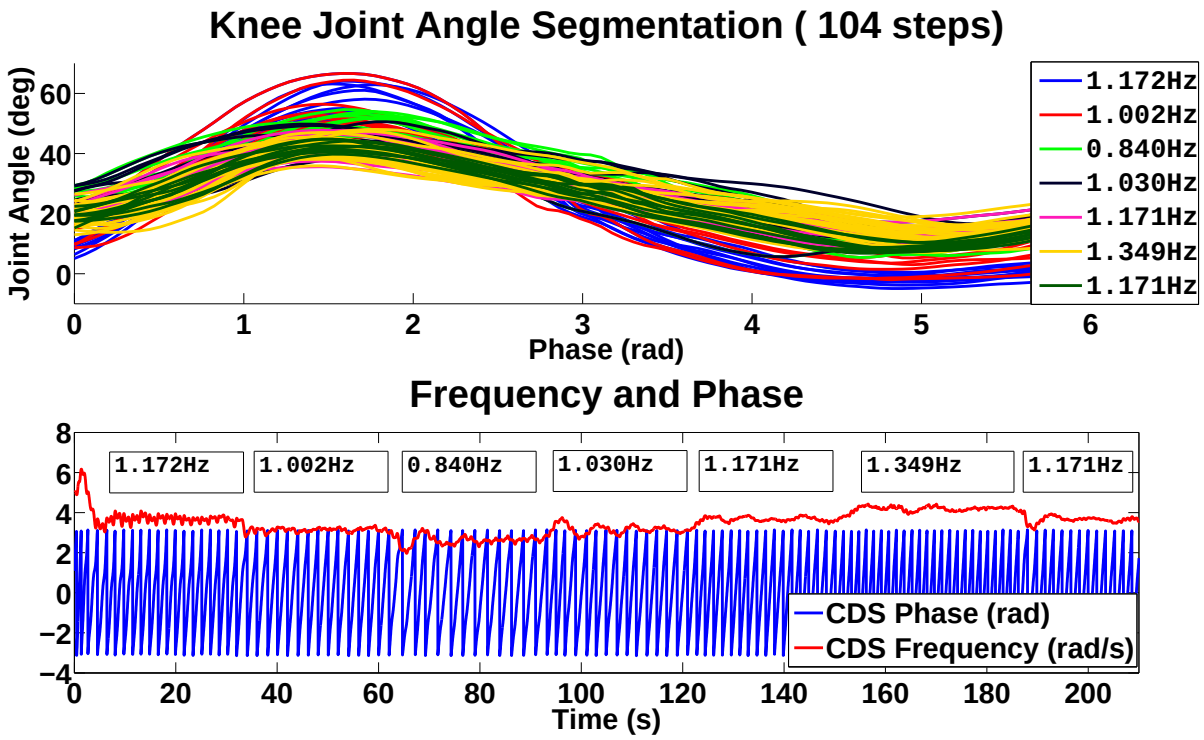


Figure 5.9: Metronome walking trial where the frequency was changed every 30 seconds (1.167Hz, 1Hz, 0.833Hz, 1Hz, 1.167Hz, 1.33Hz, 1.167Hz). Rhythmic-EKF successfully tracks the variable frequencies. The top plot depicts segments temporally aligned and stretched using the phase variable. The bottom plot shows the frequency and phase tracking, as well as mean frequency estimated for each 30 second interval.

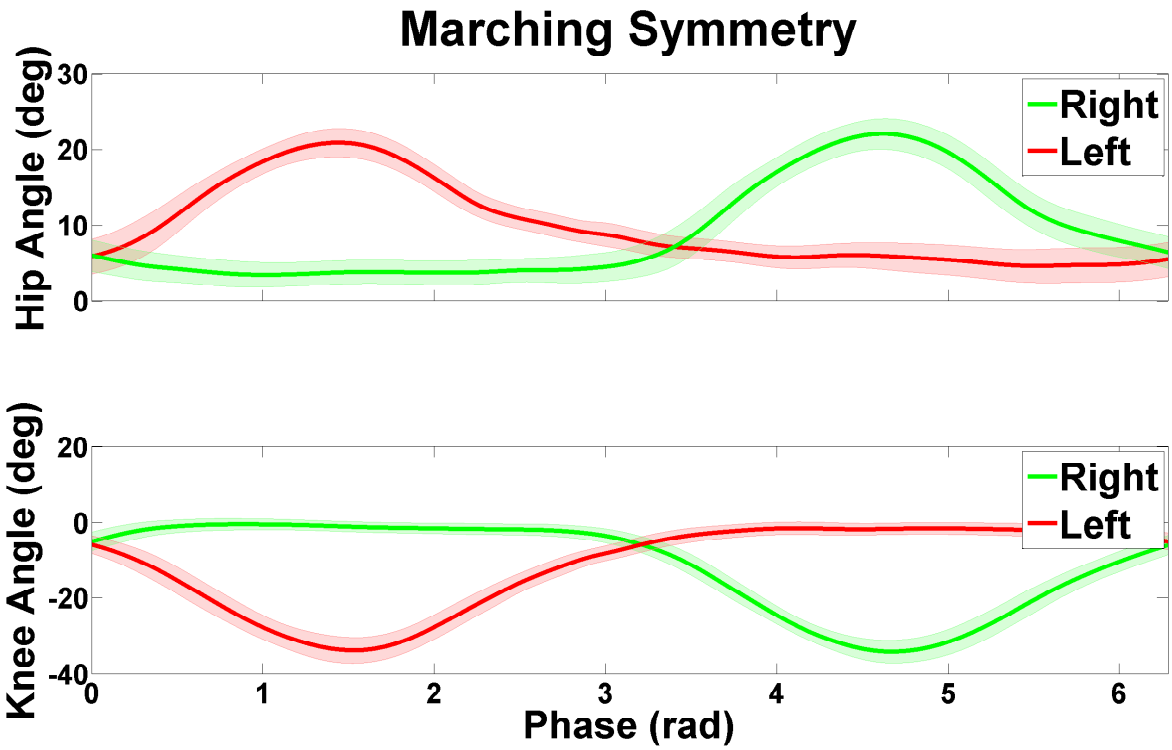


Figure 5.10: Mean and standard deviation of the joint angle segments during walking exercise. Accurate tracking of phase makes temporal alignment of the segments possible.

the symmetry between the left and right sides which is very useful in physiotherapy. Figure 5.10 shows the mean and standard deviation of hip and knee joint angles during the marching exercise.

5.3.3 Rehabilitation Patient Results

While the algorithm works well for healthy participants it is possible that rehabilitation patients do not exhibit rhythmic motions and would not be able to extract the pose as well as meaningful features. To evaluate the suitability of the proposed approach for pathological gait, data has been collected for two patients while they undergo regular gait rehabilitation in the hospital setting. Patient 1, age 67, suffered hemorrhagic stroke

and lost right side mobility. Due to a previous heart attack patient 1 was kept on beta-blockers during the rehabilitation period preventing pulse increase over 120 beats per minute during exercise. Patient 2, age 77, suffered ischemic stroke which also affected his right side. He had no previous morbidities but began experiencing hip pain in the stroke affected side towards the end of the rehabilitation period likely due to developing arthritis because of lost muscle strength. The study has been approved by Waterloo and Grand River Hospital research ethics boards. Specifically IMU data was collected during monthly patient progress assessments, which consists of a 6 minute walking task where the distance walked and step lengths are measured. Data from each IMU was put through a low pass filter of 10Hz to eliminate accelerometer spikes during heel strikes. The kinematic model for each patient was defined using measured pelvis width, depth, and leg length. The same noise and CDS parameters were used as with healthy participants. Since no motion capture data is available to compute the transformation between each link and the IMU frames, a known initial static pose before gait was assumed and orientation with respect to each kinematic link was computed using the gravity vector.

Despite significantly more variance in the motion due to missteps and weak muscle control, the Rhythmic-EKF succesfully converges to the patients' motion and can be used for pose estimation and segmentation. Figure 5.11 shows the convergence of Rhythmic-EKF to the patient's motion within only a few steps.

While the distance walked d during the six minute walking task is trivial to measure, to get the patient's step length the physiotherapist tries to place coins at points where the patient stepped and measures the distance between them. Unfortunately this is a very error prone task as the exact spot where the patient stepped is not clear. Furthermore patients do not have perfect control of their muscles and thus step lengths are likely to have high variance and the physiotherapists only measure one or two due to time constraints. Using the phase variable provided by the Rhythmic-EKF we segment the six minute walking task into discrete steps. It is then possible to calculate average step length $\bar{l}$ using the largest distance between left and right ankle at each step. To verify that that the step length estimate is correct we calculate the walking speed as $f\bar{l}$ and compare it to the average walking speed during the six minute walking task. Figure 5.12 shows that the average step length calculated using Rhythmic-EKF provides an excellent estimate of the patients' walking speed and thus their step lengths.

Stroke patients often rely on a compensatory gait strategy due to lost muscle control and spasticity (muscle tone). Thibaut *et al.* [68] analyzed spasticity patterns in patients and their side effects. Spasticity patterns affecting gait are adducted thigh, stiff knee, flexed knee, and equinovarus foot. These may result in difficulties in limb clearing, extended knee and toe dragging during gait, short step lengths, and deviant knee flexion

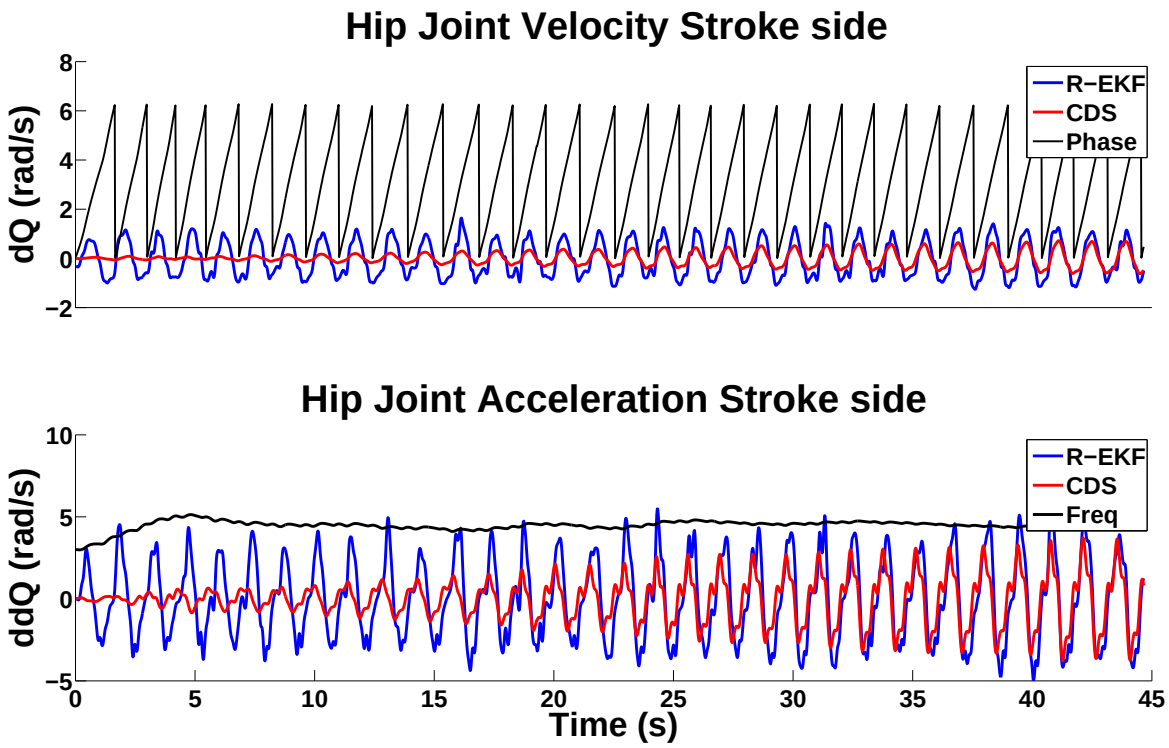


Figure 5.11: Frequency converges to the patient's motion within 5 steps. The Fourier coefficients take longer to converge but begin to improve estimation as soon as frequency is converged.

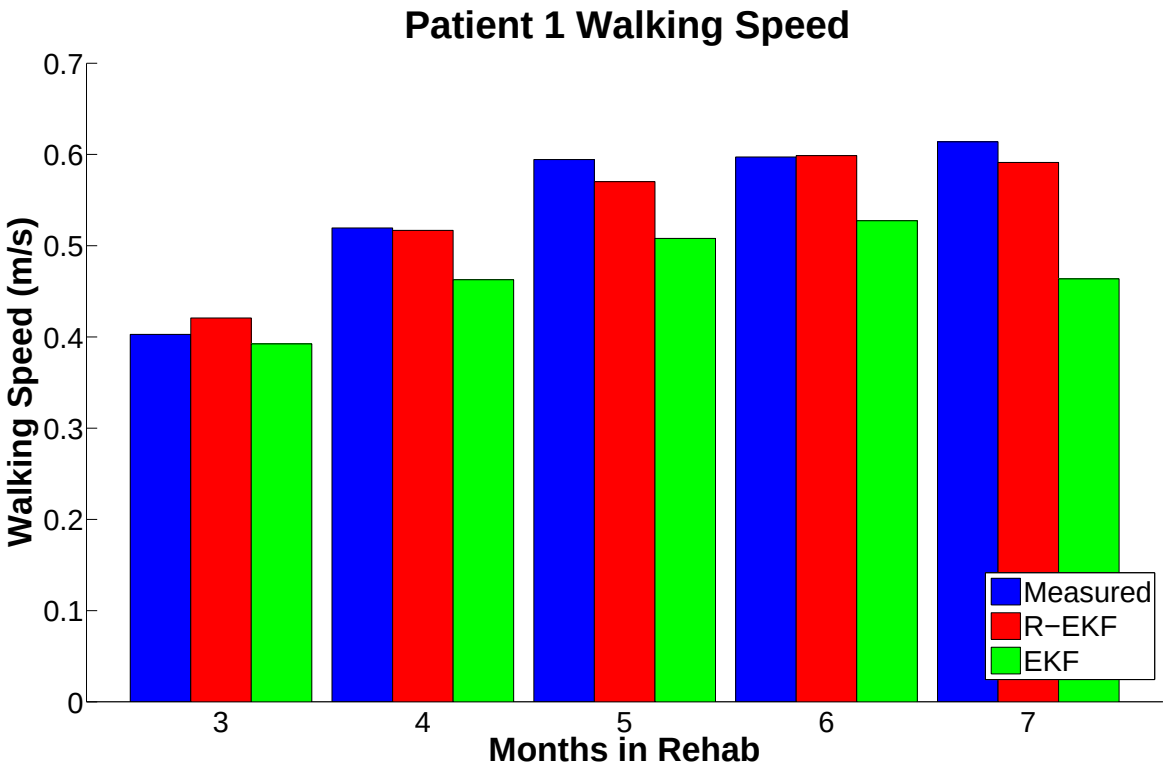


Figure 5.12: Comparison of patient 1 measured and calculated walking speed. The walking speed is computed based on average frequency and step length as estimated by Rhythmic-EKF.

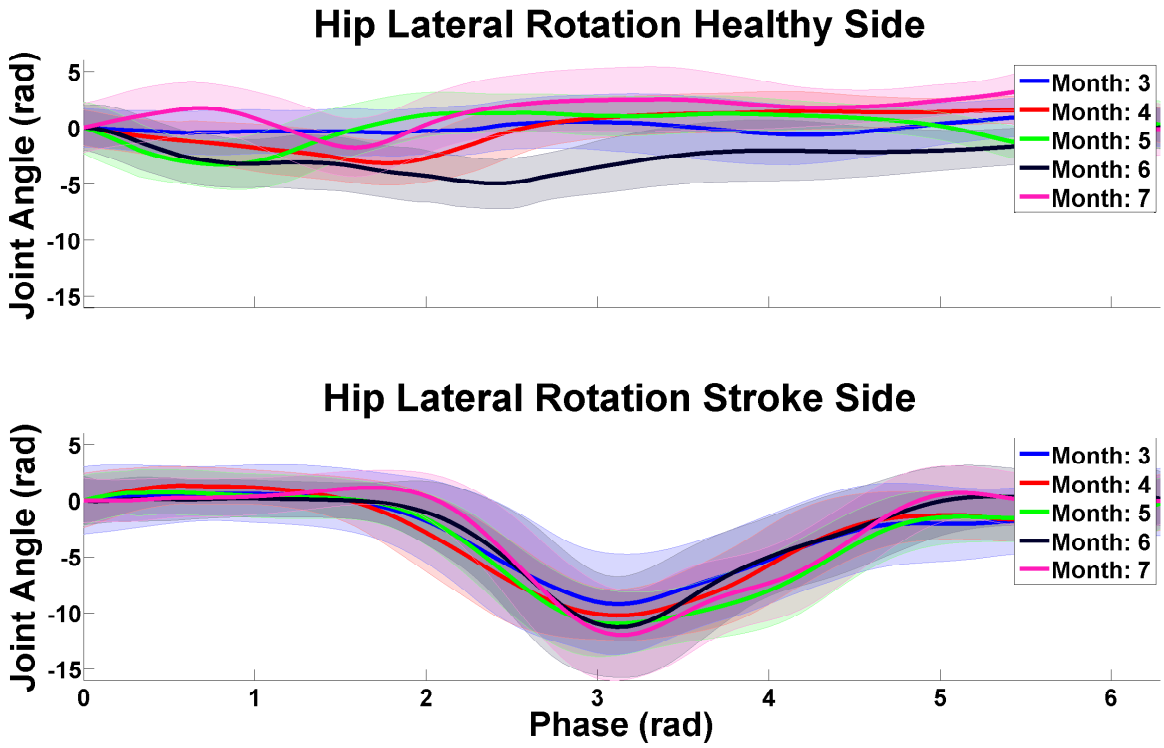


Figure 5.13: Patient 1 hip lateral rotation strategy.

during pre swing respectively. Simple measures such as an increase in speed and step lengths over multiple sessions shows a patient's improved ability to walk but does not reveal whether the patient is regaining lost muscle control or is adapting a compensatory gait strategy. Understanding the compensatory strategies of a patient can help physiotherapists in creating specialized exercise regiments to regain muscle control and natural gait. Computing the average joint angles over a gait cycle for each monthly test allows to look for symmetry between the healthy and stroke affected sides. Patient 1 experiences the stiff knee spasticity pattern and our analysis shows that they did not recover natural gait over the rehabilitation period, instead they adopted a strategy to compensate for difficulty in knee flexion using hip lateral rotation. Figure 5.13 shows the average hip lateral rotation over the rehabilitation period.

Since gait patterns are individual specific it is hard to determine what defines a healthy cycle. For stroke patients therapists often compare the affected side to the healthy side as a measure of progress. Accurate estimate of phase allows shifting the

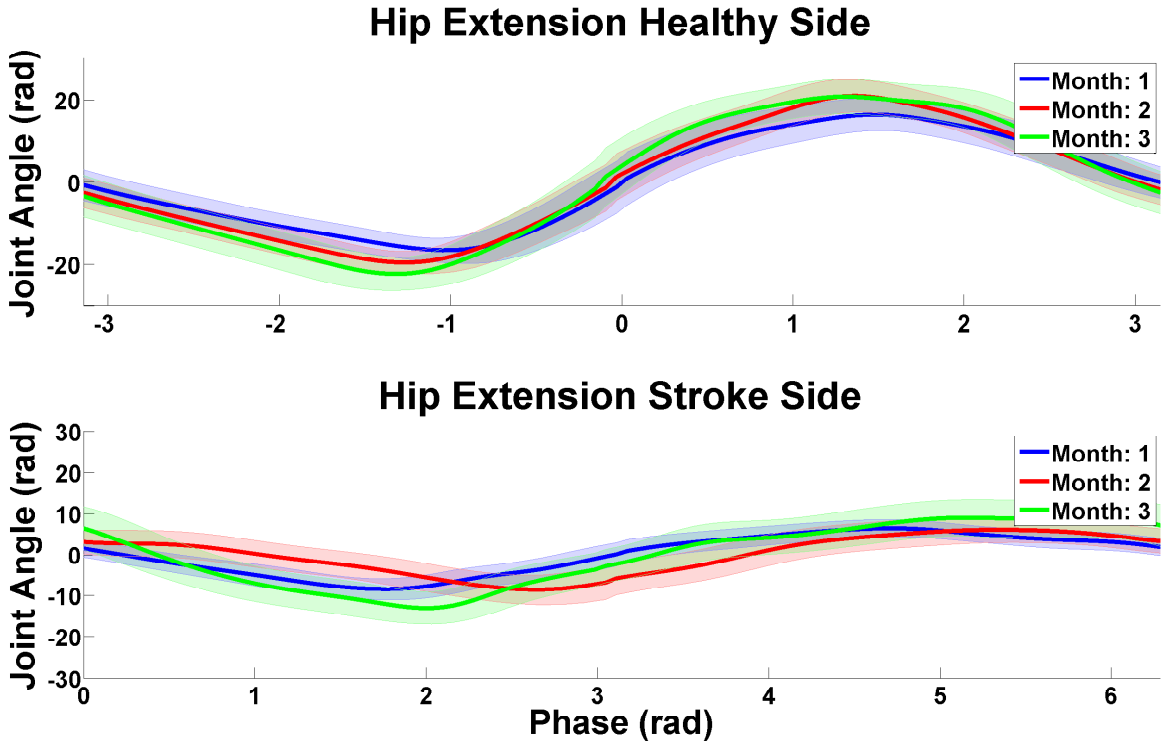


Figure 5.14: Patient 2 healthy and stroke side hip extension comparison.

gait cycle of one side by π for direct comparison. Figure 5.14 shows the mean hip extension comparison between the two sides. Calculating the joint angle root mean squared error between the two sides can help the therapist in patient assessment and progress tracking. Table 5.5 provides the hip and knee extension RMSE between healthy and stroke sides over the rehabilitation period. After the first assessment session patient 1 suffered from increased muscle tone and his performance did not improve. Patient 2 shows slight improvement in hip flexion and extension as seen in figure 5.14 and in the table but no changes in knee flexion.

The accurate pose estimation in combination with phase and frequency information makes the Rhythmic-EKF a well suited approach for gait rehabilitation. Using the estimated pose and frequency walking speed can be computed accurately and used as a performance measure. The estimated phase variable allows for segmentation of gait into cycles from which the patient's mean joint trajectories and variance can be extracted to evaluate their compensatory strategies. For stroke patients the affected side

Table 5.5: Joint angle RMSE between healthy and stroke affected sides over the rehabilitation period. Data for patient 2 month 3 is not available due to failure during that day's measurement session.

	Patient 1		**Patient 2**	
	RMSE between effected and helthy sides (deg)			
Month	**Hip**	**Knee**	**Hip**	**Knee**
1	4.0313	13.5149	12.9751	12.1886
2	10.2263	18.5648	11.5758	13.6418
3	10.8714	19.2319		
4	8.1691	19.3495	11.0128	11.4533
5	12.1272	15.082		

motion can be compared with the healthy side using phase shift allowing the therapists to evaluate patient specific improvement.

Chapter 6

Motion Segmentation

While gait is a very rhythmic movement for both healthy individual and physiotherapy patients, many exercises are not. Rehabilitation regiments may consist of multiple exercises performed in sequence and rehabilitation patients often need longer rest between repetitions. Rhythmic-EKF described in chapter 5 cannot segment such motions. This chapter proposes segmenting joint angle time-series data into exercise repetitions by classifying individual time points as a *segment point* (p_1) or a *non-segment point* (p_0). The proposed method is appealing as this converts the temporal segmentation task, which is traditionally a difficult task, into a more established classification problem. Classifying into segment points or non-segment points allows the algorithm to handle motions that have not been observed before, and be applied to general movements. Once segment points are determined, additional classification can be done to identify the underlying motions. Figure 6.1 outlines the steps of the algorithm [1].

6.1 Pre-processing

To generate a training data set, joint angles and velocities are estimated from IMU data using the method described in section 4.3 and the start and end of each repetition is manually labeled using video playback. In the clinic, *a priori* knowledge of patient movement may not exist, so the classifier must generalize from the training data to

[1] This work was completed in collaboration with Jonathan Feng-Shun Lin and published in IEEE Engineering in Medicine and Biology Conference 2014 [43]

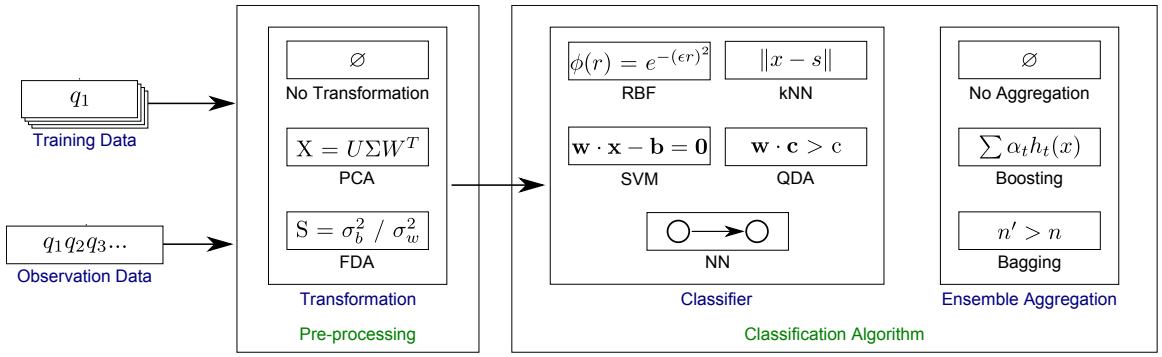


Figure 6.1: Flow diagram for the training and classifying task. A given instance of the algorithm will consist of a dimensionality transformation component, a base classifier, and an aggregation component.

the target patient population. Prior to segmentation, pre-processing is performed to normalize and balance the data and remove outliers, as detailed below:

6.1.1 Normalization

For human motion data, consideration must be given to the inter- and intra-participant variability, due to differences in physiology, fatigue, familiarity with the motion, and the participant's initial posture.

To reduce the impact of this variability, the mean of the joint angles is removed from each motion. Another common normalization technique is to normalize to unity covariance, but initial testing revealed that this adversely impacted the performance of classifiers that utilize the variance, and thus was not employed.

6.1.2 Manual Segment Point Shifting

Two common methods for manual segment generation for exercise motions are (1) by video playback, while an observing expert denote the time points where p_1 occurs, or (2) the entire time series data is presented as a graph, and an expert annotates the p_1 points. The former method is preferred, as it incorporates the expert's intuitive understanding of a segment. However, this method of creating manual segment is time-consuming, and is prone to errors from viewing angle and reaction speed.

To reduce the impact of manual rater reaction delay, the manual segments are shifted backwards in time to the closest joint zero-velocity crossing (ZVC), if a ZVC exists within 0.5 seconds of a manual segment. No shifting is done if no ZVC is found.

6.1.3 Manual Segment Point Expansion

The manual segments only declare a single time point to denote the start and end of a segment. For a classifier, this is not a suitable approach, as there is likely minimal difference between the data at a given time t_n and the data at time t_{n+1}. The data described by the p_1 points denotes characteristics such as joint angle turning points, and shares these characteristics with more data points than the ones denoted by the manual p_1. Additional data points are added to better represent this.

Two different techniques are applied to help with this issue: (1) The data points between the end of one segment and the start of the next are all labeled as p_1. In exercise motions, where the subject performs a given motion multiple times in succession, the subject is not likely to be performing spurious movements between each repetition. This allows for additional training data points to be added, that share similar characteristics as the manual p_1. (2) An additional n_{exp} points before and after each cluster of segment points are also denoted as p_1. This n_{exp} is arbitrarily set, and requires tuning. See Figure 6.2 for an illustration of the manual segment point shifting and expansion.

6.1.4 Outlier rejection

The quality of the EKF data is checked by examining its velocity, in order to remove velocity spikes. In a given exemplar, the peak velocity for each segment is determined, and clustered by k-means, where $k = 2$. It is assumed that only one or two segments will contain poorly estimated joint angles, and that the participant's movement profile does not vary dramatically from one repetition to the next, thus joint angle spikes are considered outliers. If the higher cluster in the k-means contains only one or two peaks, the segments containing these velocity spikes are removed from consideration for training. Data between t_1 and the start of the first segment is also rejected from the training data, since EKF may take some time to converge to the actual joint angles, depending on the quality of the EKF initialization. To increase robustness of this velocity spike rejection, only a subset of all the input features were examined. The joint angles examined are determined by calculating the variance of the exemplar, over the whole dataset, and selecting the joint angle that varies the most.

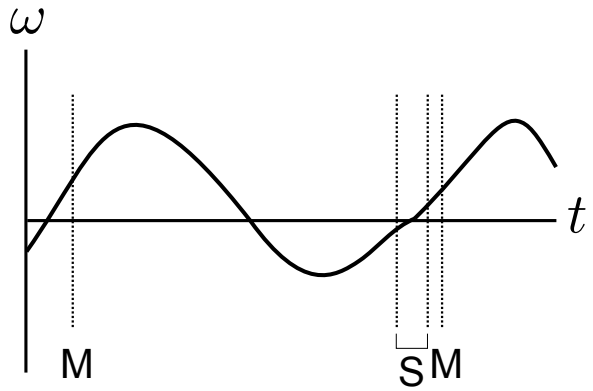


Figure 6.2: A time-series waveform denoting joint velocity ω. The M point denotes the location marked by a manual segment, a p_1. Due to the way that the manual segmentation is done, the manual segment may not be in the proper location. The manual segment can be shifted to the closest zero-velocity crossing, and all points in a region around the zero-velocity crossing, S, are declared as segment points.

6.1.5 Input vector stacking

Classifier techniques do not typically consider temporal factors. Although the input vector could include data with a temporal nature, such as joint velocity, additional temporal data may be required to adequately capture the temporal nature of the movement data. The input vector is stacked, so that a given data point includes data from a few time steps before and after the current data point. That is, $t_{use} = [t_{n-n_{stack}} \cdots t_{n-1}, t_n, t_{n+1} \cdots t_{n+n_{stack}}]$. This provides an additional source of temporal data into the classifiers. n_{stack} is another variable that requires tuning to optimize for the data.

6.1.6 Downsampling

The proposed algorithm must be capable of handling unbalanced datasets, where there are significantly more data samples of a given label [74]. For example, in a given set of training data, data points labeled as a p_1 are uncommon, when compared to p_0. In addition, if different exercises are used in the training, the training set could end up with more data from a slow moving participant, and skew the training data.

To reduce the impact of unbalanced data, two layers of downsampling are employed. (1) The number of p_1 and p_0 for each exemplar is noted, and the smallest values are denoted as $p_{1_{min}}$ and $p_{0_{min}}$, respectively. Each exemplar will be randomly downsampled

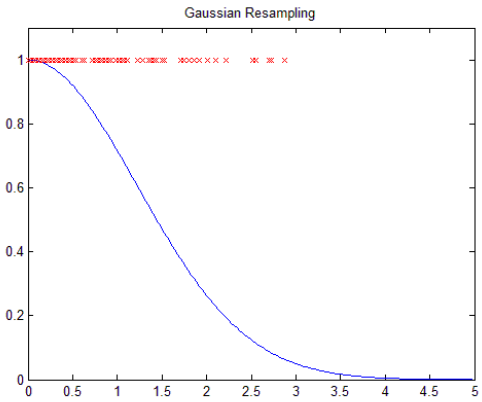


Figure 6.3: The blue curve denotes a Gaussian distribution, where $x = 0$ denotes the location of a p_1. The red points denote the training data selected for p_0.

to match $p_{1_{min}}$ and $p_{0_{min}}$, to eliminate the unbalanced dataset. (2) The full training data is resampled to make sure it does not exceed a limit of 7500 data points for either p_1 or p_0, as MATLAB is unable to create arrays larger than 7500^2, which is required for certain algorithms. For p_1, it is randomly resampled. For p_0, Gaussian resampling to favour p_0 points close to existing p_1 is used. See Figure 6.3 for an illustration.

6.2 Model Training

Several different algorithms are assessed to see if they are capable of performing this classification task. Cross validation is used to verify the results, where a portion of the data is used as training exemplars for the algorithm, and the remaining dataset is used for testing. It is likely that the algorithms examined will perform better with motion specific templates (that is, n_t templates constructed for n_m different motions) than a single unifying template (that is, 1 template constructed for n_m different motions), so both schemes are tested for comparison purposes. Common approaches to classification involve either utilizing dimensionality reduction or feature selection, then using a simple classifier, or to forgo any dimensional alternation and use a powerful classifier instead. We have split the testbed into three main sections, which can be found in Figure 6.1.

6.3 Testing on data unseen during training

To classify, each set of observation data is pre-processed in a similar fashion as the training data. The mean is subtracted from the data, the manual p_1 are expanded, and the input vector is stacked with data points from before and after the time point itself. If a dimensionality reduction algorithm is used, it is then applied, before classification begins. If n_{stack} data stacking was used, then time points t_1 to $t_{n_{stack}}$ and $t_{n-n_{stack}}$ to t_n cannot be classified since not enough data is available to be stacked, and thus are excluded from the classification effort.

6.4 Verification

Each point in the observation is labeled p_1 or p_0. The number of correctly identified segment points (TP), as well as all false positives (FP) and false negatives (FN), are aggregated together and reported as the F_1 score, which represents a measure that aggregates both precision and recall accuracy. The p_0 points are not considered as the larger volume of the p_0 points may obscure the true accuracy of the model. The F_1 score is calculated as follows:

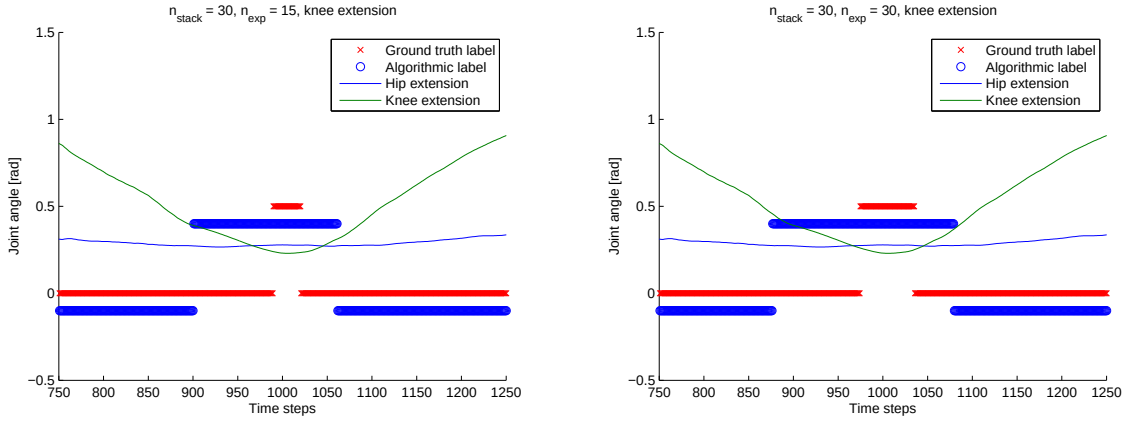
$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

6.5 Experimental Results

This section describes the experimental results of the motion segmentation algorithm proposed. The testing data came from a database of 20 healthy participants performing 5 rehabilitation motion types each. The exercises performed were: knee extension while seated, sit to stand, squats, knee/hip flexion while supine and hip extension while supine. The data was collected via 3 Shimmer IMU sensors [18], transmitting at 128 Hz. Joint angles were estimated from IMU data using the EKF algorithm described in section 4.3. The experiment was approved by the University of Waterloo Research Ethics Board, and signed consent was obtained from all participants. Manual segments of the exercises were generated and labeled by experts watching the motion in video playback of motion capture. The data from 5 healthy participants were used for training, then tested against another 5 healthy participants.

A series of batch testing was performed in order to narrow down the parameter search space: (1) Input space parameter search, (2) Classifier parameter search with a single exercise, (3) Classifier parameter search with multiple exercises.

6.5.1 Parameter Tuning



(a) $n_{exp} = 15$

(b) $n_{exp} = 30$

Figure 6.4: Segmented knee extension motion, with $n_{stack} = 30$. Points at around the $y = 0$ area are ground truth (red) and algorithmic (blue) points denoted as p_0, while points at around $y = 0.5$ are p_1 points. Notice that, for Figure 6.4(b), while the ground truth points increased, so did the span for the algorithmic segment points, since the training points have expanded.

Initial testing was performed with the following input data parameters, to clarify a suitable parameter space:

- Training and testing data: knee extension
- Combinations of input datatype: combinations of joint angle q, joint velocity $\dot{q}$ and/or joint acceleration $\ddot{q}$
- Manual segment point expansion: $n_{exp} = [0, 3, 9, 15, 30]$
- Input vector stacking: $n_{stack} = [0, 3, 9, 15, 30]$

The SVM classifier was used, with no dimensionality reduction and no aggregator. Since SVM only relies on a few support vectors close to the segmentation boundary of the training set it is expected to work well even with large motion variations as long as the start and end of the motion are similar.

As can be noted from Table 6.1, having both the joint angle and the joint velocity is a successful combination, as well as having a large n_{exp}, while n_{stack} seems less influential. Comparing the rank 1 entry to the rank 22 entry (not shown in Table 6.1), which has no temporal information, with $n_{stack} = 0$ and with only the q vector, including temporal information improves the performance by over 10%. However, many entries with temporal information perform very poorly, particularly vectors with only $\dot{q}$ and $\ddot{q}$. Although not shown, the data with only $\dot{q}$ and $\ddot{q}$ correlate with high F_1 training score, but low F_1 testing score, suggesting overfitting. These results indicate that future tests should include q and $\dot{q}$, with a non-zero n_{stack}, but the selection of n_{stack} does not seem to have a heavy impact on the outcome. n_{stack} was set to 15 for subsequent tests, to balance between the high runtime required for $n_{stack} = 30$ and accuracy.

The selection of n_{exp} is less straight forward. n_{exp} changes the number of manual p_1 points, thus altering the ground truth data. Evidently, the higher n_{exp}, the more p_1 training points became available (see Figure 6.4(a) and 6.4(b)). n_{exp} cannot be scaled upwards indefinitely, or else the entire training set would be p_1 points. After examining the results, n_{exp} was to be 12, which correlates to 0.18 seconds, to match prior work on this dataset [42], who used 0.2 seconds for error tolerance. Each repetition is, on average, 4 seconds long, making the total p_1 region less than 10% of a repetition.

6.5.2 Single Exercise Segmentation

The following range of parameters are tested in the second batch:

Input data:

- Training and testing data: knee extension
- Input datatype: Joint angle q and joint velocity $\dot{q}$
- Manual segment point expansion: $n_{exp} = [12]$
- Input vector stacking: $n_{stack} = [15]$

Dimensionality reduction algorithms:

Table 6.1: Results for test set 1, where SVM was used to classify segment points on the knee extension motion. The F_1 and accuracy scores for training and testing for each batch 1 combination are presented. For brevity, only the top 10 results is reported, the entire table is shown in appendix A.1. The time taken accounts for both training and testing time.

	Input data					**Training**	**Testing**			
Rank	n_{stack}	n_{exp}	**Data type**			F_1	**Total**	**Correct**	F_1	**Time taken [s]**
1	30	30	q	$\dot{q}$		94%	9112	8571 (94%)	86%	563
2	30	30		$\dot{q}$		94%	9112	8542 (94%)	82%	325
3	15	30	q	$\dot{q}$		93%	9412	8698 (92%)	82%	370
4	30	15	q	$\dot{q}$		93%	6099	5780 (95%)	81%	641
5	15	15	q	$\dot{q}$		93%	6382	6041 (95%)	81%	373
6	9	30	q	$\dot{q}$		93%	9532	8679 (91%)	81%	220
7	3	30	q	$\dot{q}$		92%	9652	8745 (91%)	80%	79

- No dimensionality reduction
- PCA and FDA: set by elbow at 80% (denoted as 0 in the results) or set the dimensions to 2
- ksPCA: set the dimensions to $d = [2, 6, 30]$ and $\sigma = [0.1, 0.4, 0.8, 3.2, 6.4, 12.8, 25.6, 51.2]$

Classifier algorithm:

- k-NN: $k = [3, 9]$
- RBF: number of RBFs used, $n_{RBF} = [10, 20]$
- SVM: kernel function used: linear, polynomial, radial
- ANN: layers and neuron count, $n_{layers} = [10, 10], [10, 10, 10], [20, 20, 20]$

Aggregator algorithm:

- No aggregator
- Boosting: iteration count $n_{iter} = [3, 5]$

- Bagging: iteration count $n_{iter} = [3, 5]$

ksPCA requires applying a kernel to n training vectors and eigenvalue decomposition of an $n \times n$ matrix [5]. Memory constraints on MATLAB and the size of the training data prevented ksPCA from being tested optimally. As a result, ksPCA was ran without any input kernels and a delta kernel for the labels, effectively reducing ksPCA to only sPCA. Although it is expected that ksPCA would perform at least comparably to PCA, the training and testing F_1 scores was consistently lower. With the PCA retained dimensions set to 30, which is roughly the number of dimensions kept when PCA dimensions are set by the elbow technique, PCA SVM achieved a training F_1 of 97% and a testing F_1 of 86%, while ksPCA achieved a training F_1 of 90% and a testing F_1 of 69%. This suggests that ksPCA is overfitting. Lowering the number of training data to 2000 allows ksPCA to be ran with a linear or RBF kernel. This results in an high training F_1 of 98%, but still only a testing F_1 of 57%. With a larger set of training points, it is possible that ksPCA would perform significantly better.

Table 6.2 shows that SVM with radial kernel performs the best (see Figure 6.5). Examining the other top-scoring techniques such as ANN or k-NN, it can be noted that the training F_1 is higher for these other techniques than the SVM, even though the testing F_1 is lower. This shows that the ANN algorithm is overfitting.

Table 6.2 also shows that aggregated techniques do not have a significant impact on segmentation accuracy, as the top 5 classifiers are all radial SVM, but with different aggregators. This suggests that the training data sampling scheme is sufficient, since the sampling scheme emphasizes the p_0 points close to the p_1, which are likely to have a higher chance of misclassification to begin with.

6.5.3 Multiple Exercise Segmentation

The same parameters for test set 2 were used here. Instead of using a single exercise as training and testing, 5 different exercises were used instead: knee extension, sit to stand, squats, supine hip extension, supine knee hip flexion.

From Table 6.3, it is noted that k-NN performs well, even though the dataset examined is not well separable. The exercise segments are characterized by a constant resting q and $\dot{q}$, so when the observation data presents an input vector that is close to the appropriate q and $\dot{q}$, k-NN is able to label these points appropriately. This, however, suggests that k-NN only performed well due to the specific characteristics of this dataset, and

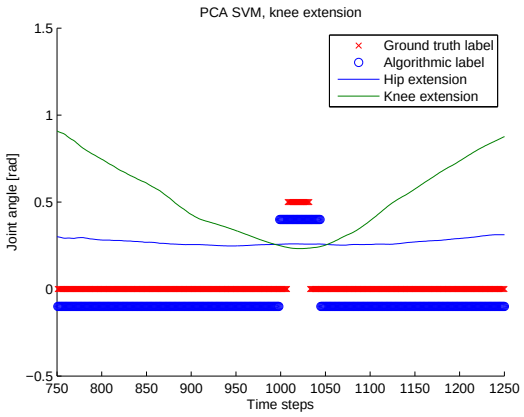


Figure 6.5: Segmented knee extension motion, with $n_{exp} = 12$ and $n_{stack} = 15$. Segmentation performed with PCA SVM, with no aggregator. Points at around the $y = 0$ area are ground truth (red) and algorithmic (blue) points denoted as p_0, while points at around $y = 0.5$ are p_1 points.

Table 6.2: Results for test set 2, where the input vector consist of q and $\dot{q}$, $n_{exp} = 12$ and $n_{stack} = 15$, and the exercise examined is knee extension flexion. The F_1 and accuracy scores for training and testing for each combination is presented. For brevity, only the top 10 results are reported, the entire table is presented in appendix A.2. The time taken accounts for both training and testing time.

	Classifier parameter			**Training**	**Testing (Total correct p_1 = 5776)**		
Rank	**Dim Reduct**	**Classifier**	**Aggregator**	F_1	**Correct**	F_1	**Time taken [s]**
1	PCA, 0	SVM, radial	Boosting, 3	96%	5391 (93%)	87%	156
2	PCA, 0	SVM, radial	Boosting, 5	97%	5381 (93%)	87%	194
3	PCA, 0	SVM, radial	-	96%	5318 (92%)	87%	75
4	PCA, 0	SVM, radial	Bagging, 3	96%	5347 (93%)	86%	133
5	PCA, 0	SVM, radial	Bagging, 5	96%	5327 (92%)	86%	206
6	PCA, 0	ANN, 20-20-20	Bagging, 3	99%	5052 (88%)	86%	1853
7	PCA, 0	ANN, 10-10	Bagging, 5	99%	5031 (87%)	85%	956

may not generalize well to other datasets that are not easily separable. Similar to Section 6.5.2, radial SVM performed well in this case as well. ANN did not perform well, likely due to overfitting.

It is also worthwhile to note that the segmentation performance can be improved by removing contradictory motions. The motions sit to stand, knee hip flexion, and squat, are all movements that require both the hip and knee to move. The resting pose (and thus the segment points) for the knee-hip flexion and the sit to stand motions is the same, with the knees bent, and goes through similar motions as the squat, but the resting pose for the squat is with the knees straight. These different motions sometimes confuse the classifier, causing false positives. This suggests that a single classifier may not be appropriate to segment all motions, and that multiple classifiers may be required. Alternatively, splitting a segment window in half, so that each segment consists only of a half motion, may help address this problem.

Table 6.3 also suggests that the aggregators do not help significantly in the classifying effort, since the top 10 entries consist of the same classifiers but with different aggregators, with no appreciable difference in F_1 score.

Table 6.3: Results for test set 3, where the input vector consist of q and $\dot{q}$, $n_{exp} = 12$ and $n_{stack} = 15$, and the exercises examined were 5 lower body rehabilitation exercises. The F_1 and accuracy scores for training and testing for each combination is presented. For brevity, only the top 10 results is reported, the entire table is presented in appendix A.3. The time taken accounts for both training and testing time.

	Classifier parameter			**Training**	**Testing (Total correct p_1 = 35819)**		
Rank	**Dim Reduct**	**Classifier**	**Aggregator**	F_1	**Correct**	F_1	**Time taken [s]**
1	PCA, 0	k-NN, 9	-	95%	29773 (83%)	76%	1343
2	PCA, 0	k-NN, 9	Boosting, 5	95%	30020 (84%)	76%	5194
3	PCA, 0	SVM, radial	-	91%	30881 (86%)	76%	275
4	PCA, 0	k-NN, 9	Boosting, 3	95%	29851 (83%)	76%	3910
5	PCA, 0	k-NN, 9	Bagging, 3	93%	30265 (85%)	76%	3694
6	-	k-NN, 3	Bagging, 3	95%	28998 (81%)	76%	32245
7	-	k-NN, 9	Bagging, 3	93%	29905 (84%)	76%	27157

6.5.4 Summary

The results show that a time-series segmentation task can be reformulated as a two-class classification problem. Using PCA SVM, knee extension extension exercise segment points can be detected with a F_1 accuracy of 87%. When using 5 different exercise

motions as training, PCA SVM can detect segment points with a F_1 accuracy of 76%. The top performing classifiers overwhelmingly utilize PCA confirming that the time-series data is redundant and correlated. The classifiers with the highest $F_{1_{Seg}}$ score were the SVM, the ANN and the k-NN. These classifiers provided high accuracy in both the individual and multi-exercise tests, suggesting that generalization between exercises is possible. However, k-NN is computationally expensive and is unsuitable for on-line applications. Aggregators have not improved segmentation accuracy, the top 10 performing classifiers include both-aggregated and non-aggregated variants. We attribute this to a type of aggregation already performed in the preprocessing step by the sampling technique, which preferably selects non-segment points closer to the segmentation boundary for training.

Chapter 7

Conclusions and Future Work

7.1 Summary

This thesis proposes pose estimation and segmentation algorithms for an automated rehabilitation system using wearable inertial measurement units. It describes two different lower body modeling approaches, and proposes two pose estimation algorithms. The first approach uses the constant acceleration assumption and the kinematic models of the body to formulate an Extended Kalman filter framework for estimating joint angles, velocities and accelerations. The second algorithm improves upon the first by learning a model of rhythmic movement that removes the constant acceleration assumption. The approach is capable of estimating lower body pose and segmenting rhythmic motions into exercise repetitions. For non rhythmic motion a classifier based method is developed to segment estimated pose into repetitions. Validation is performed in simulation, on healthy participant data, and stroke patient monthly assessment data.

Lower Body Modeling and Pose Estimation

The majority of kinematic based approaches for pose estimation assume a stationary base and thus cannot handle accurate pose and global position estimation for gait exercises. This thesis develops two different approaches for using branched kinematics to model the human lower body with a moving base. The first approach describes the transformation from world frame to the base frame as three prismatic and revolute joints. The second approach relies on switching the base of the kinematic model

when a heel strike is detected. Using either of the two kinematic models, the approach from [44] is adapted to estimate the joint position, velocity and acceleration during non-stationary movement.

A major benefit of the first approach is that the movement of the base is built into the model. Thus, it is only necessary to estimate the joint angles, velocities, and accelerations to fully describe how the patient is moving. However, the extra joints needed to represent the moving base introduce additional uncertainty. Also, this model is not bounded so if there is an error in velocity or acceleration in the first three prismatic joints and integration is used to position the model in 3d space the errors will quickly accumulate causing the model to move out of the observable region.

Since the second approach binds the model to the floor, measurement errors do not cause it to exit the observable region and it provides more accurate results when only IMU sensors are used, avoiding the need for additional position sensors. This model also has less uncertainty due to a smaller number of degrees of freedom. The main drawback is that a step detection algorithm needs to be employed.

The switching base model is shown to be superior in pose estimation.

Rhythmic Extended Kalman Filter

An accurate motion model can be used to extract objective performance measures of the patient during gait, and also to improve pose estimation performance. Previous methods consider the pose estimation and motion model learning as two separate steps. The developed Rhythmic extended Kalman filter algorithm is able to learn rhythmic motion models online and use the learned model to improve pose estimation. It also segments the motion into repetitions and extracts useful objective measures. The performance of the Rhythmic and regular extended Kalman filters is compared in simulation, with healthy participant gait, and stroke patients undergoing rehabilitation. Rhythmic extended Kalman filter is shown to estimate joint angles with 2.4° root mean squared error and segment the motion into repetitions with 96% accuracy.

Motion Segmentation

For non rhythmic motions, segmentation is typically done by a priori learning of a motion template for each possible exercise or using time-series features such as zero velocity crossings. Template based methods require the motion to be known to perform

segmentation into repetitions, feature based methods tend to over segment and are difficult to tune for whole body movements. This thesis proposes reformulating time-series segmentation as a two-class classification problem. Well developed classifiers are used to segment dimensionally reduced time-series data. The proposed approach enables generalization between exercises and subjects. After training on data from 5 participants, using principal component analysis dimensionality reduction and the support vector machine classifier, knee extension exercise segment points are detected with a F_1 accuracy of 87% for 5 previously unseen participants. The algorithm achieves an F_1 accuracy of 76% when the specific exercise motion is not known but is one of 5 possible motions.

7.2 Future Work

7.2.1 Lower Body Modeling and Pose Estimation

The current methods makes the assumption that sensors are rigidly attached to links in the kinematic model. However due to soft tissues and muscle flexion the sensors may shift as the exercise is performed. To improve pose estimation a more complex kinematic model is needed which takes into account possible sensor movement.

In the clinical setting, sensor placement and individualized patient modeling is important. Currently it is assumed that the patient is standing still for some initial time and the sensor rotations around world x and y axis are computed based on the measured gravity vector. There is no method to compute sensor's rotation about the z axis. Patient's hip joint centers are calculated based on their pelvis width, depth, and leg length. A method to compute sensor attachment and hip joint centers based on a known calibration movement is a direction for future work.

7.2.2 Rhythmic Extended Kalman Filter

Future work will focus on adaptive estimation of process noise parameters in Rhythmic-EKF. As the algorithm learns the rhythmic motion it is expected that the state update prediction error decreases and thus the process noise parameters should adapt accordingly. An online adaptation of noise parameters is expected to significantly improve pose estimation.

Currently the heel strike detection algorithm requires large acceleration spikes in the ankle sensor data, does not run real time, and requires manual correction. The ankle acceleration spikes may not be present in slow walking and scenarios with soft contact due to ground or foot wear material. The estimated phase of gait cycle can be used to improve heel strike detection. Future work will combine the proposed time-series classification with the Rhythmic-EKF to accurately detect heel strikes in real time and switch the base of the model. The different classes can be defined as various gait cycle events of interest such as heel strike, swing, and toe off, and the ankle accelerometer signal as well as the estimated phase and pose can be used as part of the feature vector.

Finally it is important to investigate applicability of Rhythmic-EKF to different periodic, almost periodic, and non periodic signals. Developing a quantitative measure based on the measurement signal to decide if EKF or Rhythmic-EKF will result in better state estimation can be very useful.

7.2.3 Motion Segmentation

Generalization of the algorithm to exercises not present in the training dataset will be evaluated. To test the performance for the target population the algorithm will be applied to a patient dataset.

APPENDICES

Appendix A

Motion Segmentation Detailed Results

A.1 Parameter Tuning

Table A.1: Results for test set 1, where SVM was used to classify segment points on the knee extension motion. The F_1 and accuracy scores for training and testing for each batch 1 combination are presented. The time taken accounts for both training and testing time.

	Input data					**Training**	**Testing**			
Rank	n_{stack}	n_{exp}	**Data type**			F_1	**Total**	**Correct**	F_1	**Time taken [s]**
1	30	30	q	$\dot{q}$		94%	9112	8571 (94%)	86%	563
2	30	30		$\dot{q}$		94%	9112	8542 (94%)	82%	325
3	15	30	q	$\dot{q}$		93%	9412	8698 (92%)	82%	370
4	30	15	q	$\dot{q}$		93%	6099	5780 (95%)	81%	641
5	15	15	q	$\dot{q}$		93%	6382	6041 (95%)	81%	373
6	9	30	q	$\dot{q}$		93%	9532	8679 (91%)	81%	220
7	3	30	q	$\dot{q}$		92%	9652	8745 (91%)	80%	79
8	0	30	q	$\dot{q}$		91%	9712	8638 (89%)	80%	31
9	9	15	q	$\dot{q}$		92%	6502	6127 (94%)	79%	227
10	3	15	q	$\dot{q}$		91%	6622	6158 (93%)	78%	84
11	15	9	q	$\dot{q}$		92%	5172	4902 (95%)	77%	360
12	0	15	q	$\dot{q}$		91%	6682	6160 (92%)	77%	33
13	15	30		$\dot{q}$		90%	9412	8488 (90%)	77%	253
14	9	9	q	$\dot{q}$		92%	5290	4996 (94%)	77%	235
15	30	9	q	$\dot{q}$		92%	4899	4603 (94%)	77%	722
16	3	9	q	$\dot{q}$		91%	5410	5103 (94%)	76%	85
17	0	9	q	$\dot{q}$		91%	5470	5108 (93%)	75%	37
18	30	30	q			90%	9112	7898 (87%)	74%	430
19	9	30		$\dot{q}$		88%	9532	8526 (89%)	74%	184
20	9	30	q			89%	9532	8149 (86%)	74%	160
21	15	30	q			90%	9412	8045 (86%)	74%	258
22	0	30	q			89%	9712	8194 (84%)	73%	31
23	3	30	q			89%	9652	8134 (84%)	73%	60
24	30	0	q	$\dot{q}$		93%	3099	2801 (90%)	72%	575
25	0	30	q	$\dot{q}$	$\ddot{q}$	92%	9712	7476 (77%)	72%	79

A.2 Single Exercise Segmentation

Table A.2: Results for test set 2, where the input vector consist of q and $\dot{q}$, $n_{exp} = 12$ and $n_{stack} = 15$, and the exercise examined is knee extension flexion. The F_1 and accuracy scores for training and testing for each combination is presented. The time taken accounts for both training and testing time.

	Classifier parameter			**Training**	**Testing (Total correct p_1 = 5776)**		
Rank	**Dim Reduct**	**Classifier**	**Aggregator**	F_1	**Correct**	F_1	**Time taken [s]**
1	PCA, 0	SVM, radial	Boosting, 3	96%	5391 (93%)	87%	156
2	PCA, 0	SVM, radial	Boosting, 5	97%	5381 (93%)	87%	194
3	PCA, 0	SVM, radial	-	96%	5318 (92%)	87%	75
4	PCA, 0	SVM, radial	Bagging, 3	96%	5347 (93%)	86%	133
5	PCA, 0	SVM, radial	Bagging, 5	96%	5327 (92%)	86%	206
6	PCA, 0	ANN, 20-20-20	Bagging, 3	99%	5052 (88%)	86%	1853
7	PCA, 0	ANN, 10-10	Bagging, 5	99%	5031 (87%)	85%	956
8	PCA, 0	k-NN, 9	Bagging, 5	98%	5119 (89%)	85%	2940
9	-	k-NN, 9	Bagging, 3	98%	5154 (89%)	85%	10527
10	PCA, 0	ANN, 20-20-20	Bagging, 5	99%	4995 (87%)	84%	4655
11	-	k-NN, 9	Boosting, 5	99%	5068 (88%)	84%	15524
12	PCA, 0	k-NN, 9	Boosting, 5	99%	5082 (88%)	84%	2848
13	-	k-NN, 9	Bagging, 5	98%	5116 (89%)	84%	40873
14	-	k-NN, 9	-	99%	5065 (88%)	84%	3278
15	PCA, 0	k-NN, 9	Boosting, 3	99%	5034 (87%)	84%	1895
16	PCA, 0	k-NN, 9	-	99%	5047 (87%)	84%	603
17	-	k-NN, 9	Boosting, 3	99%	5061 (88%)	84%	55712
18	PCA, 0	k-ANN, 9	Bagging, 3	98%	5079 (88%)	84%	1812
19	-	k-ANN, 3	Bagging, 5	99%	5015 (87%)	84%	14456
20	PCA, 0	k-NN, 3	Bagging, 3	99%	4990 (86%)	83%	1764
21	PCA, 0	k-NN, 3	Bagging, 5	99%	4948 (86%)	83%	2867
22	-	k-NN, 3	Bagging, 3	99%	4983 (86%)	83%	10074
23	PCA, 0	k-NN, 3	-	100%	4943 (86%)	83%	569
24	PCA, 0	ANN, 10-10-10	Bagging, 5	99%	4923 (85%)	83%	1220
25	-	k-NN, 3	-	100%	4934 (85%)	83%	3273

A.3 Multiple Exercise Segmentation

Table A.3: Results for test set 3, where the input vector consist of q and $\dot{q}$, $n_{exp} = 12$ and $n_{stack} = 15$, and the exercises examined were 5 lower body rehabilitation exercises. The F_1 and accuracy scores for training and testing for each combination is presented. The time taken accounts for both training and testing time.

	Classifier parameter			**Training**	**Testing (Total correct p_1 = 35819)**		
Rank	**Dim Reduct**	**Classifier**	**Aggregator**	F_1	**Correct**	F_1	**Time taken [s]**
1	PCA, 0	k-NN, 9	-	95%	29773 (83%)	76%	1343
2	PCA, 0	k-NN, 9	Boosting, 5	95%	30020 (84%)	76%	5194
3	PCA, 0	SVM, radial	-	91%	30881 (86%)	76%	275
4	PCA, 0	k-NN, 9	Boosting, 3	95%	29851 (83%)	76%	3910
5	PCA, 0	k-NN, 9	Bagging, 3	93%	30265 (85%)	76%	3694
6	-	k-NN, 3	Bagging, 3	95%	28998 (81%)	76%	32245
7	-	k-NN, 9	Bagging, 3	93%	29905 (84%)	76%	27157
8	PCA, 0	SVM, radial	Bagging, 3	91%	30941 (86%)	76%	403
9	-	k-NN, 3	-	97%	28616 (80%)	75%	10764
10	PCA, 0	k-NN, 9	Bagging, 5	93%	30205 (84%)	75%	5218
11	None	k-NN, 9	Boosting, 3	95%	29137 (81%)	75%	38526
12	PCA, 0	SVM, radial	Boosting, 5	91%	31004 (87%)	75%	568
13	PCA, 0	SVM, radial	Bagging, 5	90%	30812 (86%)	75%	659
14	PCA, 0	SVM, radial	Boosting, 3	91%	30853 (86%)	75%	478
15	None	k-NN, 9	None	94%	29786 (83%)	75%	12814
16	PCA, 0	k-NN, 3	Bagging, 3	96%	28560 (80%)	75%	3714
17	PCA, 0	k-NN, 3	Bagging, 5	96%	28376 (79%)	75%	5189
18	PCA, 0	ANN, 20-20-20	Bagging, 5	94%	28291 (79%)	75%	7887
19	PCA, 0	k-NN, 3	Boosting, 5	98%	28454 (79%)	75%	5756
20	PCA, 0	k-NN, 3	None	98%	28461 (80%)	74%	1417
21	PCA, 0	k-NN, 3	Boosting, 3	98%	28305 (79%)	74%	3825
22	None	k-NN, 3	Boosting, 3	98%	28171 (79%)	74%	39400
23	PCA, 0	ANN, 10-10	Bagging, 5	93%	28695 (80%)	74%	2060
24	PCA, 0	ANN, 10-10-10	Bagging, 3	92%	29740 (83%)	73%	1722
25	PCA, 0	ANN, 10-10-10	Bagging, 5	94%	29589 (83%)	73%	2986

References

[1] A. Ataya, P. Jallon, P. Bianchi, and M. Doron. Improving activity recognition using temporal coherence. In *Proc Ann Int Conf IEEE Eng Med Biol Soc*, 2013.

[2] Mobolaji Ayoade and Lynne Baillie. A novel knee rehabilitation system for the home. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 2521–2530. ACM, 2014.

[3] S. Baby and V. Krüger. Primitive based action representation and recognition. In *Image Analysis*, volume 5575, pages 31–40. Springer, 2009.

[4] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. Online handwriting recognition with support vector machines-a kernel approach. In *Frontiers in handwriting recognition, 2002. proceedings. eighth international workshop on*, pages 49–54. IEEE, 2002.

[5] Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoon Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recogn*, 44:1357–1371, 2011.

[6] Ari Y Benbasat and Joseph A Paradiso. An inertial measurement framework for gesture recognition and applications. In *Gesture and Sign Language in Human-Computer Interaction*, pages 9–20. Springer, 2002.

[7] Katherine Berg, Sharon Wood-Dauphinee, and JI Williams. The balance scale: reliability assessment with elderly residents and patients with an acute stroke. *Scandinavian journal of rehabilitation medicine*, 27(1):27–36, 1995.

[8] Jeroen HM Bergmann, Ruth E Mayagoitia, and Ian CH Smith. A portable system for collecting anatomical joint angles during stair ascent: a comparison with an optical tracking device. *Dynamic Medicine*, 8(1):3, 2009.

[9] JHM Bergmann and AH McGregor. Body-worn sensor design: what do patients and clinicians want? *Annals of biomedical engineering*, 39(9):2299–2312, 2011.

[10] Aude G Billard, Sylvain Calinon, and Florent Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5):370–384, 2006.

[11] Vincent Bonnet, Claudia Mazzà, John McCamley, and Aurelio Cappozzo. Use of weighted fourier linear combiner filters to estimate lower trunk 3d orientation from gyroscope sensors data. *Journal of neuroengineering and rehabilitation*, 10(1):29, 2013.

[12] J. Boyd and H. Sundaram. A framework to detect and classify activity transitions in low-power applications. In *Proc IEEE Int Conf Multimedia Expo*, pages 1712–1715, 2009.

[13] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[14] Jaap J Brunnekreef, Caro JT van Uden, Steven van Moorsel, and Jan GM Kooloos. Reliability of videotaped observational gait analysis in patients with orthopedic impairments. *BMC musculoskeletal disorders*, 6(1):17, 2005.

[15] Herman Bruyninckx and Joris De Schutter. Symbolic differentiation of the velocity mapping for a serial kinematic chain. *Mechanism and machine theory*, 31(2):135–148, 1996.

[16] Horst Bunke, Markus Roth, and Ernst Günter Schukat-Talamazzini. Off-line cursive handwriting recognition using hidden markov models. *Pattern recognition*, 28(9):1399–1413, 1995.

[17] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov*, 2:121–167, 1998.

[18] Adrian Burns, Barry R Greene, Michael J McGrath, Terrance J O'Shea, Benjamin Kuris, Steven M Ayer, Florin Stroiescu, and Victor Cionca. Shimmer–a wireless sensor platform for noninvasive biomedical research. *Sensors Journal, IEEE*, 10(9):1527–1534, 2010.

[19] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, 2007.

[20] Timothy I Carter, Brian Pansy, Aviva L Wolff, Howard J Hillstrom, Sherry I Backus, Mark Lenhoff, and Scott W Wolfe. Accuracy and reliability of three different techniques for manual goniometry for wrist motion: a cadaveric study. *The Journal of hand surgery*, 34(8):1422–1428, 2009.

[21] Mahmoud El-Gohary and James McNames. Shoulder and elbow joint angle tracking with inertial sensors. *Biomedical Engineering, IEEE Transactions on*, 59(9):2635–2641, 2012.

[22] Canadian Institute for Health Information. Hip and knee replacements in canada: Canadian joint replacement registry 2013 annual report. 2013.

[23] Cecille Freeman, Dana Kulić, and Otman Basir. An evaluation of classifier-specific filter measure performance for feature selection. *Pattern Recognition*, 48(5):1812–1826, 2015.

[24] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[25] Andrej Gams, Auke J Ijspeert, Stefan Schaal, and Jadran Lenarčič. On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Autonomous robots*, 27(1):3–23, 2009.

[26] ME Harrington, AB Zavatsky, SEM Lawson, Z Yuan, and TN Theologis. Prediction of the hip joint centre in adults, children, and patients with cerebral palsy based on magnetic resonance imaging. *Journal of biomechanics*, 40(3):595–602, 2007.

[27] Jun-Da Huang. Kinerehab: a kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 319–320. ACM, 2011.

[28] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[29] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese. On the representation, learning and transfer of spatio-temporal movement characteristics. *Int J Hum Robot*, 1:613–636, 2004.

[30] Anil K Jain, Robert PW Duin, and Jianchang Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

[31] Anil K Jain, Jianchang Mao, and KM Mohiuddin. Artificial neural networks: A tutorial. *Computer*, (3):31–44, 1996.

[32] Janet Jones, M Des, Kathleen Norman, and Spencer Saunders. Trends and drivers of change in physiotherapy in ontario in 2014.

[33] Vladimir Joukov, Vincent Bonnet, Michelle Karg, Gentiane Venture, and Dana Kulic. Rhythmic ekf for pose estimation during gait. In *International Conference on Humanoid Robots, 2015 IEEE-RAS*. IEEE, 2015.

[34] Vladimir Joukov, Michelle Karg, and Dana Kulic. Online tracking of the lower body joint angles using imus for gait rehabilitation. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 2310–2313. IEEE, 2014.

[35] Nathan Koenig and Maja J Mataric. Behavior-based segmentation of demonstrated tasks. In *Proceedings of the International Conference on Development and Learning*. Citeseer, 2006.

[36] N. C. Krishnan, D. Colbry, C. Juillard, and S. Panchanathan. Realtime human activity recognition using tri-axial accelerometers. In *Sensors, Signals and Information Processing Workshop*, 2008.

[37] D. Kulić, W. Takano, and Y. Nakamura. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Trans Robot*, 25:1158–1166, 2009.

[38] Dana Kulić and Yoshihiko Nakamura. Incremental learning of human behaviors using hierarchical hidden markov models. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4649–4655. IEEE, 2010.

[39] Belinda Lange, Chien-Yen Chang, Evan Suma, Bradley Newman, Albert Skip Rizzo, and Mark Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1831–1834. IEEE, 2011.

[40] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.

[41] Carlos Lavernia, Michele D'Apuzzo, Mark D Rossi, and David Lee. Accuracy of knee range of motion assessment after total knee arthroplasty. *The Journal of arthroplasty*, 23(6):85–91, 2008.

[42] J. F. S. Lin and D. Kulić. On-line segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Trans Neural Syst Rehabil Eng*, (in print).

[43] Jonathan Feng-Shun Lin, Vladimir Joukov, and Dana Kulic. Human motion segmentation by data point classification. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 9–13. IEEE, 2014.

[44] Jonathan FS Lin and Dana Kulić. Human pose recovery using wireless inertial measurement units. *Physiol. Meas.*, 33(12):2099, 2012.

[45] Zhu Liu, Yao Wang, and Tsuhan Chen. Audio feature extraction and analysis for scene segmentation and classification. *J VLSI Signal Process Syst Signal Image Video Technol*, 20:61–79, 1998.

[46] Kay Soon Low, Guo Xiong Lee, and Tanim Taher. A wearable wireless sensor network for human limbs monitoring. In *Instrumentation and Measurement Technology Conference, 2009. I2MTC'09. IEEE*, pages 1332–1336. IEEE, 2009.

[47] L. Lu, H. J. Zhang, and H. Jiang. Content analysis for audio classification and segmentation. *IEEE Speech Audio Process*, 10:504–516, 2002.

[48] Lie Lu, Stan Z Li, and Hong-Jiang Zhang. Content-based audio segmentation using support vector machines. In *Proc ICME*, volume 1, pages 749–752, 2001.

[49] F. Lv and R. Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Proc Euro Conf Comp Vision*, pages 359–372, 2006.

[50] Benoit Mariani, Hossein Rouhani, Xavier Crevoisier, and Kamiar Aminian. Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors. *Gait & posture*, 37(2):229–234, 2013.

[51] Jun Nakanishi, Jun Morimoto, Gen Endo, Gordon Cheng, Stefan Schaal, and Mitsuo Kawato. A framework for learning biped locomotion with dynamical movement primitives. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 2, pages 925–940. IEEE, 2004.

[52] Cynthia C Norkin and D Joyce White. *Measurement of joint motion: a guide to goniometry*. FA Davis, 2009.

[53] Stepan Obdrzalek, Gregorij Kurillo, Ferda Ofli, Ruzena Bajcsy, Edmund Seto, Holly Jimison, and Misha Pavel. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *Engineering in medicine and biology society (EMBC), 2012 annual international conference of the IEEE*, pages 1188–1193. IEEE, 2012.

[54] Federation of Canadian Municipalities. Canadas aging population: The municipal role in canadas demographic shift. 2013.

[55] Marco Parvis and Franco Ferraris. Procedure for effortless in-field calibration of three-axial rate gyro and accelerometers. *Sensors and Materials*, 7(5):311–330, 1995.

[56] Tadej Petrič, Andrej Gams, Auke Jan Ijspeert, and Leon Žlajpah. On-line frequency adaptation and movement imitation for rhythmic robotic tasks. *The International Journal of Robotics Research*, 30(14):1775–1788, 2011.

[57] Réjean Plamondon and Sargur N Srihari. Online and off-line handwriting recognition: a comprehensive survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):63–84, 2000.

[58] Alexander Rampp, Jens Barth, Samuel Schulein, Karl-Gunter Gasmann, Jochen Klucken, and Bjorn M Eskofier. Inertial sensor-based stride parameter calculation from gait sequences in geriatric patients. *Biomedical Engineering, IEEE Transactions on*, 62(4):1089–1097, 2015.

[59] Ludovic Righetti, Jonas Buchli, and Auke Jan Ijspeert. Dynamic hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena*, 216(2):269–281, 2006.

[60] Ali-Akbar Samadani and Dana Kulic. Hand gesture recognition based on surface electromyography. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 4196–4199. IEEE, 2014.

[61] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In *Robotics Research. The Eleventh International Symposium*, pages 561–572. Springer, 2005.

[62] Loren Arthur Schwarz, Diana Mateus, and Nassir Navab. Discriminative human full-body pose estimation from wearable inertial sensor data. In *Modelling the Physiological Human*, pages 159–172. Springer, 2009.

[63] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Resampling or reweighting: A comparison of boosting implementations. In *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, volume 1, pages 445–451. IEEE, 2008.

[64] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[65] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[66] E. H. Spriggs, F. De la Torre Frade, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *IEEE Workshop Egocentric Vision*, 2009.

[67] David Tedaldi, Alberto Pretto, and Emanuele Menegatti. A robust and easy to implement method for imu calibration without external equipments. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3042–3049. IEEE, 2014.

[68] Aurore Thibaut, Camille Chatelle, Erik Ziegler, Marie-Aurélie Bruno, Steven Laureys, and Olivia Gosseries. Spasticity after stroke: physiology, assessment and treatment. *Brain Injury*, 27(10):1093–1105, 2013.

[69] B Toro, C Nester, and P Farren. A review of observational gait assessment in clinical practice. *Physiotherapy Theory and Practice*, 19(3):137–149, 2003.

[70] Laurens JP van der Maaten, Eric O Postma, and H Jaap van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[71] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[72] Philip A Wolf, Ralph B D'Agostino, Albert J Belanger, and William B Kannel. Probability of stroke: a risk profile from the framingham study. *Stroke*, 22(3):312–318, 1991.

[73] Chou LS Lin JH Chang CC Xu X, McGorry RW. Accuracy of the microsoft kinect for measuring gait parameters during treadmill walking. *Gait and Posture*, 42(2):145–51, 2015.

[74] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *Int J Tech Decis*, 05(04):597–604, 2006.

[75] Hong Ying, Carmen Silex, Andreas Schnitzer, Steffen Leonhardt, and Michael Schiek. Automatic step detection in the accelerometer signal. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pages 80–85. Springer, 2007.

[76] L. Zhao and G. Sukthankar. An active learning approach for segmenting human activity datasets. In *Proc ACM Conf Multimedia*, pages 765–768, 2009.

[77] Huiyu Zhou and Huosheng Hu. Human motion tracking for rehabilitationa survey. *Biomedical Signal Processing and Control*, 3(1):1–18, 2008.

[78] Huiyu Zhou, Thomas Stone, Huosheng Hu, and Nigel Harris. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical engineering & physics*, 30(1):123–133, 2008.