Automated Rehabilitation Exercise Motion Tracking

by

Jonathan Feng-Shun Lin

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, 2012

I hereby declare that I am the sole author of this 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.

Abstract

Current physiotherapy practice relies on visual observation of the patient for diagnosis and assessment. The assessment process can potentially be automated to improve accuracy and reliability. This thesis proposes a method to recover patient joint angles and automatically extract movement profiles utilizing small and lightweight body-worn sensors.

Joint angles are estimated from sensor measurements via the extended Kalman filter (EKF). Constant-acceleration kinematics is employed as the state evolution model. The forward kinematics of the body is utilized as the measurement model. The state and measurement models are used to estimate the position, velocity and acceleration of each joint, updated based on the sensor inputs from inertial measurement units (IMUs). Additional joint limit constraints are imposed to reduce drift, and an automated approach is developed for estimating and adapting the process noise during on-line estimation.

Once joint angles are determined, the exercise data is segmented to identify each of the repetitions. This process of identifying when a particular repetition begins and ends allows the physiotherapist to obtain useful metrics such as the number of repetitions performed, or the time required to complete each repetition. A feature-guided hidden Markov model (HMM) based algorithm is developed for performing the segmentation. In a sequence of unlabelled data, motion segment candidates are found by scanning the data for velocity-based features, such as velocity peaks and zero crossings, which match the pre-determined motion templates. These segment potentials are passed into the HMM for template matching. This two-tier approach combines the speed of a velocity feature based approach, which only requires the data to be differentiated, with the accuracy of the more computationally-heavy HMM, allowing for fast and accurate segmentation.

The proposed algorithms were verified experimentally on a dataset consisting of 20 healthy subjects performing rehabilitation exercises. The movement data was collected by IMUs strapped onto the hip, thigh and calf. The joint angle estimation system achieves an overall average RMS error of 4.27 cm, when compared against motion capture data. The segmentation algorithm reports 78% accuracy when the template training data comes from the same participant, and 74% for a generic template.

Acknowledgments

Research is rewarding but often difficult. As such, the completion of this thesis was made possible only through the contributions of various people and organizations.

I would like to thank my supervisor Dr. Dana Kulić for her support, guidance, and expertise throughout the duration of this project. I am truly grateful for the advices and suggestions given in our various meetings, and for all the hours spent reviewing my writing. I am also indebted to her for funding my travels abroad where I was able to gain exposure to the international biomedical engineering community.

I would like to thank Cardon Rehabilitation and the University of Waterloo (Iron Ring Graduate Scholarship) for providing the funding that made this research possible.

Special thanks go to my research participants, for taking out time out of their busy schedules to provide my study with the data needed to verify my algorithms. I would also like to extend my appreciation to Dr. Mitchell Fergenbaum and Joanna Wong for providing key insight into physiotherapy exercises and needs of the field. The completion of my thesis was only made possible by the support of each and every single one of you.

I would like to thank the other students of the Kulić lab for their continual ideas, support, and the extra pair of hands to start the motion capture system. I would like to acknowledge Dr. Martin Schwartz [92] and Dr. Jorge Cham [18] for their respective work, for reminding the members of academia that research is difficult for everyone, and that the more comfortable we are with knowing what we do not know, the deeper we will be able to tinker with the unknown.

Most importantly, I would like to thank my family and friends, from both inside and outside of the University of Waterloo, from the UWCCF, Grad Cell, and elsewhere, for their unwavering support and encouragement throughout my many years at Waterloo. This thesis would not have been possible without you.

In his heart a man plans his course, but the Lord determines his steps. - *Proverbs 16.9*

Dedication

This thesis is dedicated to my parents, James and Shirley, and my brother, Jackson.

Each encouragement, support, and lesson did much to get me to where I am today.
Thank you for every decision and sacrifice made, for always thinking of me.

Table of Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

AV	Aggregated Velocity
DH	Denavit-Hartenberg convention
DOF	Degree of Freedom
DP	Dynamic Programming
DTW	Dynamic Time Warping
EKF	Extended Kalman Filter
HMM	Hidden Markov Model
IMU	Inertial Measurement Units
KF	Kalman Filter
LL	Log-Likelihood
MARG	Magnetic, Angular Rate, and Gravitational sensor
MEMS	Micro-Electro-Mechanical devices
RMS	Root-Mean-Square
TJR	Total Joint Replacement
ZVC	Zero-Velocity Crossing

Variables

A	Matrix that relate the state at time $t-1$ to the state at time t. For the EKF, an linearization can be approximated by $\partial f/\partial \boldsymbol{s_t}$
A_H	State transition probability matrix
B_H	Observation probability matrix
C	Matrix the relate the observation to the state. For the EKF, an linearization can be approximated by $\partial h/\partial \boldsymbol{s_t}$
K_t	Kalman gain at time t
LL_{thres}	Log-likelihood threshold for HMM templates
P_t	*A posteriori* error covariance matrix at time t
P_t^-	*A priori* error covariance matrix at time t
P_y	Intermediate noise covariance matrix used to calculate K_t for the EKF
$R_{i-1,i}$	Rotational matrix from the $i-1^{th}$ frame to the i^{th} frame
$T_{i-1,i}$	Transformation matrix from the $i-1^{th}$ frame to the i^{th} frame
T_{tt}	Tool transform matrix
V	Jacobian matrix of partial derivatives of the measurement function h with respects to the measurement noise v. Calculated by $\partial h/\partial \boldsymbol{v_t}$
W	Jacobian matrix of partial derivatives of the state function f with respects to the state noise w. Calculated by $\partial f/\partial \boldsymbol{w_{t-1}}$
a_i	Distance between the z_{i-1} and the z_i axes, along the x_i axis
d_i	Distance from frame origin $i-1$ to the x_i axis, along the z_{i-1} axis
f	State function
g	Gravity vector
h	Measurement function
$\boldsymbol{q_c}$	Joint limit. The upper limit is $\underline{\boldsymbol{q}}_{\boldsymbol{c}}$ and the lower limit is $\overline{\boldsymbol{q}}_{\boldsymbol{c}}$
q_i	Local angle vector of the current frame. It is vectorized as $\boldsymbol{q_i} = [0; 0; q_i]$
$\dot{q}_i$	Local angular velocity vector of the current frame. It is vectorized as $\dot{\boldsymbol{q}}_{\boldsymbol{i}} = [0; 0; \dot{q}_i]$
$\ddot{q}_i$	Local angular acceleration vector of the i^{th} frame. It is vectorized as $\ddot{\boldsymbol{q}}_{\boldsymbol{i}} = [0; 0; \ddot{q}_i]$
$\boldsymbol{r_i}$	Joint length vector of joint i
$\boldsymbol{r_{lShim}}$	Offset vector from the calf to the calf sensor
$\boldsymbol{r_{uLen}}$	Joint length vector from hip to knee
$\boldsymbol{r_{uShim}}$	Offset vector from thigh to thigh sensor
$\boldsymbol{s_t}$	State at time t
$\hat{\boldsymbol{s}}_{\boldsymbol{t}}$	*A posteriori* estimate of $\boldsymbol{s_t}$ at time t
$\hat{\boldsymbol{s}}_{\boldsymbol{t}}^-$	*A priori* estimate of $\boldsymbol{s_t}$ at time t

$\tilde{\boldsymbol{s}}_{\boldsymbol{t}}$	Noise-less estimate of s_t at time t
t	Time
Δt	Timestep
t_{pp}	Peak-to-peak time threshold
$\boldsymbol{v_t}$	Observation measurement noise at time t. A normally distributed random variable, this variable can be characterized as $p(v) \sim N(0, Q_t)$, where Q_t is the measurement noise covariance
v_p	Velocity magnitude threshold
$\boldsymbol{w_t}$	State process noise at time t. A normally distributed random variable, this variable can be characterized as $p(w) \sim N(0, R_t)$, where R_t is the process noise covariance
$\boldsymbol{x_i^i}$	Position vector in the i^{th} frame
$\dot{\boldsymbol{x}}_{\boldsymbol{i}}$	Linear velocity vector of the i^{th} frame
$\ddot{\boldsymbol{x}}_{\boldsymbol{i,e}}$	Linear acceleration vector of the end effector at the i^{th} frame
$\ddot{\boldsymbol{x}}_{\boldsymbol{i,i}}$	Linear acceleration vector from the i^{th} frame
$\boldsymbol{z_t}$	Observation at time t
$\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$	Noise-less estimate of z_t at time t
$\boldsymbol{\Sigma_H}$	State standard deviations
α_i	Angle from z_{i-1} to the z_i axis, about the x_i axis
$\boldsymbol{\alpha_i^i}$	Total angular acceleration vector of the i^{th} frame
$\epsilon_{divThres}$	Condition number threshold for P_y
η	Potential field coefficient
γ_i	Potential field acceleration
λ	HMM template variables. $\lambda = (A_H, B_H, \boldsymbol{\pi_H})$
$\boldsymbol{\mu_H}$	State means
$\boldsymbol{\pi_H}$	initial state distribution, the prior
ρ_i	Distance from the potential field limit for joint angle i. The upper bound distance is $\underline{\rho}_i$, whereas the lower bound distance is $\overline{\rho}_i$
$\rho_{0,i}$	Distance limits of the potential field influence for joint angle i. The upper bound limit is $\overline{\rho}_{0,i}$, whereas the lower bound limit is $\underline{\rho}_{0,i}$
θ_i	angle between the x_{i-1} and the x_i axes, about the z_{i-1} axis
$\boldsymbol{\omega_i^i}$	total angular velocity vector of the i^{th} frame

Chapter 1

Introduction

Physiotherapy, a component of modern healthcare, is concerned with the development, maintenance and restoration of body movement and functionalities after illness or injury. Through a regimen of physiotherapist-prescribed exercises, strength, range of motion and balance are restored, to the greatest possible extent.

Long wait-lines are a major concern in Canadian physiotherapy. A survey of 3000 Ontario family physicians in 2004 found that physicians identified long wait times as one of the most common barriers to referring patients to rehabilitation [21].

Over the next decade, the Canadian demand for total joint replacements (TJR) is predicted to grow annually by 8.7% [58]. TJR is a type of orthopedic operation where both of the surfaces of the joint pair are removed and replaced by artificial ones. These surgeries are usually performed on elderly patients who have suffered major falls and damaged their joints, or experienced joint degradation due to osteoarthritis or rheumatoid arthritis [16]. Post-TJR patients typically require 1-2 weeks of physiotherapy to recover from the operation.

Given the expected increase in demand for services, wait-lines for physiotherapy constitute an increasingly growing concern. Technological innovation and policy attention are needed to address this field. One potential approach for improving services is through improved measurement and assessment via the use of human motion measurement tools.

During the rehabilitation process, the therapist typically initiates each session by performing an assessment of the patient's current status, using visual observation of the patient's movement as well as questionnaires, such as the Community Balance and Mobility Scale [40] and the Falls Efficacy Scale [108]. Following assessment, movement exercises are prescribed by a physiotherapist to help improve or recover muscle strength, endurance and range of motion. To ensure that such exercises are performed correctly and safely, the physiotherapist typically observes the patients while they perform the exercises. However, with the exception of goniometry, most of the exercise assessment tools available to the physiotherapist tend to be subjective in nature. Goniometry [75] is a human joint angle measurement technique that isolates a single body joint in order to assess a subject's range of motion. However, goniometrics cannot be employed accurately when the subject is moving, thus reducing its usefulness during exercise or functional rehabilitation.

Research from kinesiology suggests that, when participants are requested to reproduce a specific motion, they tend to focus on the motion of the end-point in Cartesian space (the end-effector) and not on joint trajectory, potentially leading to improper form [29]. For example, hip replacement patients asked to perform hip abduction while standing may focus only on their foot position and bend their torso sideways to achieve the requested foot position, leading to improper form. These incorrect exercises, often too subtle to be easily noticeable, can lead to additional injury. An electronic measurement system could catch these joint angle errors and report them to the physiotherapists.

Thus, the development of a rehabilitation exercise monitoring and assessment system can serve to improve the rehabilitation process and reduce wait-lines by freeing up physiotherapists during exercise supervision, providing more detailed joint angle measurements over a wider range of situations, and automatically tracking the patient's progress over each rehabilitation session.

1.1 Thesis Contributions

This thesis proposes a method to measure a patent's joint angles and identify each exercise repetition during a physiotherapy session via lightweight sensors. An overview of the system components is shown in Figure 1.1.

On-line pose estimation from body-worn sensors. Joint angles are computed using state estimation tools and articulated rigid body chain modeling. The proposed method is an improvement compared to goniometrics as it allows joint angle estimation throughout the patient's exercise routine and not only during static postures.

On-line exercise segmentation and identification. Relevant motion segments are isolated and identified via multi-tier pattern matching and data analysis algorithms. Segmented motion sequences provide additional information, such as the number of

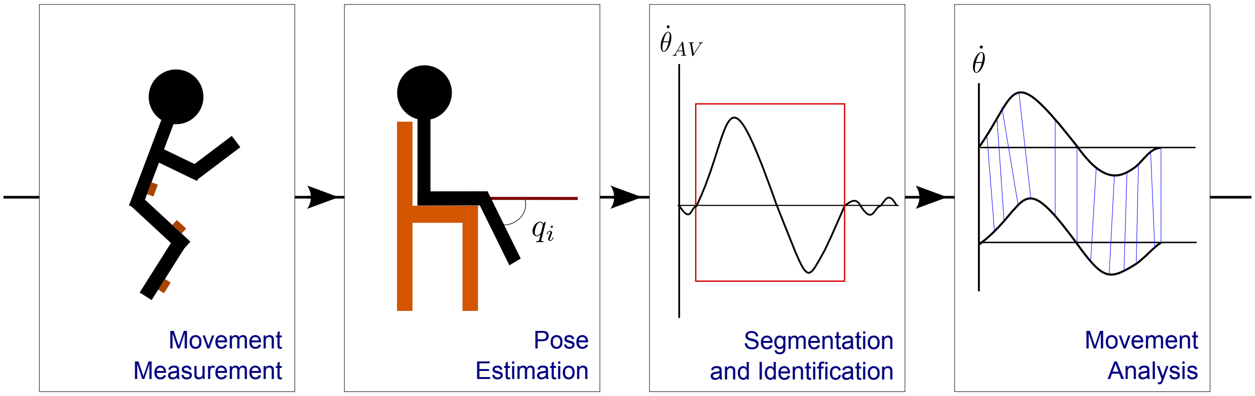


Figure 1.1: Components of an automatic rehabilitation supervision system. This thesis proposes algorithms for pose estimation and segmentation.

repetitions of a given exercise performed, and the time taken to complete these repetitions. This information allows physiotherapists to assess the patient's performance quantitatively, to assess the patient's progress and the effectiveness of treatment.

Experimental validation. The proposed system was verified on rehabilitation exercise data from 20 healthy subjects. Five rehabilitation exercises, typically assigned during TRJ rehabilitation, were performed by each subject: leg extension while seated, sit-to-stand, squats, knee/hip flexion while supine and straight leg raise while supine. Both the pose estimation and segmentation and identification algorithms were verified on the collected dataset.

1.2 Thesis Outline

Chapter 2 provides an overview of existing research into automated rehabilitation and related techniques. First, an overall survey of automated rehabilitation and monitoring systems is provided. Specific fields examined are robotics-based rehabilitation, telerehabilitation and wearable solutions. Secondly, various joint angle estimation algorithms that employ inertial measurement units are examined. These algorithms include algorithms that rely on accelerometers for incline, algorithms that utilize gyroscope for when a participant is moving rapidly, and algorithms that employ magnetometers to determine orientation. Lastly, existing work on motion segmentation and identification is considered. The primary algorithms examined are zero-velocity crossing, dynamic time warping and the hidden Markov model.

Chapter 3 provides conceptual and algorithmic background information on the mathematical tools employed in this thesis. A robotic modeling framework, the Denavit-Hartenberg convention, is introduced. Using this convention, a sequence of articulated rigid bodies can be modeled with forward kinematics. A general overview on data collection via inertial measurement units, as well as data processing via Kalman filter and the extended Kalman filter, is given. Segmentation techniques such as zero-velocity crossings and the hidden Markov model are also introduced.

Chapter 4 details the proposed joint angle estimation algorithm. The Denavit-Hartenberg convention and forward kinematics are utilized in the extended Kalman filter as the observation equations. Noise covariance adaption and potential fields are applied to the extended Kalman filter to increase the robustness of the joint estimation algorithm and reduce drift, especially in the joints that do not experience large changes. Sample results are provided as examples of the algorithm in action.

Chapter 5 details the proposed segmentation and identification algorithm. During the training phase, the algorithm requires a series of exemplar motions, extracts key velocity features and generates a hidden Markov model for each template. During the on-line execution phase, a sliding window searches the observation data for velocity features that match the template and identifies them as possible segment potentials. These segment potentials are verified by the hidden Markov model for closeness of match. Sample results are provided as examples of the algorithm in action.

Chapter 6 describes the verification experiments. A dataset consisting of 20 healthy subjects performing 5 rehabilitation exercises was collected, using both inertial measurement units and motion capture. A detailed analysis of the performance of both the pose estimation and the segmentation algorithms is performed, together with a comparison to competing algorithms.

Chapter 7 summarizes the findings and conclusions of this research and outlines directions for future work.

Chapter 2

Related Work

Many groups have proposed physiotherapy assistive systems in an attempt to address long wait-lines, as well as provide physiotherapists with additional patient data. This chapter reviews common approaches to rehabilitation technology, including robotics, telerehabilitation, home-based solutions and body-worn sensors. Research groups have proposed a wide variety of grip-based and exoskeleton robotics for rehabilitation. Telepresence solutions, which allow patients to consult physiotherapists and perform supervised rehabilitation at a distance, have also been proposed. Different wearable sensing solutions have been devised, to determine patient posture and joint angles. Lastly, motion segmentation and identification algorithms that crop and label observed data are reviewed.

2.1 Automated Rehabilitation

2.1.1 Rehabilitation Robotics

One approach for automating the rehabilitation process is the use of robotics [76]. Garcia *et al.* [32] provides an overview of robotics used in rehabilitation. Rehabilitation robotics have been used in a wide range of applications, such as artificial limbs, physiotherapist assistants, or providing personal care for patients in hospitals and residential sites. However, many assistive robotic systems tended to use industrial robot arms for economic and availability reasons. Industrial robots, designed for speed, accuracy, rigidness and power, are not suitable for human interactions. Within a factory setting, no human operators are allowed near the robots while they are in operation. Rehabilitation robots must incorporate some degree of compliance control to minimize the risk of injury to patients.

Surveys [66] of physiotherapists and occupational therapists from Australia, Canada and the United States ($n = 233$) revealed that, for a rehabilitation robot to be useful, it must be able to facilitate a variety of movements, be usable while the subject is seated, provide biofeedback to patients, be able to facilitate activities specific to daily living, be useful inside the home, have adjustable resistances to meet clinical needs, and cost under $6000 USD. A comprehensive robotic system that achieves all the stated criteria does not appear to currently exist, or is not well known, as only 6% of the medical providers surveyed have used robot-facilitated rehabilitation.

Despite the difficulties, many prototype systems have been developed. An early attempt to develop rehabilitation robotics examined utilizing continuous passive motion (CPM), where an actuated brace moves the injured joint through a single degree of freedom (DOF) motion, similar to how a physiotherapist would manually stimulate leg motion [89]. However, because CPM devices only actuate in one DOF, they do not exercise certain muscle groups, and do not account for patient motion, thus potentially causing injury [86].

The MIT-MANUS [56] are a series of modular planar haptic robotic arms that have been applied to stroke rehabilitation with success. These robotic arms were designed to address the safety issues posed by industrial robots by including backdrivability and compliance, and target upper-body rehabilitation by guiding the participants in a series of goal-directed tasks. If the user of the robot was not able to perform these tasks at the expected speed, the robot applied an assistive force that is scaled according to how close the user's movements are to the desired trajectory [13]. Clinical trials show that patients that received robot-facilitated rehabilitation performed comparably or better than control groups when assessed by functional metrics.

Certain exercises, such as stroke recovery hand exercises, are difficult for a single physiotherapist to perform, since the patients tend to clinch their hands in a fist-like posture. Systems such as the Hand Exoskeleton Rehabilitation Robot (HEXORR) [91] can help the patients perform hand exercises, leaving the physiotherapist free to perform other tasks. Patients with weakness in lower limbs can utilize devices such as the ZeroG [37] to alleviate a portion of the patient's body weight, by using elastic straps and actuators. This safeguards them from falling. The goals of such devices are to assist the therapist so that they may safely train patients in standing, walking, and performing balance activities early after injuries. These tasks are difficult for therapists alone, but are facilitated with robotic support. However, even with such systems already developed, clinician attendance for conferences relating to rehabilitation robotics tends to be poor, suggesting poor exposure. Clinicians cite the following reasons why robots are not currently being used: financial cost, long set up time, and the system is too complicated for the patient or physiotherapist to run without engineering help [38].

Cyberdyne[1] developed the Hybrid Assistive Limb (HAL) exoskeleton [90], designed to physically support the user's rehabilitation movements. The HAL detects muscle biosignals to estimate the user's intended torque, and supports the motions correspondingly. When biosignals are not reaching the lower-body due to spinal injuries, the HAL can also check joint angles, centre of gravity and ground reaction forces to predict the user's conditions and assist when possible. The HAL has been tested with a spinal cord injury patient and has shown that the patient was able to walk at a normal pace.

Other rehabilitation or assistive exoskeletons have also been developed or are commercially available [70]. Ekso Bionics[2] developed the Ekso exoskeleton walker [94] for paraplegic users. Farris *et al.* [26] developed the gait-assist Vanderbilt exoskeleton for spinal cord injury patients. The University of Washington UL-EXO7 7 DOF exoskeleton arms [78] have been developed and tested in a clinical trial on stroke rehabilitation [52] and showed comparable results in joint angle range recovery to typical rehabilitation techniques.

However, there are still outstanding questions involving rehabilitation robotics and exoskeletons. Although the desired patient range of motion can be achieved with the robots, the motion performed tends to be passive and not active. With the robot exerting the bulk of the energy required to perform the motion, the patients do not exert themselves and thus are not benefiting from the rehabilitation process to the fullest extent. This is because it is difficult to automatically determine the extent of assistance the robot should provide during rehabilitation. It is also difficult to determine if the robot-facilitated motions generalize to all functional movements required for daily living.

2.1.2 Telerehabilitation

Telerehabilitation is the process of delivering rehabilitation over the telephone or other communicative channels, allowing rehabilitation to be delivered to patients who cannot travel to a physiotherapy clinic for in-person sessions due to severity of disease or disability, or due to distance and travel time.

Tousignant *et al.* [95] developed an Internet-based videoconferencing solution, where patients perform rehabilitation exercises at home with physiotherapist supervision over the Internet, reducing patient travel time and costs. Assessments showed that patient improvements were observed. This proof-of-concept study noted several limitations. In particular, the need for someone to be on-site to ensure that no falls occur. Also, the

[1] Cyberdyne Inc., www.cyberdyne.jp

[2] Ekso Bionics, www.eksobionics.com

exercise activities chosen did not require any additional equipment. The study noted that it can be difficult to assess certain conditions, such as functional autonomy, strength, walking and balance, via cameras. Concerns about maintaining a high-speed streaming videoconferencing connection for over an hour were also cited as some disconnections and pixel loss were noted. A follow-up pilot study [96] with the same setup showed high clinician and participant satisfaction with the system, suggesting that videoconferencing rehabilitation is feasible for total knee replacement patients. However, in both the proof-of-concept and the pilot study, no control groups were used, making it difficult to compare telerehabilitation with in-person clinical visits. In addition, only a small sample population was assessed: $n = 4$ for the proof-of-concept study, and $n = 5$ for the pilot.

A similar system, developed by Russell *et al.* [85], did include a control group ($n = 34$), along with the test group ($n = 31$), and is designed to be usable with low-bandwidth or dial-up Internet connections. The study was carried out in the hospital, with the control group in a physiotherapy clinic and the telerehabilitation group in rooms designed to look like a typical home environment without special rehabilitation equipment. The telerehabilitation group showed no significant difference in improvements using the Western Ontario and McMaster Universities Osteoarthritis Index (WOMAC) when compared to the control group ($p = 0.08$) after six weeks. Both groups showed significant improvement from the baseline. Similar to other cited studies, high levels of satisfaction were reported by the patients. Some complaints were made about the video quality, but it did not seem to impact the rehabilitation effectiveness. A commentary on the article [35] suggests that telerehabilitation will be well received by the orthopedic community and that it will reduce the cost of physiotherapy, offer patients greater convenience, and reduce wait-lines.

Fook *et al.* [31], citing problems with contemporary usability of videoconferencing rehabilitation packages, the necessity of expensive cameras, and automatic assessment of exercise performed, developed a system based on existing videoconferencing tools and additional body-worn sensors. For exercise assessment, they utilized accelerometers, and examined features such as peak detection and root-mean-square (RMS) values. The motion data was presented to the patient and/or physiotherapist in real-time via 3D Studio Max model, allowing the patient to correct improperly performed exercises as soon as the system detects them.

2.1.3 Home-based Rehabilitation

Home-based rehabilitation has also been examined. Baskett *et al.* [7] suggested that patients may not take ownership of their own rehabilitation and may feel that rehabilitation only happens during official therapy sessions. In their study, home rehabilitation was used in an effort to change patient attitudes. Stroke patients were randomly assigned to a rehabilitation clinic as control ($n = 50$) or were given an information session and sent home for home rehabilitation with physiotherapist checkups at regular intervals ($n = 50$). Both methods showed similar recovery rates after three months.

Other groups tried to remove the need for physiotherapist supervision altogether by employing systems that can analyze the various home rehabilitation situations that the patients are in and provide an appropriate feedback. To achieve this aim, Cavallo *et al.* [17] developed the Pervasive Intelligent System for Rehabilitation and Assistance (PISRA). PISRA employed software packages, wearable wireless sensors and robotics to assist the patients and their home-based caregivers. Although the system implemented features such as fall detection and location tracking, it did not perform postural detection and thus cannot be used for direct home rehabilitation exercise supervision.

2.1.4 Wearable Sensors for Rehabilitation

Lymberis [69], Bergmann and McGregor [9] examined the requirements of patients and healthcare providers for wearable sensors by looking at collections of published work. Wearable sensors should be lightweight, compact, and simple to operate and maintain. They should not affect daily behaviour but should provide reliable and constant monitoring of the patient's status, as well as provide feedback to the patient. They should be designed to work along side, not replace, a healthcare professional.

Developing Sensing Packages and Communication Protocols

Various inertial measurement unit (IMU) and wireless sensing packages have been proposed. Van Acht *et al.* [100] introduced the Philips[3] wireless inertial sensor node (PI-node), an IMU magnetic, angular rate and gravitational (MARG) sensor array, which consists of an accelerometer, a gyroscope and a magnetometer. It transmits data wirelessly via IEEE 802.15.4, a wireless protocol designed for a wireless personal area network (PAN).

Burns *et al.* [14] developed the Shimmer[4] wireless sensory system, which transmits accelerometer data over IEEE 802.15.4 and Bluetooth, with options to extend the system to accommodate for a gyroscope, magnetometer or electromyograph (EMG).

[3] Royal Philips Electronics, www.research.philips.com

[4] Shimmer Research, www.shimmer-research.com

Afonso *et al.* [1] concluded that the IEEE 802.15.4 was insufficient for real time applications and suffered from too many packet losses. They developed their own wireless protocol, and implemented it onto CrossBow[5] MICAz [22] transmitters. The system tracked heart rate, respiratory rate, and posture. The paper focused on verifying that their setup was feasible from a power usage and data transmission rate point of view, and was not used to measure patient status. No comparisons were made to existing systems, making it difficult to assess the merits of the system.

Hadjidji *et al.* [36] created a new communication protocol to optimize battery use, also based on the IEEE 802.15.4. They tested their system with 3 Shimmer sensors and showed that it could run for 10 hours per charge. They have also shown that, with 5 nodes transmitting at 400 Hz, they achieved less than 5% of packet loss.

Fergus *et al.* [27] proposed a more immersive solution. In a home setting, patients were given wireless sensors to wear as a PAN. In the home, a larger network was also installed, which the PAN could interface with, and included existing networked devices such as computers. This allowed patients to interact with a wide variety of tools in their own home without having to purchase additional equipment. Exercise motions can then be monitored by the physiotherapist and played back later for the patient.

Using Sensor Packages for Rehabilitation

Salarian *et al.* [88] used IMUs to record the Timed Up and Go (TUG) test, a standard assessment of balance of mobility. The TUG requires a subject to stand up from a chair, walk 3 meters, turn around, and return to the chair. The IMU signals were used to determine how long it takes for the motions to occur, and reported a TUG assessment based on time of walking.

Varkey *et al.* [101] emphasized the distinction between activity recognition, which only determines the type of activity a given subject is performing, against movement recognition, which involves breaking down the motions and analyzing them. They used support vector machines (SVMs) and feature extraction on accelerometers and gyroscopes in order to identify motions. The features examined were mean, standard deviation, maximum, peak-to-peak, RMS, and correlation between accelerometer and gyroscope values. They employed a two-level windowing, where the larger window identified the general activity, and the small window, moving only within the larger window, identifies the motion intervals. For this, they employed Shimmer motes, and achieved over 80% accuracy with exercise motions such as walking, standing and writing.

[5] Crossbow Technology, www.xbow.com

The above works use wireless sensing for gross movements or activity recognition. Previous work focusing on pose recovery and joint angle estimation are reviewed in detail in Section 2.2.

A survey [107] on wireless sensing for clinical use concluded that wearable sensing must be made as simple as possible in order to be utilized by clinicians. This survey identified three major barriers to the widespread acceptance of wearable sensing in the clinic: financial challenges, lack of technical standards and usability guidelines, and lack of education for end-users. The latter two were cited to be especially significant. The lack of standards leads to increased implementation costs for the clinicians. Clinicians sometimes feel overloaded with information, so collected data should automatically be inserted into electronic healthcare records for later review. Many of the current barriers are not related to technology, but to the question of whether patients and clinicians will find the developed systems convenient and easy to use. The development of standards and a tighter integration with existing systems would help with this problem.

Also, measurements generated by current sensors are not part of traditional measures of health. The assessment of angular velocity and force exertion is not currently used in physiotherapy practice and thus technology developed to assess these is not easily applied in current practice. Some efforts need to be diverted into making these measures relevant to the healthcare professional [107].

2.2 Pose Estimation

This section focuses on joint angle recovery methods utilizing lightweight IMU packages that can be mounted non-invasively onto the patient via Velcro straps. Source-less IMUs are favored over camera, acoustic or mechanical based motion detection due to size, cost and environment concerns [110]. The use of ambulatory sensors also enables applications in outdoor settings, such as for sports training. However, using IMUs to obtain an accurate estimate of the joint angles is a challenging problem. Gravity contamination, sensor noise and integrational drift degrade the accuracy of the joint angle estimation, and must be accounted for in any joint angle estimation solution.

2.2.1 Inclinometers

Many of the existing works in joint angle determination utilize accelerometers as inclinometers to measure knee angles. Often, the operational space is simplified, and motions are only examined in a 2D sagittal plane. For slow motions, the acceleration contributed by the subject's motion is assumed to be negligible compared to gravity. Trigonometry is applied to the acceleration signal to obtain the incline of the accelerometer.

Bergmann *et al.* [8] used accelerometers in this manner to obtain the incline of the knee and ankle. The ankle incline angle was subtracted from the knee incline angle to obtain the knee joint angle for healthy subjects performing a stair climbing task, and produced comparable accuracy to joint angles determined by a motion capture system. However, there are numerous applications where the slow motion assumption does not hold. When the motion is fast, it is difficult to determine which component of the acceleration is caused by motion, and which is caused by gravity.

Dong *et al.* [23] considered the leg sagittal acceleration in gait movement. They utilized two accelerometers per link segment and combined them into a virtual accelerometer at a common location in order to use the difference in the accelerometers' readings to mathematically remove the link velocity. The incline angle was obtained from the accelerometers, and inter-frame subtraction was applied to determine joint angles. This approach requires two sensors per joint, making it cumbersome to use.

Low *et al.* [65] developed an accelerometer sensing suite, calculated orientation by trigonometry, and showed that going through a 180° motion at 160°/sec results in less than 1° error when compared to a rotary encoder. In a later work [60], they showed that their system performs well for sagittal arm motion, but have not demonstrated that the system can be generalized to 3D motion.

In general, using accelerometers alone to estimate joint angles is a limited solution, since this approach can only determine the link incline for joints moving in a plane perpendicular to gravity.

2.2.2 IMU Systems

Gyroscopes can be added to accelerometers to improve accuracy and handle fast and non-sagittal motions. However, gyroscopes can also introduce additional difficulties. The most well-known issue is the gyroscopic drift. Gyroscope calibration is often imperfect, and a non-zero DC bias is introduced into the sensor reading, producing a non-zero angular velocity measurement when no motion is actually occurring. When integrated, the estimated position diverges.

Complementary filter

Boonstra *et al.* [11] determined the incline of the thorax, thigh and calf by taking the arctangent of the accelerometer, integrating the gyroscope data and combining the two results. The acceleration signal was low-pass filtered to reduce sensor noise, while the gyroscope signal was high-pass filtered to reduce the impact of the DC offset. This arrangement, also known as the *complementary filter* [39], is common in mobile robotics, vehicular and aviation navigation applications due to its computational and mathematical simplicity. Boonstra's implementation does not recover joint angles, but sagittal incline. Although the above approaches showed good accuracy in 2D motion analysis in the sagittal plane, it is not trivial to generalize these algorithms to handle 3D motion, as all of these algorithms rely on the assumption that the motion is constrained in the gravity-acting plane, so that the accelerometer can be used as an inclinometer.

Ayoade *et al.* [3] used pre-built SparkFun MARG arrays, attached to each major limb segment to determine the orientation. The data was processed by a complementary filter. Since they employed a large number of sensors, they were not able to run the system at a speed higher than 50 Hz, which may have contributed to drift issues. Their system included an automatic drift correction mechanism, which required an initial time period before the motions started. The paper noted that data with drift was discarded, but did not state the effectiveness of the drift correction mechanism.

Strapdown integration

Williamson and Andrews [106] employed accelerometers during stationary or slow-moving situations to obtain the knee joint angle after applying inter-frame incline angle subtraction to the accelerometer signals. When the variance of the accelerometers reached a threshold, the angle was calculated by the integration of the gyroscope instead. This allows the gyroscope to be employed in fast-moving situations, and also resets the drift error by switching to the accelerometer signal on a regular basis. This process of directly integrating the gyroscope signal is commonly known as *strapdown integration* [53]. Williamson's system was used to recover sagittal knee angle in a sit-to-stand task, and showed improvements over accelerometer-only setups. However, this approach would only work with rhythmic motions, where stops are frequent. For a long motion, the integrator would not have a chance to reset, increasing the impact of drift on the joint estimates.

Luinge and Veltink [67] utilized strapdown integration in a Kalman filter framework. A set of IMUs were placed on the pelvis, trunk and forearm. Gyroscope data was used to generate the rotational matrices for the motions of each body segment, and allowed the direction of gravity in the sensor frame to be identified. This allowed the acceleration contributed by the motion to be disambiguated from the acceleration due to gravity, to obtain the inclination of the link. A gyroscope DC bias term was also included as a state in the Kalman filter, to control the gyroscopic drift. Although Luinge's implementation is capable of generating estimates of the full 3D rotation of each body segment with respect to the world frame, only the body segment incline and heading were evaluated in the paper. Luinge's algorithm treats each body segment as an independent rigid body, without considering any kinematic relationships or mobility constraints between body segments. This means that position and orientation estimates for adjacent body segments may drift and move with respect to each other in ways that are not physically realizable.

Zhou *et al.* [110] also utilized strapdown integration to obtain the joint rotation of each link, to develop a home-based stroke rehabilitation system. Unlike previous works, they subtracted out the gravity vector, as opposed to relying on it to obtain the link incline. They obtained joint end-effector location by double-integrating the gravity-subtracted acceleration signal to estimate the position. To reduce the impact of integration drift on the end-effector calculation, Lagrangian optimization was applied with kinematic constraints. Although this algorithm has a mechanism for dealing with accelerometer integrational drift, it does not have an explicit method for handling gyroscopic integrational drift.

Zhu and Zhou [111] removed inertial acceleration by strapdown integration, then combined the gravity vector from the accelerometer with the magnetometer to obtain incline and orientation. Inter-frame subtraction was used to obtain joint angles. This was facilitated within a Kalman filter framework. It was shown that this produced a more accurate joint angle recovery than using only strapdown integration or only accelerometer and magnetometer. This arrangement was verified with two planar elbow flexion/extension motions. One drawback of using magnetometers is that the magnetometer is sensitive to local fluctuations of the Earth's magnetic field that occur when ferromagnetic materials are nearby.

Roentenberg, Baten and Veltink [82] attempted to counter this issue by mounting a set of magnetic coils on the subject. At regular intervals, the magnetic coils generated a magnetic field which was detected by the magnetometers, providing relative distance and orientation information to supplement the existing IMU signals. Strapdown integration was applied to obtain the joint angle and remove the gravity vector. Dead reckoning was applied to obtain position. The magnetometer sampling rate was lower than that of the IMU, so the magnetometer information was only used to correct for drift. These sensors were combined via Kalman filter. This system was verified on several upper body motions, arm motions and walking. It is possible that ambient ferromagnetic materials or magnetic fields could still interfere with this system. This system also requires additional equipment to be mounted on the patient, which is undesirable.

2.2.3 Commercial Packages

In a follow-up work, Roentenberg, Luinge and Slycke [83] created the Xsens MVN suit[6] for kinematic measurements. The MVN suit consists of 17 MARG modules integrated into a wearable suit. Orientation of the sensors is obtained by a series of calibration motions. Link lengths are measured manually and inserted into the kinematic model for accuracy. The Kalman filter is applied to combine the sensor data to determine segment positions.

Similar commercial packages, such as the Animazoo IGS series[7], or the Biosyn FAB[8], are widely used in academic research. However, these commercial solutions may not be well suited for physiotherapy applications. These solutions tend to be expensive, and require a long calibration routine, which may not be possible for patients who have difficulty moving. Suit size and fit is also a concern, especially as obesity is a risk factor for many musculoskeletal injuries. Furthermore, equipment upkeep, such as cleaning, is nontrivial, especially for the suit-based products.

2.2.4 Summary

Most of the existing works implement body segment orientation estimation without the use of kinematic constraints (e.g. [11], [67]). This allows these algorithms to be applied on any arbitrary object and determine its orientation with respect to the world frame. However, human kinematics constrains the possible movements that can be physically realized. These constraints should be considered to improve joint angle estimates, such as the knee joint's inability to perform abduction motion. Without a kinematic model, the knee joint would be modeled as a 3 DOF joint, which it is not, allowing drift to occur in this direction. By utilizing a kinematic model in the proposed algorithm, the space of possible motions is realistically constrained, providing a more accurate estimate of joint angles.

Without a kinematic model, these algorithms also cannot provide joint angle estimates. Instead, they provide orientation estimates relative to the world frame, such as incline angle from ground or heading angle from the vertical. To produce more clinically relevant

[6] Xsens Technologies, www.xsens.com

[7] Animazoo Ltd., www.animazoo.com

[8] Biosyn Systems, www.biosynsystems.net

information, the estimated orientations must be converted to joint angles by incorporating kinematic information in a post-processing step. By incorporating the kinematic information directly into the estimation process, the proposed algorithm produces the joint angle data of interest directly and improves robustness against drift.

In this thesis, an algorithm (Chapter 4) to estimate arbitrary 3D human motion is proposed. Small and light-weight IMU sensor packages attached to the thigh and calf are used to wirelessly transmit the patient's acceleration and gyroscope profile during motion. A minimal number of sensors are employed to minimize the encumbrance on the patients. IMU sensors are unaffected by environment factors, which gives them an advantage over magnetometer-based methods. Using a kinematic model, the sensor information is fused with an extended Kalman filter. Kinematic constraints and filter noise adaptation are employed to reconstruct the joint angles of the subject. The proposed system is capable of recovering the leg pose in terms of joint angles during arbitrary 3D motion. This is a significant improvement over existing works based on incline angles in the sagittal plane, which are incapable of providing 3D angle recovery.

2.3 Segmentation and Identification

Technology to measure and analyze human motion has the potential to provide physiotherapists with more accurate tools for assessment and progress measurement, as well as to provide patients with real-time feedback.

To enable automated measurement and analysis, the system must measure the human movement and identify exercise movement segments from the time-series data. Human movement can be measured via either motion capture systems [43, 68] or ambulatory sensors such as IMUs [64, 84, 109]. Given the measured time-series data, *segmentation* is the process of identifying the starting and ending locations of each movement of interest, breaking a continuous sequence of motion data into smaller components, termed *motion primitives*. If the patient is performing more than one type of exercise in a given recording session, *identification* (i.e., labeling) of each segment with the appropriate exercise type is also required.

One major application field of segmentation is imitation learning for robotics applications. The most intuitive way for humans to teach any given action is by demonstration, which would make robotics more accessible to non-specialists [10, 15, 42, 44]. This is particularly useful in potentially dangerous environments where robots can replace humans, such as in manufacturing plants or search and rescue [32].

Another major application field is gesture recognition. Gesture recognition allows users to interact with computer systems in a more intuitive fashion, such as through handwriting and touchpads [41, 105], or camera-based systems [71]. It is also useful for sifting through large libraries of motions, such as in cinematography [5, 59], or to ensure a given motion is correct, in fields such as signature verification [47] or sports training and other medical applications [25, 99].

The large number of DOFs in the movement data makes scalability an important issue. Simple models like singular limb studies contain 4-6 DOFs, while full body human models can have 20-30 DOFs [77]. Some algorithms scale poorly to higher dimensions, and cannot be used in an on-line setting. For rehabilitation, on-line processing is essential to allow the system to inform the patient if a motion is being performed incorrectly.

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

2.3.1 Template-free Approaches

Zero-velocity Crossings

If the motions to be observed are not known *a priori*, then non-template based methods are required. Pomplun and Matarić [79] employed zero-velocity crossings (ZVCs) to identify points where the velocity value changes sign, denoting when a joint segment direction change, as segment points. If multiple joints are examined simultaneously, segment points can be declared by thresholding the sum of squares of the velocities. A minimum threshold for segments was included, to prevent spurious ZVCs from creating large numbers of false positives. Although a fast algorithm, ZVC tends to over-segment, particularly with noisy data or with increasing number of DOFs. Since ZVC does not consider motion templates, it is difficult to determine which crossing points can be safely ignored. In addition, the ZVC algorithm does not provide a method for motion identification. Pomplun suggests that a distance metric can be used, but these metrics are sensitive to spatial and temporal variations, and may not provide reliable movement labels. The algorithm was used to assess a 11-subject study on human imitation learning, where video clips of arm motions were shown to the participants. The participants were either instructed to practice the observed motion before data collection, or not. The collected data was segmented and compared against the demonstrator's motions. Since the study was on the impact of rehearsing on the ability to accurately reproduce a motion, segmentation accuracy of the algorithm was not explicitly reported.

Fod *et al.* [30] expanded on the basic ZVC approach, considering both Pomplun's velocity thresholds, as well as declaring a segment point when multiple DOFs exhibit ZVCs within a short timespan. This method works only with certain types of movements, since not all motions would be characterized by multiple ZVCs. For example, a circular motion would not cause ZVCs in joint-based DOFs. Slow movement at the start or end of segments would also prevent a ZVC declaration at the correct location, even though the movement has effectively ceased.

Lieberman and Breazeal [61] also expanded on the ZVC, demonstrating that the velocity threshold method can be complemented by other signal types, such as tactile contact. The segmenting velocity threshold was also modified so that it is dynamically determined as a function of the velocity as opposed to being a static value. This allows the threshold to account for speed changes in the motion. However, using a single velocity threshold means that DOFs that have large velocity valleys and peaks would dominate the auto-threshold determinations, thus hiding the segmentation points that would be best characterized by DOFs with small valleys and peaks. Also, at high dimensionality, the algorithm would likely report weaker performance since most of the DOFs are likely to be stationary, thus underreporting the velocity value and leading to false positives. Lieberman applied this algorithm to a human demonstrator in a motion capture suit, to generate and segment motions for imitation learning purposes, so explicit segmentation metrics were not provided.

Techniques that rely purely on ZVC tend to be difficult to use, as spurious crossings can often occur. Noises and vibrations can cause excess ZVCs, leading to oversegmentation. Smooth transition sequences of motions can also lead to the velocities to approach ZVC, but not actually cross zero. Although it may be possible to combine oversegmented components, it is difficult to determine which segment points may be ignored without additional algorithmic guidance. In addition, ZVC performs segmentation only and does not identify the underlying motion. It is possible to use distance metrics and thresholding, such as mean-square-error, to label the movements, but this approach does not provide any mechanism to account for temporal variations. For complex motions, multiple DOFs may exhibit ZVCs simultaneously, but slightly offset from each other. In these cases, it may be difficult to determine which ZVC marks the actual segment point. Although ZVC provides a fast and intuitive segmentation method, its various shortfalls make it difficult to use.

Variance

Koenig and Matarić [54] proposed considering changes in signal variance as segmentation points. A moving window is used to examine the input data. When the windowed variance is high for several DOFs, it implies that the current windowed data is in the progress of changing into another primitive, and thus is an appropriate location to segment. This allows for a quick and systematic method for segment point declaration. However, examining only variance may not provide sufficient information. A participant may perform a motion that inherently has a large variance value, or perform several different motions that involve the same DOFs but is a different motion. For example, an arm flexion and extension motion may be considered as two different motions, but since variance calculations ignore the direction of the movement, the variance may not have changed significantly between the flexion and the extension. Keonig applied this method to navigational robotic imitation learning, but it may not be as suitable for movement segmentation.

A similar work from Kohlmorgen and Lemm [55] applied the same principles, using the observational probability distribution as the segmentation cost function. The data is windowed and the probability density function (PDF) of the windowed data is calculated, for each adjacent window, over the length of the observation. The PDFs are used to train a hidden Markov model (HMM), which is traversed by the Viterbi algorithm [80] to generate the most likely state sequences from the PDFs. These state transitions mark the segment points. Janus and Nakamura [46] applied Kohlmorgen's algorithm [55] to human movement data. HMM state changes occurred when the signal's distribution function is sufficiently different from the previous state, suggesting possible segmentation points at these state changes.

2.3.2 Deterministic Template Approaches

If the motions to be observed are known, motion templates can be used to assist in the identification of motions. However, temporal and spatial variability of motion makes direct comparison of the template with the observation data difficult. Any segmentation and identification should provide some mechanism to account for these variabilities.

Dynamic time warping (DTW) is one example of a template-based method. Originally a popular technique in the field of speech recognition [74], DTW found application in motion segmentation as well. DTW [87, 49] identifies the temporal variations between the observed motion and the motion template by selectively warping the time scale of the observed sequence to the template. The warping is done by employing dynamic programming

(DP). Specifically, DTW calculates the distance from each data point of the template to each data point of the observation, creating a distance matrix. The warping path that leads to the minimum distance is selected, creating a mapping between the template and the observation. The distance metric utilized is typically the Euclidean distance, but alternative distance metrics, such as the autocorrelation [49] have been proposed. Therefore, segments can be accurately segmented and identified even with significant spatial and temporal variations between the observation and the template. However, DP-based algorithms become very computationally expensive at higher dimensionality, preventing them from being utilized on-line.

Poor warping can also lead to singularity issues, where large portions of one motion are warped to small portions of the other. The severity of the singularity issue can be mitigated by constraining the warping path. However, generating an appropriate constraining band requires the algorithm to be tuned, so a fixed band is typically used [87]. Ratanamahatana and Keogh [81] proposed an algorithm that calculates an optimal band dynamically. This algorithm starts by performing DTW between two data series with a constraint band width of zero, and increases the band width until the distance between the two data series is minimized, or a threshold is reached. The algorithm then bisects the data, and performs the same iteration individually, on each half of the data series. The algorithm continues to iterate with smaller data lengths, until a minimum length of $\sqrt{m}/2$ is reached, where m is the length of the data series, as anything smaller increases the risk of overfitting. This creates a band that varies in width as a function of time. The algorithm was verified on human arm motion performing gun-draws, synthetic data simulating nuclear power plant instrumentation failure and a handwritting dataset; it was found that the dynamic band method reported better identification rates then the existing fixed-band method. Although this algorithm produces a more customized warping band, it suffers from several drawbacks. This iterative search adds more computation time to an algorithm that is already known to be computationally expensive. A new constraint band needs to be generated for different situations, since a constraint band optimized for the mapping of a given participant performing a specific motion may not be optimal for another participant performing the same motion. Although this method addresses the singularity issue, it does so by increasing the calculation time.

Keogh and Pazzani [50] noted that classic DTW considers only the position data and does not account for higher level features. They proposed the Derivative DTW (DDTW), and examined the derivative of the data streams instead by using the square of the difference of the derivatives of the two signals instead of the Euclidean distance. This allows the algorithm to negate global effects such as signal bias. The derivative also allows more of the latent features to be emphasized. Using only the Euclidean distance, DTW will map two points of identical value together, even if one point is part of a falling trend and the other is part of a rising trend. Using the derivate will allow the larger pattern to be captured. This algorithm was verified on a set of space shuttle sensor readings, the currency exchange rate between the German Deutschmark and five other European currencies over the span of six months, as well as a set of electroencephlograph (EEG) measurements. All sets of data show similar trends but are non-identical. Comparison between DTW and DDTW shows that DTW tends to be overaggressive in the warping, while DDTW provides a more accurate mapping. Although the DDTW algorithm outperforms standard DTW, they have similar runtime, meaning that it does not scale well to higher dimensions.

Ilg *et al.* [43] employed DTW in a multi-tier fashion. The observation signal is dimensionally reduced by removing all data points that are not at a velocity zero, as velocity zeros denote turning points in the motion, and thus can be considered as key features of the motion. DTW is performed on this reduced dataset to align these key features. A tolerance is included to allow for missed key features, as the template and the observation may not have the same velocity zeros, to reduce the number of mapping singularities. Each window in these high-level segments is resampled to have the same number of data points. A finer alignment is performed in each of these windows. Assuming that the observed motion can be warped to the template with the proper temporal and spatial warping, the algorithm uses DTW to calculate an optimal temporal mapping path between the template and the observation, and applies some interpolated shifting around the suggested mapping path to minimize the temporal difference between the two signals. Once the optimal time warping is found, the spatial distance offset can be calculated. The temporal and spatial warping variables are also constrained to minimize the amount of warping required to obtain the best fit [33]. The algorithm was implemented as part of a motion generation algorithm, and not specifically for segmentation, so segmentation and identification accuracy were not reported. Similar to previous DTW algorithms, this algorithm does not address the computational costs of using DP, and thus may not scale well to higher dimensions.

2.3.3 Probabilistic Template Approaches

An alternative method is to employ non-deterministic algorithms, such as HMMs. Templates encoded as HMMs can be used as to assess the similarity between a template and an observation sequence.

Bashir *et al.* [6] crops observation data based on zero-crossing occurrences in the curvature data. The Cartesian curvature was used as it is a function of the observation data and its first and second derivative, and therefore incorporates high-level features inherently.

The cropped data is reduced using principal component analysis (PCA). Motions that have similar PCA coefficients are clustered together via k-means spectral clustering, and are used to train HMMs. Observation segmentation was performed by reducing the observation data to PCA coefficient vectors, and comparing them to the HMMs to calculate likelihood. However, utilizing PCA coefficients instead of the original data can be difficult, as similar motions may have similar PCA profiles, making them difficult to differentiate. Using curvature to segment may also lead to oversegmentation issues such as those found in the ZVC algorithms.

Other researchers used HMM to both segment and identify. Lv and Nevatia [68] used HMM templates as classifiers in an AdaBoost algorithm. Features such as joint angles and joint Cartesian coordinates were used. HMM templates are generated for M action types and N features, resulting in $M \times N$ number of classifiers. Multi-class AdaBoost was employed to weigh the discriminative powers of each of the HMM classifiers. Segmentation was determined by dynamic programming. The observation data is split into two, with the first window starting at some minimum length l_{min}, and increased per iteration. Both ends of the data are ran through the AdaBoost classifiers, and the window configuration that resulted in the highest likelihood was selected. The algorithm is run multiple times, with the starting point of the first window advancing at each run. The algorithm was verified on various upper-body motions, and was shown to be robust even when a large number of features are examined. However, this method requires a large computational cost for both training and segmenting, due to the usage of dynamic programming, and thus cannot be used on-line.

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

Keogh *et al.* [48] proposed a 2-tier on-line segmentation algorithm. They employ a sliding window and mean square error (MSE) for coarse segment estimation. Following coarse segmentation, a second window and additional distance metrics were used to determine the actual segment. Amft *et al.* [2] extended this to a 3-tier system, by adding HMMs as an identification tool for 4 DOF arm motion identification. They report an average accuracy of 72% (94%, ignoring false positives and negatives) for 4 motion primitives.

2.3.4 On-line Template Generation

On-line template construction has also been proposed. Kulić *et al.* [57] extended Kohlmorgen's algorithm [55] by clustering together previously segmented sequences to generate new templates in real-time. Once a segment window has been identified, the segment is modeled as a HMM. The Kullback-Leibler distance between the observed HMM and existing models is calculated. If the distance is small, then the observation HMM is merged into the corresponding existing HMM. If not, it is added to the template collection, and used to improve the segmentation. The algorithm was verified on an 18-minute full-body motion sequence, and shows good segmentation accuracy, but also suffers from false positives due to the algorithm oversegmenting motion sequences into smaller subsequences that were considered to be a single segment by the manual segmentation.

Baby and Krüger [4] proposed a similar learning technique. To generate a primitive template, a HMM is generated from the observations. When new observation data becomes available, the probability of generating this new observation data, given existing HMMs, is calculated. If this value is low, implying that the newly observed motion is not similar to existing models, a new HMM is generated. The Viterbi algorithm is used to determine the underlying states from the HMMs.

2.3.5 Summary

Numerous algorithms have been proposed to date to address the problem of segmentation and identification. Focusing on techniques which can be employed on line, as required by the rehabilitation application, ZVC provides an intuitive template-free segmentation method with low runtime, but tends to oversegment. DTW accounts for both spatial and temporal variations between motions, but is also prohibitively expensive to perform online. HMM serves as a computationally lighter alternative to DTW, but can also become computationally expensive if the likelihood function is utilized repeatedly.

Chapter 5 proposes a template-based on-line technique based on both velocity features and probabilistic templates. The system consists of a training phase, where exemplar data are used to create feature and HMM motion templates, and a segmentation phase, where the observed data is swept for characteristic features that match the feature templates.

When a segment candidate is found, the HMM templates are used to identify the motion and confirm the exact segment point locations.

Unlike many of the existing approaches to temporal segmentation, the proposed algorithm is capable of running on-line during motion execution and provides accurate segments that reject false positives, with the assistance of template information. It is suitable for high dimensionality signals from full body motion, and explicitly models both spatial and temporal variability to provide accurate segmentation results without the use of data warping or the need for a subject-specific template.

Chapter 3

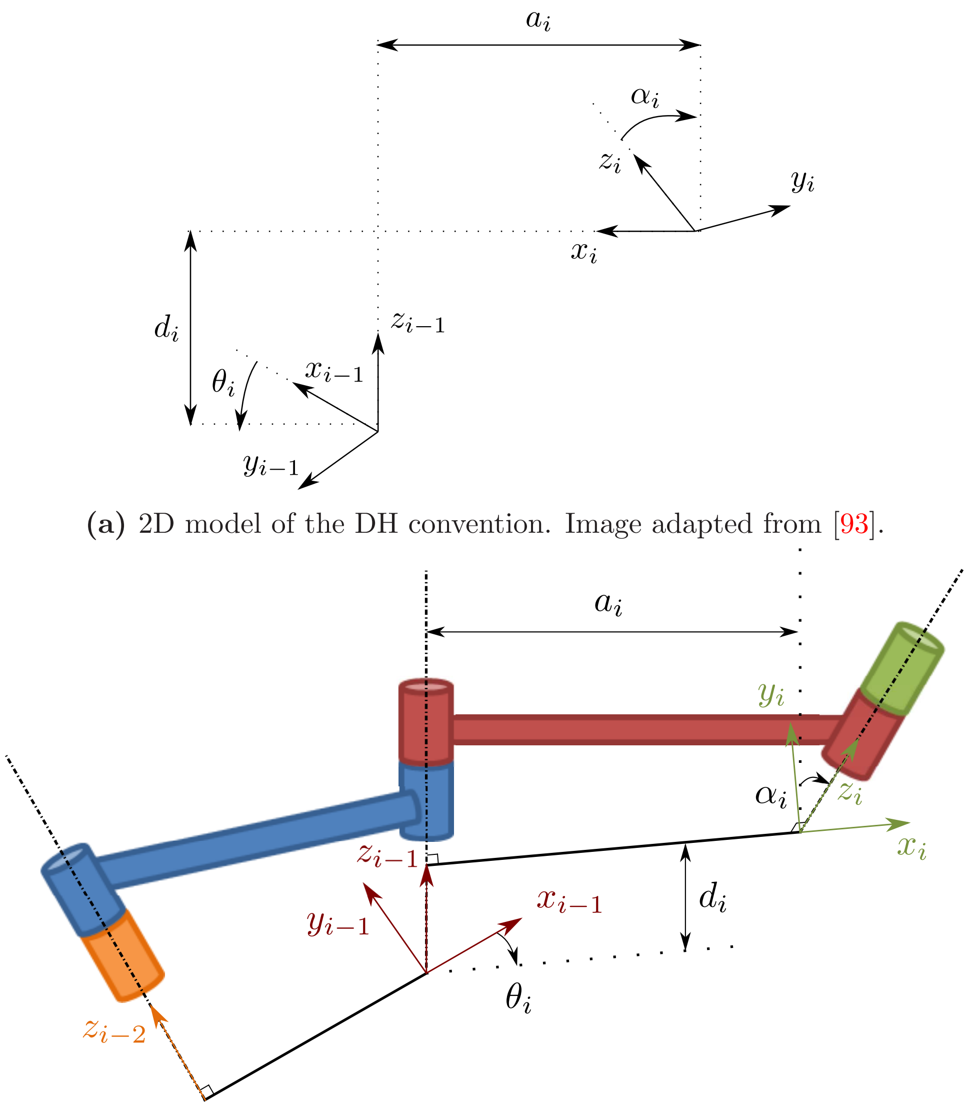
Background

This chapter overviews the existing concepts, algorithms and techniques employed in this thesis. It provides an overview of several robotic modeling concepts employed in subsequent chapters, such as the Denavit-Hartenberg frame assignment convention and forward kinematics for chains of articulated rigid bodies. General operating principles of inertial measurement units are reviewed. The Kalman filter, a sensor fusion algorithm for linear systems, is presented, along with its non-linear extension, the extended Kalman filter. Lastly, the mathematical framework for modeling time-series human-motion data, the hidden Markov model, is overviewed.

3.1 Robotic Modeling

3.1.1 Denavit-Hartenberg Convention

The Denavit-Hartenberg (DH) frame convention [98] is commonly utilized to model articulated chains of rigid body links by modeling the system mobility as a series of single DOF joints. The DH convention attaches a frame to each rigid body link, and defines the transformation from one link frame to the next as a set of 4 transformations: link length, link twist, link offset and joint angle. The definitions of these variables can be found in Table 3.1. To fully represent the 6 DOF transformation between two frames in four variables, the DH convention constrains the frames assignment by two rules: The axis x_i is perpendicular to the axis z_{i-1}, and the axis x_i intersects to the axis z_{i-1}. Since the equations for the DH formulation are defined recursively, joint i is always rigidly attached to frame $i-1$, and actuates frame i. See Figure 3.1 for an illustration.



(a) 2D model of the DH convention. Image adapted from [93].

(b) 3D model of the DH convention. Image adapted from [34].

Figure 3.1: Coordinate frames satisfying the DH convention.

Table 3.1: DH convention parameter definitions.

Joint angle	θ_i	angle between the x_{i-1} and the x_i axes, about the z_{i-1} axis
Link offset	d_i	distance from frame origin $i-1$ to the x_i axis, along the z_{i-1} axis
Link length	a_i	distance between the z_{i-1} and the z_i axes, along the x_i axis
Link twist	α_i	angle from z_{i-1} to the z_i axis, about the x_i axis

These four variables generate an overall transformation matrix from the $i-1^{th}$ frame to the i^{th} frame:

$$T_{i-1,i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The transformation matrix from base frame to the i^{th} frame can be found by:

$$T_{0,i} = T_{01}T_{12}T_{23}\dots T_{i-1,i} \tag{3.2}$$

$$R_{0,i} = T_{0,i}(1:3, 1:3) \tag{3.3}$$

$$\boldsymbol{x_i^0} = T_{0,i}(4, 1:3) \tag{3.4}$$

where $R_{0,i}$ is the rotation matrix from the base frame to frame i, and $\boldsymbol{x_i^0}$ is the end-effector position. $R_{0,i}$ is generated from the first three columns and rows of $T_{0,i}$. $\boldsymbol{x_i^0}$ is generated from the first three rows of the last column of $T_{0,i}$.

If a tool is offset from the end effector frame, and additional tool transform can be added to find the position of the tool:

$$T_{0,t} = T_{0,i}T_{tt} \tag{3.5}$$

where T_{tt} is a tool transform matrix that relates the end effector frame to the tool frame.

3.1.2 Forward Velocity and Acceleration Kinematics

The following section derives the forward velocity and acceleration equations for revolute joints only, but the concepts presented can be easily extended to prismatic joints as well. Several scalar variables introduced here, such as the joint velocity $\dot{q}_i$, are vectorized so that the matrix multiplications and cross products can be performed properly, as per the following: $\boldsymbol{\dot{q}_i} = [0; 0; \dot{q}_i]$. This convention holds for all of Section 3.1.2.

Given the total angular velocity of the preceding frame $\boldsymbol{\omega_{i-1}^{i-1}}$ and the joint velocity $\dot{q}_i$, generated by revolute joint i, the total angular velocity $\boldsymbol{\omega_i^i}$ of the next frame is:

$$\boldsymbol{\omega_i^i} = R_{i-1,i}^T\boldsymbol{\omega_{i-1}^{i-1}} + R_{i-1,i}^T\boldsymbol{\dot{q}_i} \tag{3.6}$$

This formulation allows the angular velocity of each segment to be determined recursively based on the motion of the current link (the i^{th} frame) and the link immediately previous to it (the $i-1^{th}$ frame). The angular velocity $\dot{q}_i$, describing the velocity between frames $i-1$ and i, is included by rotating $\dot{q}_i$ into the current frame with a rotation matrix $R_{i-1,i}$. $R_{i-1,i}$ is obtained directly from the DH frame definition.

The angular velocity is differentiated to obtain the angular acceleration $\boldsymbol{\alpha_i^i}$:

$$\boldsymbol{\alpha_i^i} = R_{i-1,i}^T \boldsymbol{\alpha_{i-1}^{i-1}} + R_{i-1,i}^T \boldsymbol{\ddot{q}_i} + \boldsymbol{\omega_i} \times (R_{i-1,i}^T \boldsymbol{\dot{q}_i}) \tag{3.7}$$

The linear velocity of the current frame $\boldsymbol{\dot{x}_i}$ is obtained by computing the cross-product of the angular velocity and the displacement vector from the origin of the current frame to the origin of the previous frame, $\boldsymbol{r_i}$:

$$\boldsymbol{\dot{x}_i} = R_{i-1,i}^T \boldsymbol{\dot{x}_{i-1}} + \boldsymbol{\omega_i^i} \times \boldsymbol{r_i} \tag{3.8}$$

Differentiation is carried out again to obtain the linear acceleration, $\boldsymbol{\ddot{x}_i}$:

$$\boldsymbol{\ddot{x}_{i,i}} = R_{i-1,i}^T \boldsymbol{\ddot{x}_{e,i-1}} + \boldsymbol{\alpha_i^i} \times \boldsymbol{r_i} + \boldsymbol{\omega_i^i} \times \boldsymbol{\omega_i^i} \times \boldsymbol{r_i} \tag{3.9}$$

$$\boldsymbol{\ddot{x}_{e,i}} = R_{i-1,i}^T \boldsymbol{\ddot{x}_{e,i-1}} + \boldsymbol{\alpha_i^i} \times \boldsymbol{r_i} + \boldsymbol{\omega_i^i} \times \boldsymbol{\omega_i^i} \times \boldsymbol{r_i} + R_{0,i}\boldsymbol{g} \tag{3.10}$$

In Equation 3.9, the first term represents the acceleration transmitted from the previous frame, while the second and third terms represent the tangential and radial acceleration, respectively. An extra term, $R_{0,i}\boldsymbol{g}$, is added to Equation 3.10 (describing the end effector acceleration) in order to model the gravity term $\boldsymbol{g} = [9.81; 0; 0]$, and to rotate it into the local frame.

3.2 Sensor Fusion

3.2.1 Inertial Measurement Units

IMUs refer to electronic devices that measure an object's motion. They typically consist of some or all of the following components:

accelerometer	measures linear acceleration
gyroscope	measures angular velocity
magnetometer	measures heading via Earth's magnetic field

These systems were initially developed for vehicular navigation. Until the development of micro-electro-mechanical systems (MEMS), these sensing packages tended to be large and thus unsuitable for human tracking. Advancements in semiconductor fabrication have allowed for the miniaturization of these systems and made them very cost-effective, allowing for their widespread usage.

Accelerometer

An accelerometer determines linear acceleration with test masses. It can be conceptualized as a known test mass suspended in a chamber by three springs mounted perpendicularly to each other. When acceleration is applied to the test mass, the length of spring displacement can be measured, and thus force acting on the test mass can be determined.

A MEMS accelerometer follows this principle. A suspended mass (seismic mass) is allowed to move according to accelerations it experiences. The seismic mass has fingers that extend into sets of parallel plates, forming a differential capacitor. When the fingers move due to acceleration on the seismic mass, the capacitance changes proportionally to the acceleration felt.

Gyroscope

A gyroscope determines angular velocity with a heavy spinning disk (flywheel) mounted within a series of free-rotating rings (gimbals). External torques to the flywheel changes the flywheel's angular momentum. In order to conserve the angular momentum's direction, the flywheel rotates towards the direction of the torque. The angular velocity of this rotation, known as the precession, can be used to determine the external velocity.

A MEMS gyroscope operates under a similar principle, but does not rely on gimbals. Also known as the vibrating structure gyroscope, the common MEMS gyroscope relies on two vibrating masses that oscillate in opposite directions, as a differential capacitor. When an angular velocity is applied to the gyroscope, the Coriolis forces on the vibrating masses push them apart, resulting in a capacitance change. When a linear acceleration is applied, the two masses move in the same direction, so no capacitance change is experienced.

Magnetometer

A magnetometer, which operates similarly to a magnetic compass, utilizes a suspended object sensitive to magnetic fields that aligns its orientation to the field to provide an orientation reading.

A MEMS magnetometer can use a variety of different physical phenomena for sensing. The most common is the Hall effect sensor. Electric current running through a conductor is deflected transversely to the current direction due to the Lorentz force effect. The current deflection results in a voltage potential difference across the wire in proportion to an external magnetic field.

3.2.2 Kalman Filter

The Kalman filter [104] is a sensor fusion technique that estimates system state from noisy observations, assuming that the linear state and observation model are known, and that both of the process and measurement noise are zero-mean Gaussian functions. For the standard Kalman filter, the state estimate $\boldsymbol{s_t}$ and observation update $\boldsymbol{z_t}$ are defined as:

$$\boldsymbol{s_t} = A\boldsymbol{s_{t-1}} + \boldsymbol{w_{t-1}} \tag{3.11}$$

$$\boldsymbol{z_t} = C\boldsymbol{s_t} + \boldsymbol{v_t} \tag{3.12}$$

Equation 3.11 is the process update equation that relates the previous state of the system $\boldsymbol{s_{t-1}}$ to the current state $\boldsymbol{s_t}$. The process noise $\boldsymbol{w}_{t-1}$ is assumed to be zero-mean and Gaussian, with a covariance of Q_t. Equation 3.12 is the measurement update equation, relating the current state $\boldsymbol{s_t}$ to the measurement vector $\boldsymbol{z_t}$. $\boldsymbol{v}_t$ is the measurement noise, and is also assumed to be zero-mean and white, with a covariance of R_t.

At each time step, an *a priori* state estimate $\boldsymbol{\hat{s}_t^-}$ can be made using Equation 3.11. Then, this state estimate can be updated with the current observation information by:

$$\boldsymbol{\hat{s}_t} = \boldsymbol{\hat{s}_t^-} + K_t(\boldsymbol{z_t} - C\boldsymbol{\hat{s}_t^-}) \tag{3.13}$$

The residual, $\boldsymbol{z_t} - C\boldsymbol{\hat{s}_t^-}$, indicates the discrepancy between the predicted measurement $C\boldsymbol{\hat{s}_t^-}$ and the actual measurement $\boldsymbol{z_t}$. If the residual term is zero, it implies that the prediction and the measurement are equal. The Kalman gain K_t is derived by minimizing the error covariance P_t^- for the *a posteriori* state estimate $\boldsymbol{\hat{s}}$. Given a linear system and white noise, a closed solution can be determined and is given by:

$$K_t = P_t^- C^T (CP_t^- C^T + R_t)^{-1} \tag{3.14}$$

The Kalman equations can be separated into two sections. At each time step, the state prediction equations are performed, then updated upon with the measurement update equations. These equations are summarized in Figure 3.2.

state prediction equations

$$\begin{aligned}\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-} &= A\hat{\boldsymbol{s}}_{\boldsymbol{t-1}}\\ P_t^{-} &= AP_{t-1}A^T + Q_t\end{aligned}$$

measurement update equations

$$\begin{aligned}K_t &= P_t^{-}C^T(CP_t^{-}C^T + R_t)^{-1}\\ \hat{\boldsymbol{s}}_{\boldsymbol{t}} &= \hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-} + K_t(\boldsymbol{z}_{\boldsymbol{t}} - C\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-})\\ P_t &= (I - K_tC)P_t^{-}\end{aligned}$$

Figure 3.2: Kalman filter prediction and update equations.

3.2.3 Extended Kalman Filter

The Kalman filter is applicable only to systems with linear models. The extended Kalman filter (EKF) [104] extends the Kalman filter to non-linear systems by linearizing the state and observation equations with a first-order Taylor series expansion, then utilizing the results in the Kalman filter.

The state and measurement equations can be written in a general form: $\boldsymbol{s}_{\boldsymbol{t}} = f(\boldsymbol{s}_{\boldsymbol{t-1}}, \boldsymbol{w}_{\boldsymbol{t-1}})$, where f is the non-linear state evolution function and $\boldsymbol{z}_{\boldsymbol{t}} = h(\boldsymbol{s}_{\boldsymbol{t}}, \boldsymbol{v}_{\boldsymbol{t}})$, where h is the non-linear measurement equation. These equations are linearized as follows:

$$\boldsymbol{s}_{\boldsymbol{t}} \approx \tilde{\boldsymbol{s}}_{\boldsymbol{t}} + A_t(\boldsymbol{s}_{\boldsymbol{t-1}} - \hat{\boldsymbol{s}}_{\boldsymbol{t-1}}) + W_t\boldsymbol{w}_{\boldsymbol{t-1}} \tag{3.15}$$

$$\boldsymbol{z}_{\boldsymbol{t}} \approx \tilde{\boldsymbol{z}}_{\boldsymbol{t}} + C_t(\boldsymbol{s}_{\boldsymbol{t}} - \tilde{\boldsymbol{s}}_{\boldsymbol{t}}) + V_t\boldsymbol{v}_{\boldsymbol{t}} \tag{3.16}$$

where $A_t = \partial f/\partial \boldsymbol{s}_{\boldsymbol{t}}$ is the Jacobian matrix of the partial derivatives of the state equation f with respect to the state $\boldsymbol{s}_{\boldsymbol{t}}$, $\tilde{\boldsymbol{s}}_{\boldsymbol{t}} = f(\boldsymbol{s}_{\boldsymbol{t-1}}, \boldsymbol{w} = 0)$ is the noise-less state estimate, and $W_t = \partial f/\partial \boldsymbol{w}_{\boldsymbol{t-1}}$ is the Jacobian matrix of partial derivatives of f with respect to the process noise $\boldsymbol{w}_{\boldsymbol{t-1}}$. $\boldsymbol{z}_{\boldsymbol{t}} = h(\boldsymbol{s}_{\boldsymbol{t}}, \boldsymbol{v} = 0)$ is the measurement vector, whereas $\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ is the noiseless estimate of the measurement. $C_t = \partial h/\partial \boldsymbol{s}_{\boldsymbol{t}}$ is the Jacobian matrix of the partial derivatives of the measurement equations h with respect to the state $\boldsymbol{s}_{\boldsymbol{t}}$, and $V_t = \partial h/\partial \boldsymbol{v}_{\boldsymbol{t}}$ is the Jacobian matrix of partial derivatives of h with respect to the measurement noise $\boldsymbol{v}_{\boldsymbol{t}}$.

The Kalman gain K_t is then computed as follows:

$$P_y = C_tP_t^{-}C_t^T + V_tR_tV_t^T \tag{3.17}$$

$$K_t = P_t^{-}C^T(P_y)^{-1} \tag{3.18}$$

where P_y is an intermediate noise covariance matrix. The formulation of the extended Kalman filter is shown in Figure 3.3 [104].

state prediction equations

$$\begin{aligned}\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-} &= f(\boldsymbol{s_{t-1}}, \boldsymbol{w} = 0)\\ P_t^{-} &= AP_{t-1}A^T + W_tQ_{t-1}W_t^T\end{aligned}$$

measurement update equations

$$\begin{aligned}P_y &= C_tP_t^{-}C_t^T + V_tR_tV_t^T\\ K_t &= P_t^{-}C^T(P_y)^{-1}\\ \hat{\boldsymbol{s}}_{\boldsymbol{t}} &= \hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-} + K_t(\boldsymbol{z_t} - h(\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-}, \boldsymbol{v} = 0))\\ P_t &= (I - K_tC_t)P_t^{-}\end{aligned}$$

Figure 3.3: Extended Kalman filter prediction and update equations.

These equations can be separated into *a priori* and *a posteriori* equations, indicating operations before and after the incorporation of the measurement update step. The *a priori* state $\hat{\boldsymbol{s}}_{\boldsymbol{t}}^{-}$ and error covariance P_t^{-} are estimated by the state prediction equations, and are based on the state of the previous timestep $\boldsymbol{s_{t-1}}$. The current measurement $\boldsymbol{z_t}$ is incorporated into the Kalman gain K_t calculation, allowing for the calculation of the *a posteriori* state $\hat{\boldsymbol{s}}_{\boldsymbol{t}}$ and error covariance P_t in the measurement update equations [104].

3.3 Motion Modeling

3.3.1 Hidden Markov Model

The HMM [80] is a stochastic model where the process or signal being modeled is represented by an evolving unobservable state. The HMM can be conceptualized as a set of interconnected nodes. At each timestep, the system undergoes a state transition, which may lead the system to remain in the current state or transition to another state. The state has the Markov property, i.e., the next state depends only on the current state. The transitions are represented by the state transition matrix, denoting the probabilities that the system would shift into another state, given its current state. In an HMM, the state itself is not directly observable, so the underlying state is inferred by the probabilistic relationship between the hidden state and its corresponding observable output. This contrasts with the Markov chain, where the state is directly observable. The HMM is commonly used to model human motion [103], as the model can capture both spatial and temporal variability inherent in human motion.

The HMM stores its model information as a set of three variables, commonly referred to as $\lambda = (A_H, B_H, \boldsymbol{\pi_H})$:

- The state transition probability matrix, A_H, represents the probability that a given state will transition into another state, or remain in the current state. If the probability that the state $a_{i,i}$ will remain in itself is high, this implies that the motion is slow-moving. In an ergodic (fully connected) HMM, where a given state can transition into any other state, A_H will not have any zero values. In a HMM that is non-ergodic, such as a left-right model, where a given state can transition only to itself, or advance to the next state, A_H will have zeros in some of its entries to denote impossible state transitions.

- The observation probability matrix, B_H, contains data on the observation model for each state. This variable denotes the possible observable outputs for each state. For discrete variables, B_H contains the possible output alphabet. For continuous variables, Gaussian or mixture of Gaussian distribution models are typically used. The mean $\boldsymbol{\mu_H}$ and covariance $\boldsymbol{\Sigma_H}$ of each state are stored. For human motion, this represents key pose positions and the range of variability for each pose over all the motion exemplars, respectively. B_H is also commonly known as the emission probability.

- The initial state distribution, $\boldsymbol{\pi_H}$, represents the probability that an observation sequence begins in a given state. Since the HMM only considers the previous state to calculate the next state, the initial state is influential in determining the transition sequences. This parameter is important in a fully connected HMM, but in a left-right HMM, this variable is less significant, as for the left-right HMM, it is typically assumed that each observation sequence begins in the initial state. $\boldsymbol{\pi_H}$ is also commonly known as the prior.

Thus, at each timestep, the current state can be determined by examining the previous state and A_H, and the emitted observation can be determined by examining B_H.

Forward Algorithm

To assess the similarity between the trained model and an observation sequence, the forward algorithm [80] is utilized. The forward algorithm calculates the likelihood that the observation data O could have been generated by the model.

$$\mathcal{P}(Q|\lambda) = \pi_1 a_{1,2} a_{2,3} \cdots a_{T-1,T} \tag{3.19}$$

$$\mathcal{P}(O|Q,\lambda) = b_1(O_1) b_2(O_2) \cdots b_T(O_T) \tag{3.20}$$

$$\mathcal{P}(O|\lambda) = \mathcal{P}(O|Q,\lambda)\mathcal{P}(Q|\lambda) \tag{3.21}$$

where Q is a given state sequence, and a and b are the probability of Q and O occurring, given λ, respectively. Thus, the forward algorithm determines the probability of a given state sequence given an HMM $\mathcal{P}(Q|\lambda)$ and the probability of this state sequence emitting a given observation sequence $\mathcal{P}(O|Q,\lambda)$ to determine the probability of that a given observation sequence could have been generated by a given HMM $\mathcal{P}(O|\lambda)$. Typically, the logarithm of $\mathcal{P}(O|\lambda)$ is reported, as the log-likelihood (LL) value is easier to work with than extremely small values that could be generated by $\mathcal{P}(O|\lambda)$, i.e. $LL = \log \mathcal{P}(O|\lambda)$. As an application of dynamic programming, the forward algorithm can become computationally heavy. It requires $\mathcal{O}(N^2T)$ calculations, where N is the number of states and T is the observation length [80].

Baum-Welch Algorithm

The Baum-Welch algorithm [80] is an expectation-maximization algorithm utilized to train the model in a two-step iterative fashion. The expected value E is calculated by taking the expectation of the log-likelihood of the observation O and state sequence Q given the current model estimate λ_n, $\log \mathcal{P}(O,Q|\lambda_n)$:

$$E(\lambda_n|\lambda_{n-1}) = \mathcal{E}\left[\log \mathcal{P}(O,Q|\lambda_n)|O,\lambda_{n-1}\right] \tag{3.22}$$

The maximization step computes:

$$\frac{\partial \log \mathcal{P}(O|\lambda)}{\partial \lambda} = 0 \tag{3.23}$$

and solves for λ for the updated model parameters. These two steps are repeated until convergence. The Baum-Welch algorithm only finds a local maximum, so the training initialization λ_1 is important.

Bayesian Information Criterion

A HMM's ability to represent, identify and reproduce a motion primitive is dependent on its model initialization and the number of states used. Although hidden, the state often represents some physical significance to the model, such as key poses for motion capture, or phonemes for speech analysis. A model with a small number of states may not carry enough resolution to fully model the primitive. A model with a large number of states has heavy computational costs and may cause the HMM to become overfitted [80]. The

Bayesian Information Criterion (BIC) can be used to balance between model accuracy and high state cost:

$$BIC = -2LL + N_s log(T_{obs}) \tag{3.24}$$

where N_s is the number of states used for this model, and T_{obs} is the number of observations used. Plotting the BIC against N_s shows that the LL improvement per state added decreases exponentially, so a large number of states will add a large computational cost, but may yield only marginal improvement in the HMM's ability to model the template data.

Chapter 4

Pose Estimation

This chapter develops an algorithm for on-line joint estimation from on-body IMU sensors. The joint angle estimation is achieved by estimating joint angles from linear acceleration and angular velocity measurements via EKF and a kinematic model of the human body. In order to minimize drift, angle constraints are applied to the EKF state estimate. An automated approach is also developed for estimating and adapting the process noise during on-line estimation[1].

4.1 Kinematic Modeling

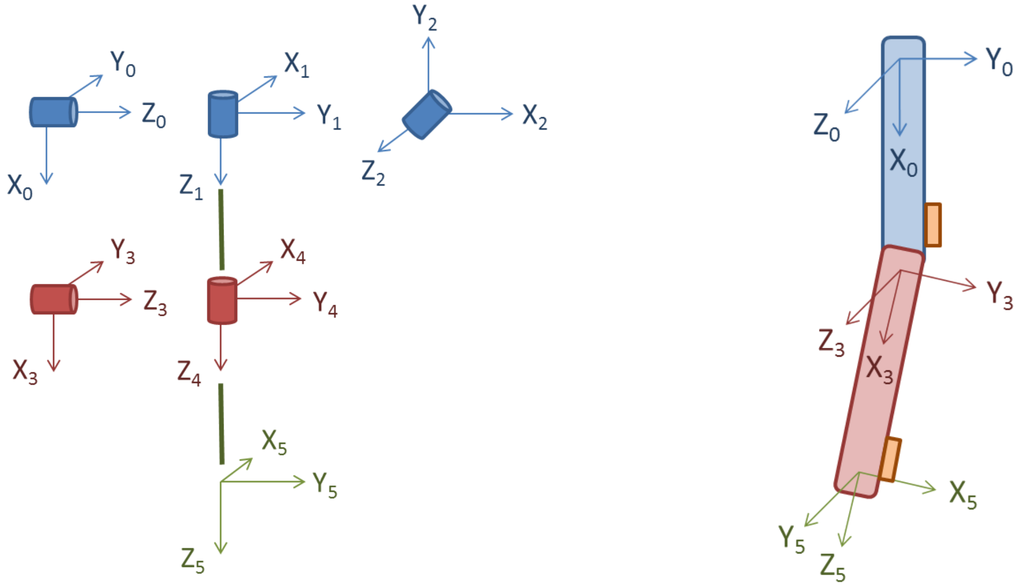
To derive a kinematic model of the lower body, the human leg is modeled as an articulated chain of rigid bodies, where each rigid body corresponds to a leg limb. The rigid bodies are connected through a set of joints (i.e., the hip, knee and ankle). To construct the kinematic model, a reference frame (denoted as the i^{th} frame) is attached to each DOF (denoted as the $i - 1^{th}$ joint) following the DH convention (Section 3.1.1). The DH parameters utilized are outlined in Table 4.1.

Using the DH convention shown in Figure 4.1, Frames 0, 1 and 2 are located at the hip, forming a 3 DOF spherical joint, and describe the motions of the thigh. This allows for hip flexion/extension, internal/external rotation, as well as abduction/adduction. $q_{1,2,3}$, $\dot{q}_{1,2,3}$ and $\ddot{q}_{1,2,3}$ describe the joint angles, angular velocities and angular accelerations of the 3 DOFs in the hip joint. The base frame is assumed to be stationary, $\boldsymbol{\ddot{x}_{e,0}} = \boldsymbol{\alpha_0^0} = \boldsymbol{\omega_0^0} = 0$, and thus $\boldsymbol{\omega_1^1} = R_{01}^T \boldsymbol{\dot{q}_1}$ and $\boldsymbol{\alpha_1^1} = R_{01}^T \boldsymbol{\ddot{q}_1}$. $\boldsymbol{\ddot{x}_{e,3}}$ describes the linear acceleration of the knee,

[1] This work has been published in *Physiological Measurement* [62].

Table 4.1: DH parameters for the EKF kinematic model

Link	$\boldsymbol{\theta}$	$\boldsymbol{d}$	$\boldsymbol{r}$	$\boldsymbol{\alpha}$
1	$q_1 + \pi/2$	0	0	$\pi/2$
2	$q_2 + \pi/2$	0	0	$-\pi/2$
3	$q_3 - \pi/2$	0	l_{upper}	$-\pi/2$
4	$q_4 + \pi/2$	0	0	$\pi/2$
5	q_5	l_{lower}	0	0



(a) The individual DH frames, with the subject in the standing position facing into the page.

(b) The DH frames on the leg, with the subject in the standing position facing right.

Figure 4.1: The DH frame assignment utilized in the EKF. The orange boxes represent the sensor packages. The hip is modeled as a 3 DOF joint, and the knee is modeled as a 2 DOF joint. Frame 0, 3 and 5 refer to the hip, knee and ankle starting frames, respectively. Frame 1, 2 and 4 correspond to intermediate hip and knee DOF frames.

and is measured by the knee accelerometer. $\boldsymbol{\omega_3^3}$ describes the angular velocity of the thigh, and is measured by the knee gyroscope. Frames 0 and 1 share the same origin, and thus have no link offset, $\boldsymbol{\ddot{x}_1} = \boldsymbol{\ddot{x}_2} = 0$.

Offset from the first 3 frames by the length of the thigh, Frames 3 and 4 form a 2 DOF joint at the knee, allowing for knee flexion/extension, as well as internal/external rotation. $q_{4,5}$, $\dot{q}_{4,5}$ and $\ddot{q}_{4,5}$ describe the joint angles, angular velocities and accelerations of the 2 DOFs in the knee joint. $\boldsymbol{\omega_5^5}$ and $\boldsymbol{\alpha_5^5}$ describe the total angular velocities/accelerations of the calf, including motion generated at the hip and knee joints. $\boldsymbol{\ddot{x}_{e,5}}$ describes the linear acceleration of the ankle, measured by the ankle accelerometer, whereas $\boldsymbol{\omega_5^5}$ describes the angular velocity of the calf, and is measured by ankle gyroscope.

Frame 5 is offset from Frames 3 and 4 by the length of the calf, and is the last frame of the kinematic chain. Homogeneous transformations are utilized to rotate between the DH frames and the local sensor frames.

The full set of equations are as follows:

$$\boldsymbol{\omega_1^1} = R_{01}^T \boldsymbol{\dot{q}_1} \tag{4.1}$$

$$\boldsymbol{\alpha_1^1} = R_{01}^T \boldsymbol{\ddot{q}_1} + \boldsymbol{\omega_1^1} \times (R_{01}^T \boldsymbol{\dot{q}_1}) \tag{4.2}$$

$$\boldsymbol{\ddot{x}_1} = 0 \text{ (due to no length between segment 1 and 2)} \tag{4.3}$$

$$\boldsymbol{\omega_2^2} = R_{12}^T \boldsymbol{\omega_1^1} + R_{12}^T \boldsymbol{\dot{q}_2} \tag{4.4}$$

$$\boldsymbol{\alpha_2^2} = R_{12}^T \boldsymbol{\alpha_1^1} + R_{12}^T \boldsymbol{\ddot{q}_2} + \boldsymbol{\omega_2^2} \times (R_{12}^T \boldsymbol{\dot{q}_2}) \tag{4.5}$$

$$\boldsymbol{\ddot{x}_2} = 0 \tag{4.6}$$

$$\boldsymbol{\omega_3^3} = R_{23}^T \boldsymbol{\omega_2^2} + R_{23}^T \boldsymbol{\dot{q}_3} \tag{4.7}$$

$$\boldsymbol{\alpha_3^3} = R_{23}^T \boldsymbol{\alpha_2^2} + R_{23}^T \boldsymbol{\ddot{q}_3} + \omega \times (R_{23}^T \boldsymbol{\dot{q}_3}) \tag{4.8}$$

$$\boldsymbol{\ddot{x}_3} = \boldsymbol{\alpha_3^3} \times \boldsymbol{r_{uLen}} + \boldsymbol{\omega_3^3} \times \boldsymbol{\omega_3^3} \times \boldsymbol{r_{uLen}} \tag{4.9}$$

$$\boldsymbol{\ddot{x}_{3s}} = \boldsymbol{\alpha_3^3} \times \boldsymbol{r_{uShim}} + \boldsymbol{\omega_3^3} \times \boldsymbol{\omega_3^3} \times \boldsymbol{r_{uShim}} \tag{4.10}$$

$$\boldsymbol{\omega_4^4} = R_{34}^T \boldsymbol{\omega_3^3} + R_{34}^T \boldsymbol{\dot{q}_4} \tag{4.11}$$

$$\boldsymbol{\alpha_4^4} = R_{34}^T \boldsymbol{\alpha_3^3} + R_{34}^T \boldsymbol{\ddot{q}_4} + \boldsymbol{\omega_4^4} \times (R_{34}^T \boldsymbol{\dot{q}_4}) \tag{4.12}$$

$$\boldsymbol{\ddot{x}_4} = R_{34}^T \boldsymbol{\ddot{x}_3} \tag{4.13}$$

$$\boldsymbol{\omega_5^5} = R_{45}^T \boldsymbol{\omega_4^4} + R_{45}^T \boldsymbol{\dot{q}_5} \tag{4.14}$$

$$\boldsymbol{\alpha_5^5} = R_{45}^T \boldsymbol{\alpha_4^4} + R_{45}^T \boldsymbol{\ddot{q}_5} + \boldsymbol{\omega_5^5} \times (R_{45}^T \boldsymbol{\dot{q}_5}) \tag{4.15}$$

$$\boldsymbol{\ddot{x}_{5s}} = R_{45}^T \boldsymbol{\ddot{x}_4} + \boldsymbol{\alpha_5^5} \times \boldsymbol{r_{lShim}} + \boldsymbol{\omega_5^5} \times \boldsymbol{\omega_5^5} \times \boldsymbol{r_{lShim}} \tag{4.16}$$

$$R_{03e} = R_{01} R_{12} R_{23} R_{tt1} \tag{4.17}$$

$$R_{05e} = R_{01} R_{12} R_{23} R_{34} R_{45} R_{tt2} \tag{4.18}$$

$$\boldsymbol{\omega_{3s}} = R_{tt1}^T \boldsymbol{\omega_3^3} \tag{4.19}$$

$$\boldsymbol{\ddot{x}_{3g}} = R_{tt1}^T \boldsymbol{\ddot{x}_{3s}} + R_{03e}^T \boldsymbol{g} \tag{4.20}$$

$$\boldsymbol{\omega_{5s}} = R_{tt2}^T \boldsymbol{\omega_5^5} \tag{4.21}$$

$$\boldsymbol{\ddot{x}_{5g}} = R_{tt2}^T \boldsymbol{\ddot{x}_{5s}} + R_{05e}^T \boldsymbol{g} \tag{4.22}$$

$$\boldsymbol{x_{e,upper}^0} = R_{03e} \boldsymbol{r_{uShim}} \tag{4.23}$$

$$\boldsymbol{x_{e,lower}^0} = R_{03} \boldsymbol{r_{uLen}} + R_{05e} \boldsymbol{r_{lShim}} \tag{4.24}$$

where $\boldsymbol{\ddot{x}_{3g}}$, $\boldsymbol{\ddot{x}_{5g}}$, $\boldsymbol{\omega_{3s}}$ and $\boldsymbol{\omega_{5s}}$ denote the accelerometer and gyroscope values for the thigh and calf IMU sensors. $\boldsymbol{x_{e,upper}^0}$ and $\boldsymbol{x_{e,lower}^0}$ are the Cartesian coordinates of the knee and ankle respectively. $\boldsymbol{r_{uShim}}$ is the vector from the hip origin to the location of the sensor on the thigh. $\boldsymbol{r_{uLen}}$ is the vector from the hip to the knee. $\boldsymbol{r_{lShim}}$ is the vector from the knee to the sensor on the calf.

For simplicity, Frame 0 is assumed to be stationary. For motions where the foot is not moving, such as sit-to-stand, the kinematic chain is reversed, such that the ankle is the 3 DOF joint, and the knee remains as the 2 DOF joint. Although the current formulation requires a stationary base frame, the proposed algorithm can be extended to motions that do not have a fixed base, by accounting for the base frame motion in the state vector. This is a common technique in the robotics field, and has been applied to the analysis of mobile manipulators. The motion of the base can be estimated within the same Kalman filter framework by incorporating data from a sensor that experiences only the base frame movement, such as a sensor located at the waist.

Modeling internal/external rotation at both joints, particularly since the knee does not have significant range of motion in this direction, allows for increased flexibility for sensor placements. It is expected that the sensors would be mounted on the front of the subject, but poorly placed sensors can be modeled by including a non-zero initial internal/external rotation (q_2, q_5).

4.2 Extended Kalman Filtering

The goal of the proposed approach is to estimate the joint angles $\boldsymbol{q}$ based on the measurements obtained from the accelerometers and gyroscopes, $\ddot{\boldsymbol{x}}_{\boldsymbol{3g}}$, $\ddot{\boldsymbol{x}}_{\boldsymbol{5g}}$, $\boldsymbol{\omega_{3s}}$ and $\boldsymbol{\omega_{5s}}$ using the EKF. As described in Section 3.2.2, the EKF formulation assumes that process and measurement noise is Gaussian distributed with zero mean. It should be noted that human motion variability may not be Gaussian, particularly at higher velocities. However, for motions performed at slow to normal velocities, such as rehabilitation exercises, noise characteristics can be approximated by the Gaussian distribution.

The system state consists of the joint angles, velocities and accelerations, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. The following kinematic motion equations are used as the process (state) model:

$$\boldsymbol{q_t} = \boldsymbol{q_{t-1}} + \dot{\boldsymbol{q}}\Delta t + \ddot{\boldsymbol{q}}\Delta t^2/2 \tag{4.25}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{t}} = \dot{\boldsymbol{q}}_{\boldsymbol{t-1}} + \ddot{\boldsymbol{q}}\Delta t \tag{4.26}$$

$$\ddot{\boldsymbol{q}}_{\boldsymbol{t}} = \ddot{\boldsymbol{q}}_{\boldsymbol{t-1}} \tag{4.27}$$

This system model includes the evolution of position and velocity, while acceleration is modeled to remain constant. The unmodeled higher order terms (i.e., changes in acceleration) are accounted for through the process noise. Note that this modeling approach requires that the process noise be modified proportional to the motion speed, as the impact of the unmodeled higher order values would become more significant at higher speeds, and thus Q_t would need to be increased accordingly during the state estimation. The initial Q_t matrix is obtained by hand-tuning, as described in Section 6.2. Additional tuning of Q_t is performed automatically during the EKF estimation, to adjust the EKF to the motion that is being estimated. The automatic tuning is described in Section 4.3.

Since all the equations are linear, the A matrix is:

$$A_t = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.28}$$

The process noise Jacobian W_t is a diagonal matrix, with 0 for q, and 1 for $\dot{q}$ and $\ddot{q}$. This effectively models the kinematic system as a constant-velocity model, as noise is being injected in both $\dot{q}$ and $\ddot{q}$. The constant-velocity model was adopted after experimentation with a constant-acceleration model, where noise is only injected into $\ddot{q}$, yielded poor results.

$$W_t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.29}$$

Equations 4.19, 4.20, 4.21 and 4.22 are used as the measurement equations, which are non-linear. They are linearized as described in Section 3.2.3 and utilized in the extended Kalman filter, as the C_t matrix.

The measurement noise Jacobian V_t is a diagonal matrix, as all sensors contribute noise:

$$V_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.30}$$

4.3 Adapation of the Noise Parameters

Poor selection of the process noise $\boldsymbol{w_t}$ can significantly impact the performance of the filter. If it is too small, the higher order terms that are not modeled in the A matrix cause improper state estimation. If it is too high, too much noise is injected into the state estimate. Both of these situations could affect the state and state variance estimates, causing the EKF intermediate matrices to become poorly conditioned. In particular, the calculation of the Kalman gain in Equation 3.18 requires the inversion of the intermediate noise covariance matrix P_y (Equation 3.17). The measurement equation C_t is a function of the state. Thus, if a poor $\boldsymbol{w_t}$ selection leads to poor state recovery, P_y can become poorly conditioned. The matrix inversion of a poorly conditioned matrix causes numerical errors, and can cause the EKF algorithm to diverge.

Since the goal is to develop an on-line estimation algorithm, an approach to automatically detect poor conditioning in the covariance matrix and prevent EKF divergence is needed. When P_y is approaching a singularity, the filter state is reset to the initial values, since a near-singular P_y matrix can introduce erroneous results into the states. After the filter has been reset, the process covariance is increased to reduce the chances of future singularities. The initial process noise parameters can be selected to be low, and be increased gradually by this tuning process until the proper process noise is found.

It is, however, difficult to set a threshold $\epsilon_{divThres}$ for the poorly conditioned matrix. The condition number of the P_y matrix fluctuates based on the state estimates, and can naturally become high and low. Since it is undesirable to accidentally reset the filter or change the noise profile while the filter is working as expected, a high value for $\epsilon_{divThres}$ was set, so that the auto-tuning code would only be triggered right before P_y becomes singular. As a result, by the time the condition number of P_y exceeds the threshold, the recovered angles are likely to have been inaccurately estimated for several timesteps. A timestep rewind is applied so that the updated filter will overwrite these poorly estimated states.

```
if $\text{cond}(P_y) \geq \epsilon_{divThres} \,\|\, q \gg 3\boldsymbol{q_c}$ then
    Rewind timesteps
    Reset filter state
    Increase process noise $w$
end if
```

Figure 4.2: Process noise auto-tune algorithm.

It may also be possible that the joint angles become exceedingly large before the condition numbers are overtly impacted. Therefore, if a given joint exceeds its corresponding joint limit, $q_{c,i}$, by a factor, the filter is also reset. The factor is arbitrary set to 3 such that this filter reset will not activate prematurely. This process is summarized in Figure 4.2.

Poor selection of measurement noise $\boldsymbol{v_t}$ can also impact filter performance. If $\boldsymbol{v_t}$ is too small, the measurement equations cannot adequately account for the amount of noise that is in the sensor data, causing integrational drift. If $\boldsymbol{v_t}$ is too high, motion data may be lost as it is being discarded by the EKF as noise. However, unlike the process noise, where poor selection could cause singularities, it is difficult to determine if the signal is drifting or if the movement is natural. However, natural motion is constrained by the human joint angle ranges, and cannot exceed them, whereas a drifting signal would grow without bound.

4.4 Kinematic Constraints

To incorporate knowledge of the range of motion constraints to reduce drift, the estimated state angles are checked at each timestep to ensure that they are within realistic bounds, as illustrated in Figure 4.3. A potential field [51] is applied to push $\boldsymbol{q}$ away from joint bounds. The potential field generates a virtual acceleration pushing the joint away from the joint boundary whenever the joint is close to the joint limit. Given

$$\underline{\rho}_i = q_i - \underline{q}_{c,i} \tag{4.31}$$

$$\overline{\rho}_i = \overline{q}_{c,i} - q_i \tag{4.32}$$

the potential field acceleration γ_i is calculated by

$$\gamma_i = \begin{cases} \eta \left(\frac{1}{\underline{\rho}_i} - \frac{1}{\underline{\rho}_{0,i}} \right) \frac{1}{\underline{\rho}_i^2} & \text{if } \underline{\rho}_i \leq \underline{\rho}_{0,i} \\ -\eta \left(\frac{1}{\overline{\rho}_i} - \frac{1}{\overline{\rho}_{0,i}} \right) \frac{1}{\overline{\rho}^2} & \text{if } \overline{\rho}_i \leq \overline{\rho}_{0,i} \\ 0 & \quad\quad \text{otherwise} \end{cases}$$

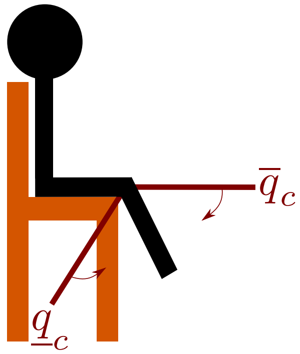


Figure 4.3: Subject moving within joint angle constraints q_c. When the joint angle estimation approaches the constraint bounds, an artificial acceleration is applied against the joint angle to push the joint angle estimate back into expected ranges.

where $\underline{q}_{c,i}$ and $\overline{q}_{c,i}$ are the lower and upper bound on the joint q_i, $\underline{\rho}_{0,i}$ and $\overline{\rho}_{0,i}$ are the distance limits of the potential field influence, and η is the potential field coefficient. As q_i approaches the joint constraint $q_{c,i}$, γ_i applies an increasingly large acceleration in the opposite direction to push the joint back within the acceptable range. The pseudocode for this process is described in Figure 4.4.

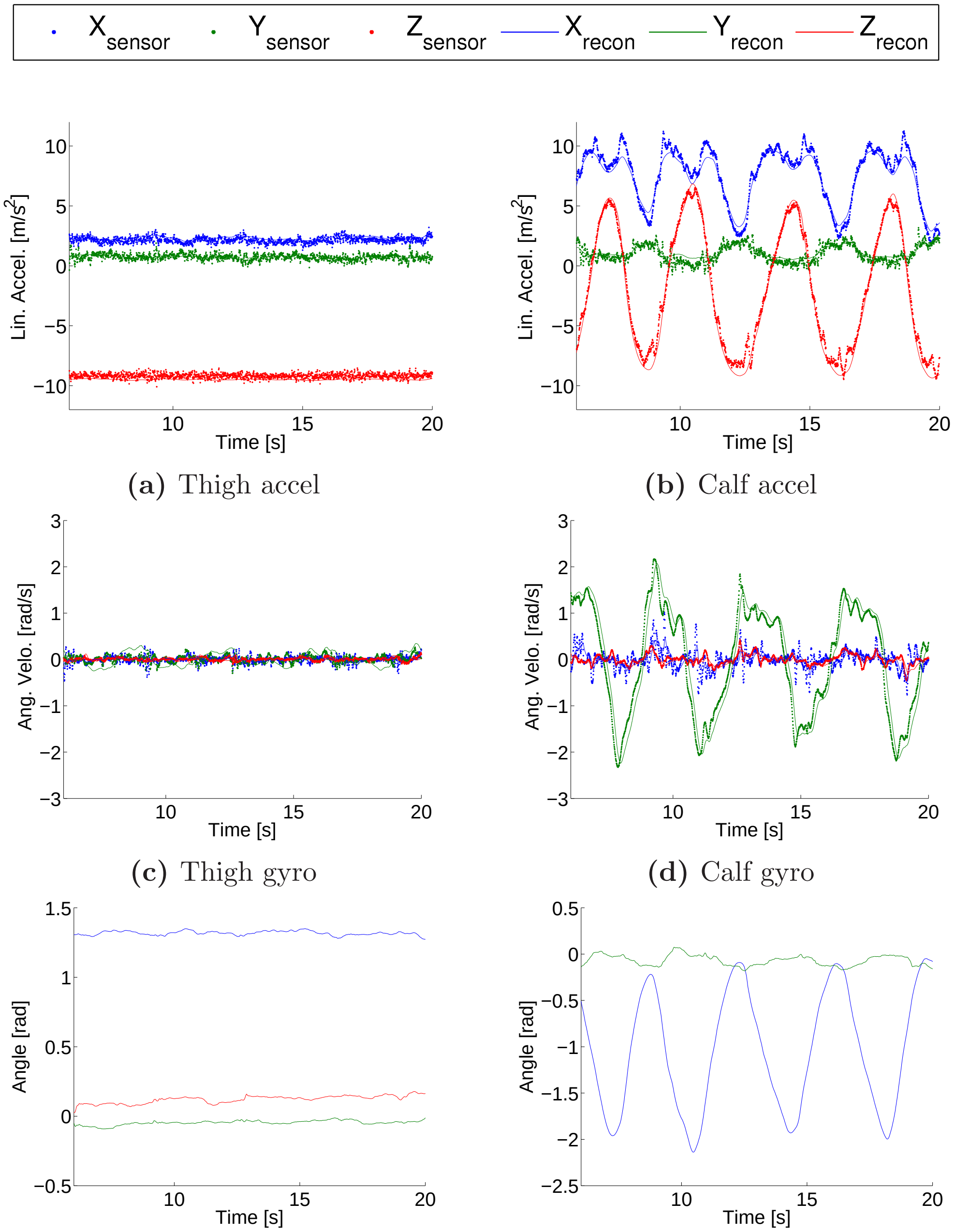
The proposed joint limit algorithm can be used to constrain the joint angle estimates to anthropometrically feasible values in the case of arbitrary 3D motion. If additional information is available about the motion to be performed, the joint limits can be modified to include this additional information. For example, for the case of motion in the sagittal plane only, joint angle limits for out of plane joints can be decreased to incorporate this additional information into the filter and further reduce drift error. Since the potential field runs on top of the EKF, it introduces additional calculations at each timestep. This increased complexity is not significant, as this implementation of the potential field does not involve computationally heavy components.

4.5 Examples

Section 6.2 describes the experimental validation of the proposed algorithm on a motion capture dataset. Figure 4.5 shows an example of knee extensions while seated. Figure 4.6 shows an example of knee-hip flexions while supine. The thick lines in the figures denote the end-effector position ground truth from motion capture, whereas the thin lines denote the estimated end-effector position from the EKF algorithm.

```
for $i = 1 \to 5$ do
    $\overline{\rho}_i \leftarrow \overline{q}_{c,i} - q_i$
    $\underline{\rho}_i \leftarrow q_i - \underline{q}_{c,i}$
    if $\overline{\rho}_i \leq \overline{\rho}_{0,i}$ then
        $\ddot{q}_{t,i} \leftarrow -\eta(1/\overline{\rho}_i - 1\overline{\rho}_{0,i})/\overline{\rho}_i^2$
    else if $\underline{\rho} \leq \underline{\rho}_0$ then
        $\ddot{q}_{t,i} \leftarrow -\eta(1/\underline{\rho}_i - 1\underline{\rho}_{0,i})/\underline{\rho}_i^2$
    else
        $\ddot{q}_{t,i} \leftarrow A\ddot{q}_{t-1,i}$
    end if
end for
```

Figure 4.4: Kinematic constraining algorithm.



(a) Thigh accel

(b) Calf accel

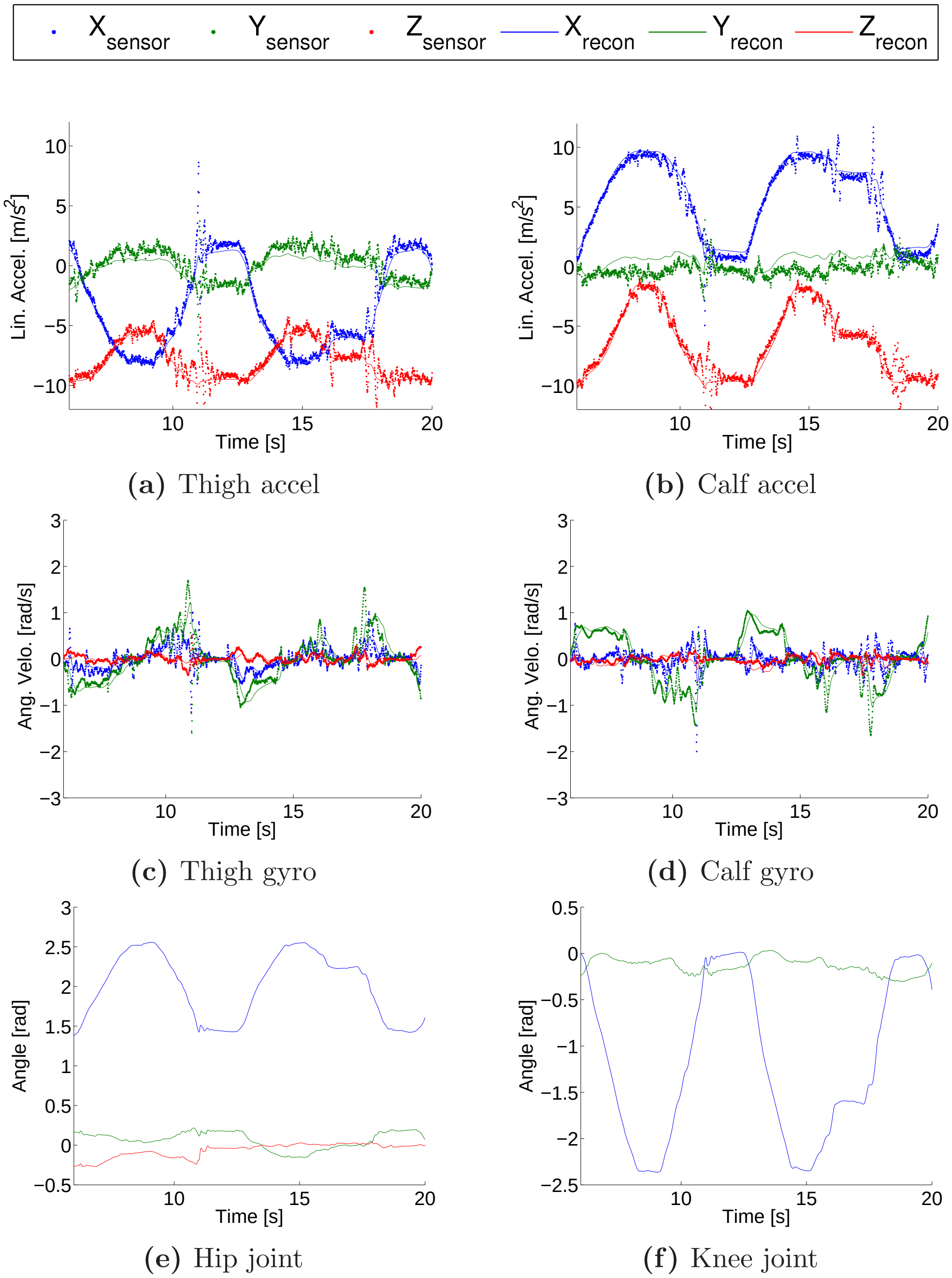
(c) Thigh gyro

(d) Calf gyro

(e) Hip joint

(f) Knee joint

Figure 4.5: Linear acceleration, angular velocity and joint angle as measured from sensors (thick) and from joint estimation (thin) of leg extension motion.



(a) Thigh accel

(b) Calf accel

(c) Thigh gyro

(d) Calf gyro

(e) Hip joint

(f) Knee joint

Figure 4.6: Linear acceleration, angular velocity and joint angle as measured from sensors (thick) and from joint estimation (thin) of knee-hip flexion motion.

Chapter 5

Segmentation and Identification

In this chapter, a template-based on-line segmentation technique based on both velocity features and probabilistic templates is proposed. The system consists of a training phase, where exemplar data are used to create feature and HMM motion templates, and a segmentation phase, where the observed data is swept for characteristic features that match the feature templates. When a segment candidate is found, the HMM templates are used to identify the motion and confirm the exact segment point locations[1].

5.1 Modeling Motion Segments

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

The Baum-Welch algorithm [80] (Section 3.3.1) is utilized to train the template model. Testing with the collected dataset (Section 6.1) has shown that 6 to 10 state HMMs result in similar levels of segmentation accuracy, similar to previous findings for human motion

[1] This work has been submitted for publication in *IEEE Transactions on Neural Systems and Rehabilitation Engineering* [63].

primitive modeling [10]. An 8-state system was selected as the optimal compromise between the time needed to perform template training and segmentation accuracy.

To assess the similarity between a trained model and a new observation sequence, the forward algorithm [80] (see Section 3.3.1) is applied. The forward algorithm calculates the likelihood that the observation data could have been generated by the model. In order to compare the continuous time series data against the template, a candidate section must be selected. A simple sliding window of fixed length W is not optimal, since intra- and interpersonal differences may result in a different time taken to complete each motion. A second alternative is to test a set of windows of variable size, $W + nW_{ext}$, where n is a multiplier on the fixed-window series, and W_{ext} is the length of addition to the observation window added. Both windowing methods are computationally expensive as they require the forward algorithm to be applied $T - W$ times for a single fixed window, or $(T - W) \times n$ for multiple fixed-windows for each HMM template.

The forward algorithm also implicitly penalizes longer observations. Since the forward algorithm calculates the likelihood via dynamic programming, and the values examined are probability values (hence a value between 0 and 1), the longer the state sequence, the smaller the probability becomes [80]. To account for this, the likelihood measure can be normalized by the length of observation data, or the observation data can be resampled so that all data passed into the forward algorithm are of the same length. The latter approach is adopted herein.

5.2 Feature Extraction and Template Training

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

It is assumed that ZVCs start and terminate any given primitive [79]. In this algorithm, a ZVC is defined as one of the following: 1) the velocity crosses from positive to negative, 2) the velocity crosses from negative into positive, or 3) the absolute mean value of the velocity over a small window is below a threshold [30]. Using velocity features allows the algorithm to locate the general shape of the motion of interest more robustly then algorithms that rely on distance measures [48].

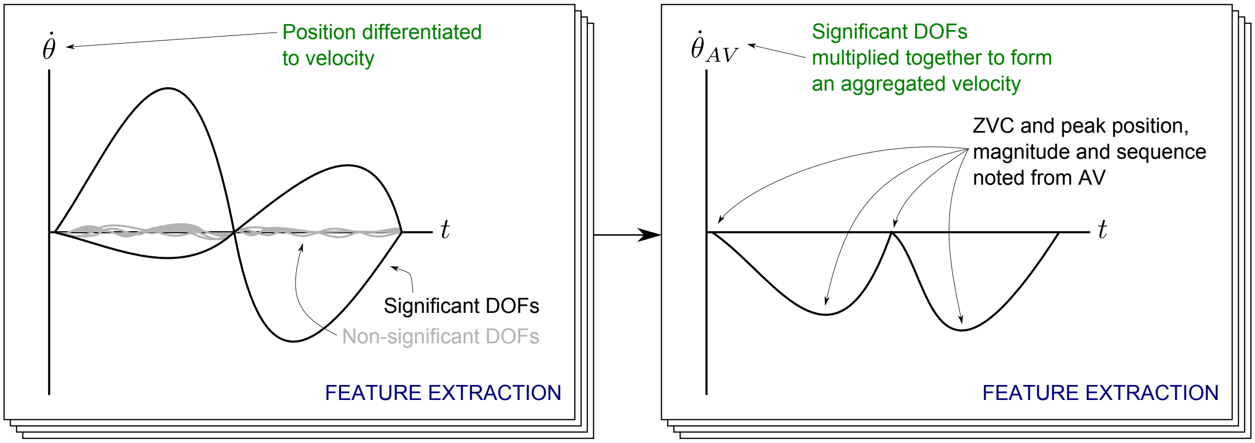

$\dot{\theta}$
Position differentiated to velocity
t
Significant DOFs
Non-significant DOFs
FEATURE EXTRACTION
$\dot{\theta}_{AV}$
Significant DOFs multiplied together to form an aggregated velocity
ZVC and peak position, magnitude and sequence noted from AV
t
FEATURE EXTRACTION

Figure 5.1: Feature component of the segmentation training procedure. The training algorithm extracts significant DOFs from velocity variances from the provided exemplars, and multiplies these velocities together to form an aggregated velocity. Velocity features (ZVC, velocity peaks) are extracted from the aggregated velocity.

Human motions, particularly typical rehabilitation exercises, are characterized by periodic velocity patterns, such as flexion and extension, making velocity peaks and zero crossings a salient indicating feature. A typical feature template of a rehabilitation motion would consists of a ZVC, then a positive or negative peak, then another ZVC, the opposite peak, and a final ZVC. The feature extraction and matching is based on the assumption that such a velocity profile sequence can be extracted and used in the template exemplars and observations. The template training procedure is summarized in graphical form in Figure 5.1 and 5.2; pseudocode is provided in Figure 5.3.

HMM template training (Figure 5.3.5-6)

The exemplars are used to train HMMs with the Baum-Welch algorithm [80], creating an HMM for each primitive. An 8-state left-right model is used. The Gaussian observation functions are initialized by k-means clustering. LL threshold values LL_{thres}, used to recognize template motions in the observation data, are derived by taking the average of the LL values of the training exemplars evaluated on the HMM, shifted by a scaling factor.

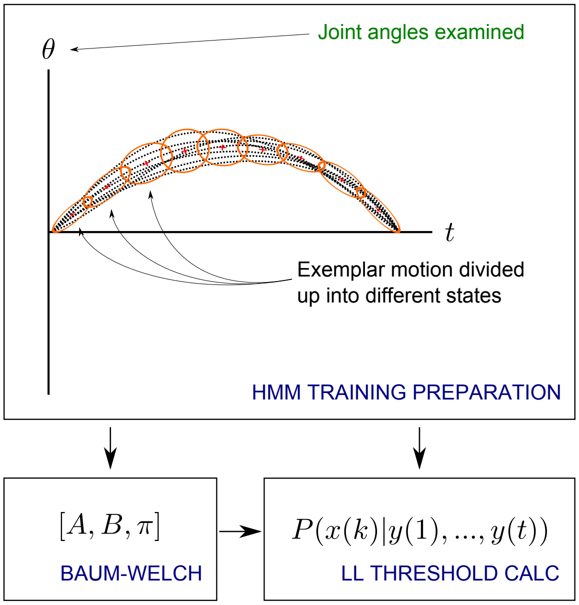


Figure 5.2: HMM component of the segmentation training procedure. The training algorithm uses means (cross) and variances (oval) of equally spaced windows as the initial values for the Baum-Welch HMM training algorithm. The exemplar motions are not actually normalized to the same length, but the window width used is adjusted for each exemplar, and thus achieves similar effects. Normalized exemplars are shown here for ease of view. The LL threshold is calculated from running the exemplars through the trained HMM.

```
 1: for all Motion templates do
 2:     for all Exemplar data do
 3:         Filter data via Butterworth filter
 4:     end for
 5:     Train HMM via Baum-Welch with exemplar data
 6:     Calculate LL threshold with exemplar data
 7:     Calculate significant DOFs via $k$-means with exemplar velocities
 8:     Compute AV from significant DOFs
 9:     for all Exemplar data AV sequences do
10:         Locate all ZVCs in AV sequence
11:         Check peak values between ZVCs
12:         Discard small peaks
13:         Note all peak signs, magnitudes and times
14:         Peak signs sequence stored for feature matching
15:     end for
16:     Select majority peak signs sequence
17: end for
```

Figure 5.3: Template training algorithm pseudocode.

Significant DOFs (Figure 5.3.7)

In order to improve the proposed algorithm's robustness in the presence of high dimensionality data, a DOF feature selection routine is implemented. Rehabilitation motion tends to be focused on improving the range of motion of the injured joints, thus it can be assumed that the joints undergoing the largest range of motion are the significant ones. The significant DOFs of a given template are selected by calculating the standard deviations of the template joint velocities and grouping them via k-means clustering, with $k = 2$. The DOFs that are in the cluster with the highest centroid are assumed to be significant for that particular template.

Velocity aggregation (Figure 5.3.8)

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

Velocity feature extraction (Figure 5.3.9-16)

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

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

5.3 Segmentation and Recognition

Once the templates are prepared, the algorithm can be used to segment observation data. This process is outlined in graphical form in Figure 5.4; pseudocode is provided in Figure 5.5.

Segmentation preparation (Figure 5.5.2-3)

During the on-line segmentation phase, a small sliding window is passed over the observation data, noting the local peak values and ZVCs of each of the DOFs. An AV stream is computed for each template, based on the template's significant DOFs.

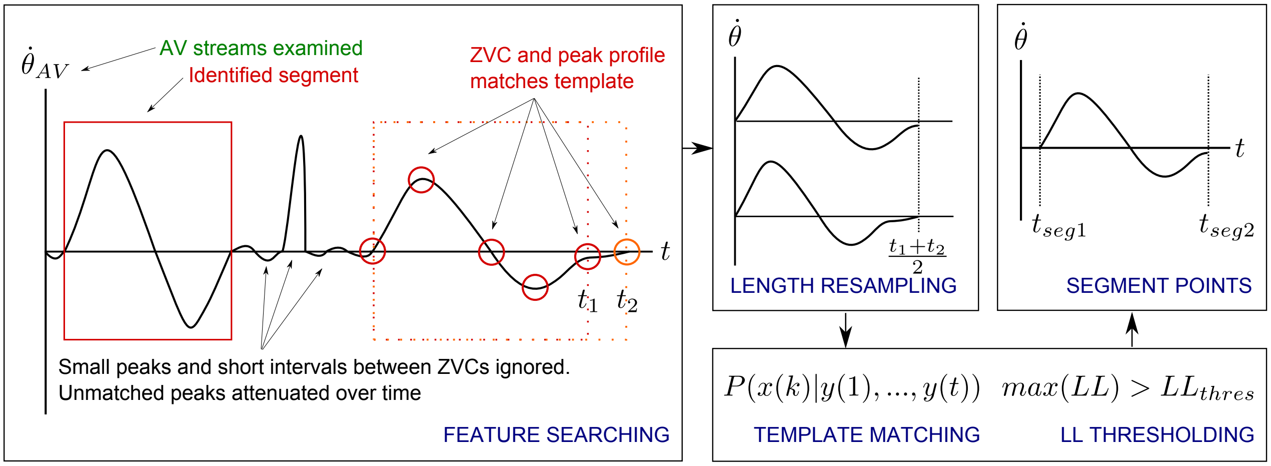


Figure 5.4: Segmentation algorithm. The observation velocity streams are converted into AV streams based on each template's significant DOFs. Velocity features are accepted or rejected based on the peak size or peak-to-peak interval data collected from the exemplars. Some subjects gradually slow to a stop, so low velocity for a period of time is also considered as a ZVC. Several ZVCs are collected to form windows. The windows are resampled to be equal length. Forward algorithm is used to assess the LL, which is used to determine which window is a segment.

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

Figure 5.5: Segmentation algorithm pseudocode.

Feature searching (Figure 5.5.2-3)

A ZVC is declared if the AV makes a zero-crossing, or if is very low for several timesteps [30]. Local peak values are tracked by an internal buffer. If the current window has a peak value higher than the stored peak value, the peak value in the buffer is updated accordingly. To avoid noise spikes in the velocity data from affecting the template matching, the peak buffer value is attenuated if it does not contribute to a match after several seconds, to prevent a large spike in the velocity from preventing feature matches.

If a given AV observes a sequence of ZVCs and peaks that matches its corresponding template, then the algorithm has located a potential segment point. Using features to estimate the start and end time of a segment candidate also accounts for temporal variations. This way, the general shape is always matched, regardless of the time it took the subject to perform the motion. The velocity magnitudes and peak-to-peak distance must exceed v_p and t_{pp} respectively in order for a potential segment point to be declared. This prevents noise, such as when the subject is stationary, from triggering the feature match.

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

HMM matching (Figure 5.5.8-10)

The template and window edge combination that results in the highest likelihood value over the threshold LL_{thres} is declared a segment. All window edge combinations are resampled so they are of equal length, to prevent the forward algorithm from favouring shorter sequences. Following HMM template matching, the recorded peak magnitude and ZVCs are reset, and the feature search resumes at the next time step.

5.4 Examples

Segmentation Example

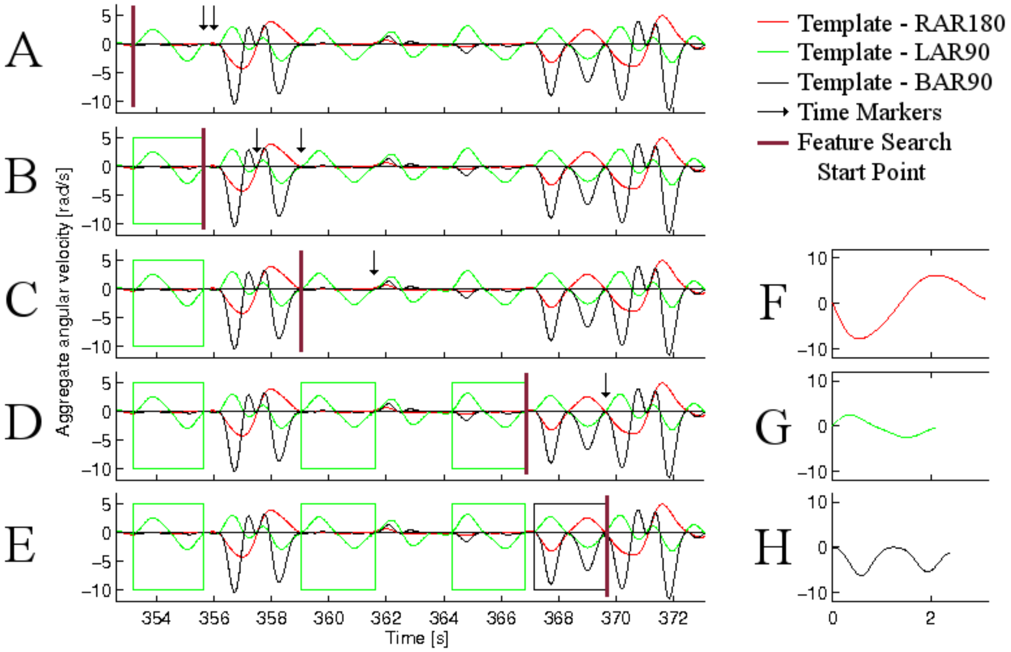


Figure 5.6: Illustration of on-line algorithm. The data shown from [57].

Figure 5.6 shows a step-through of the segmentation procedure. Horizontal bars indicate manual segmentation markers. All data shown are aggregate velocity data streams, for 180° right arm raise and lower, consisting of a right shoulder DOF (red signal, plot F, 'RAR180'), 90° left arm raise and lower, consisting of a left shoulder DOF (green signal, plot G, 'LAR90') and 90° both arm raise and lower, consisting of an aggregate velocity of a right and a left shoulder DOF (black signal, plot H, 'BAR90'). When the template features (plot F, G and H) are found in the observation data; that is, when the template pattern and the colour match a segment in the observation, then a potential segment is found. The procedure is as follows:

1. Red vertical bar marks the start of feature searching. The observation window is expanded until a potential template match (green signal, plot A) is found. Two end points are identified (black arrows, plot A).

2. Forward algorithm is ran on the two identified windows. The forward algorithm reports that the red line to first black arrow has a higher LL, as a right arm raise (RAR180), so a segment is declared (green box, plot B). The feature cache is cleared, and the start of the feature search is set to the end of the latest segment (red bar, plot B). The algorithm restarts. The first black arrow shows a similar RAR peak pattern as before, but the negative peak is below threshold, so the match was correctly deemed unsuccessful. The feature search continues to the second black arrow. At this point, large peaks were detected in the LAR90 (red signal, plot B) and BAR90 (black signal, plot B) signals, so their respective HMMs are tested.

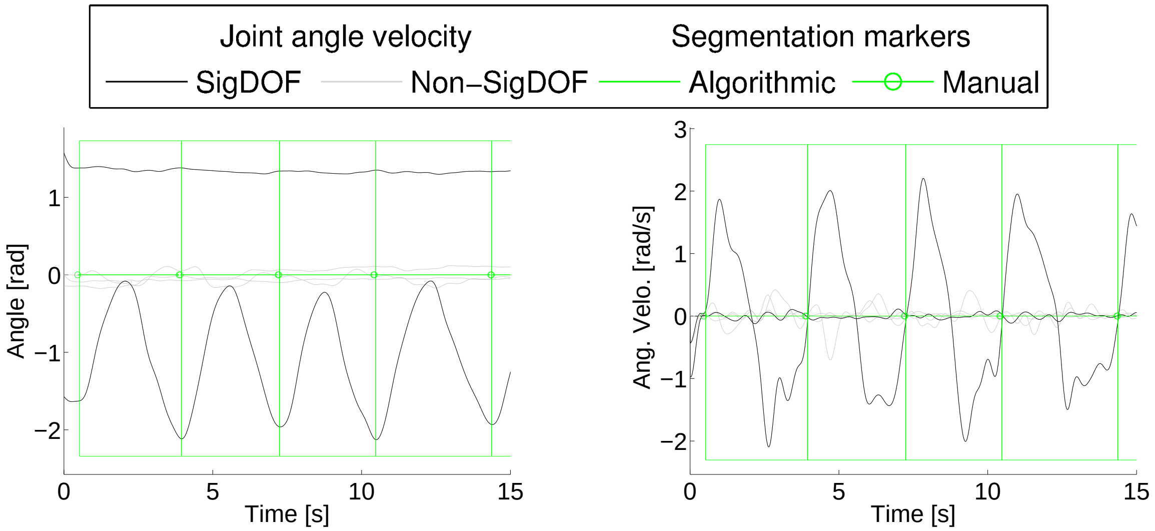
3. However, no match was made (the motion was BAR180, which was not one of the templates). The feature search start point is updated again (plot C), and the feature searching continues.

4. Two more successful segments are matched. At this point, once again, large peaks are found (BAR90, plot D), so relevant HMMs are tested.

5. A match is found for both arm raise (BAR, plot E). All four segments in this set have been successfully identified.

Segmentation Results

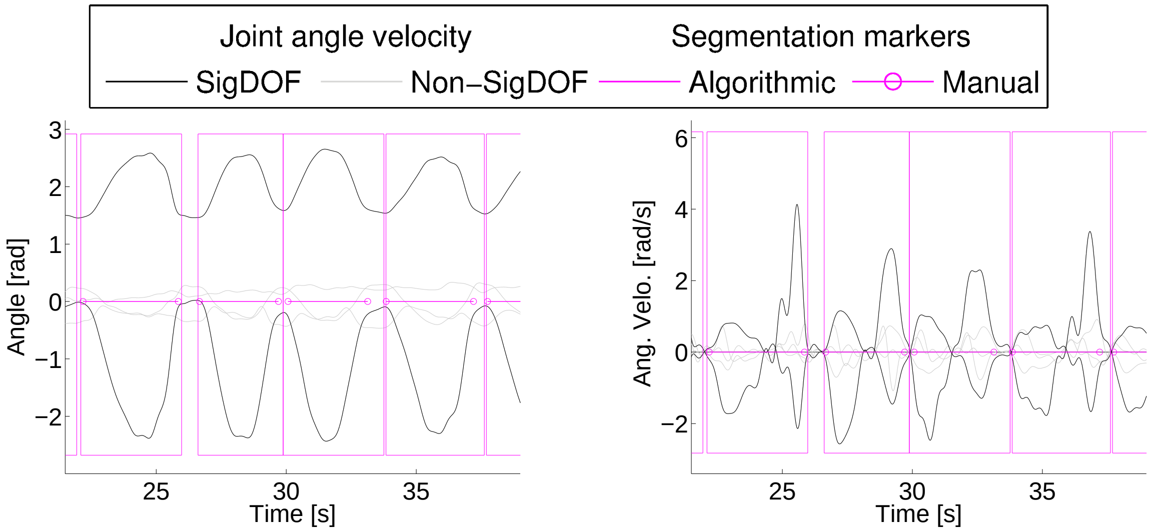
A detailed verification of the proposed algorithm is described in Section 6.3. Figure 5.7 shows an example of a segmentation result for data of a participant performing of knee extensions while seated. Figure 5.8 shows an example of knee-hip flexions while supine. The circles in the figures denote the manual segment points, whereas the boxes denote the algorithmic segments.



(a) Joint angle.

(b) Angular velocity.

Figure 5.7: Joint angles and angular velocity of a subject performing knee extensions. The algorithm segments properly when there are no pauses between each repetition.



(a) Joint angle.

(b) Angular velocity.

Figure 5.8: Joint angles and angular velocity of a subject performing knee-hip flexion. Although the participant's motions introduced spurious ZVCs in the middle of the motion, the algorithm was able to properly segment and identify the motion being performed.

Chapter 6

Experiments

The algorithms proposed in this thesis were verified on rehabilitation movements of healthy subjects. IMU data was collected and converted to joint angles using the pose recovery algorithm described in Chapter 4. The joint angle data was segmented and identified using the segmentation algorithm described in Chapter 5. This chapter describes the data collection process, as well as results from the joint angle estimation and the segmentation algorithms. All signal processing, algorithmic implementation and error calculations were programmed in MATLAB 7.12.

6.1 Data Collection

6.1.1 Inertial Measurement Units

The data was collected using the Shimmer IMU [14] sensor suite[1]. Shimmer sensors are small and lightweight, with a physical footprint of 53 mm by 32 mm by 15 mm, and weighing 22 grams. This sensor suite collects linear acceleration and angular velocity data, and transmits wirelessly via Bluetooth at 128 Hz. Each sensor package contains a Freescale MMA7361L 3D accelerometer, two InvenSense 2D IDG-500 gyroscopes mounted perpendicularly to each other to obtain 3D sensing, and a Roving Networks RN-42 Bluetooth radio. Figure 6.1 shows a detailed breakdown of the Shimmer components.

IMU sensors were favoured over line-of-sight systems such as cameras as IMUs do not suffer from occlusion, and can be deployed in a wide variety of different environments.

[1] Shimmer Research, www.shimmer-research.com

This is important as physiotherapy clinics tend to have cluttered environments obstructed by exercise equipment and patients can move around from station to station. A camera system would require a large number of cameras or be mobile to follow the patients, but these conditions are impractical.

IMU straps may interfere with free movement. During the data collection process, the subjects generally noted that the straps were tight, but do not seem to interfere with range and velocity of motion. Although the IMU straps did not interfere with the movements of healthy subjects, they may hamper movement for rehabilitation subjects.

Three Shimmer devices were used, placed on 1) the hip, along the height of the anterior superior iliac spine (ASIS, a major bony landmark on the hip), at the mid-sagittal region of the subject, 2) on the thigh near the knee, and 3) on the calf, near the ankle. Figure 6.2 shows the placement of the Shimmer sensors.

A calibration application based on Ferraris *et al.* [28] was provided by Shimmer Research, and was used to calibrate the accelerometer and gyroscope sensors. Ferraris' method was chosen as it is easy to apply, and does not require specialized equipment such as a turntable. The accelerometer alignment R_a, sensitivity K_a and bias $\boldsymbol{b_a}$ are calculated by placing the sensor on a level surface in the six different possible orientations and applying a linear regression to find the scaling parameters. The calibration equation is $\boldsymbol{a_{cal}} = (R_a)^{-1}(K_a)^{-1}(\boldsymbol{u_a} - \boldsymbol{b_a})$, where $\boldsymbol{a_{cal}}$ is acceleration in $[m/s^2]$ and $\boldsymbol{u_a}$ is the measured acceleration, in $[ADC]$.

The gyroscope alignment R_ω, sensitivity K_ω and bias $\boldsymbol{b_\omega}$ are obtained by holding the gyroscope still and then rotating the sensor 360° around its three axes. The calibration equation is $\boldsymbol{\omega_{cal}} = (R_\omega)^{-1}(K_\omega)^{-1}(\boldsymbol{u_\omega} - \boldsymbol{b_\omega})$, where $\boldsymbol{\omega_{cal}}$ is angular velocity in $[rad/s]$ and $\boldsymbol{u_\omega}$ is the measured angular velocity, in $[ADC]$.

6.1.2 Motion Capture

Data was also simultaneously collected on a Motion Analysis motion capture system[2] to obtain ground truth data. Eight Eagle cameras were used, with a sampling frequency of 60 Hz, and the Motion Analysis Cortex software was used for data collection. Markers were placed on the Shimmer sensors, the subject's shoulders, ASIS, right knee and ankle on both medial and lateral side, as well as the toe and heel. These markers were used to estimate the location of the Shimmer sensors and joint centers for joint angle verification. The position of the knee and ankle were obtained by taking the average of their respective

[2] Motion Analysis Corporation, www.motionanalysis.com

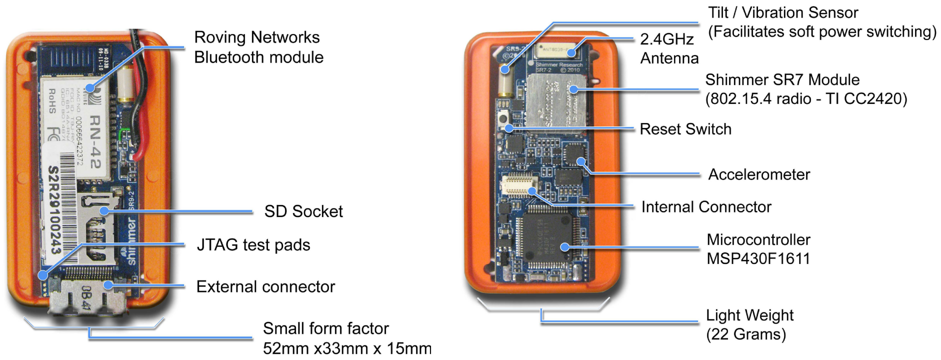


Figure 6.1: Labeled image of the Shimmer sensor. Battery and gyroscope modules not shown for brevity. Image used with permission from Shimmer Research.

medial and lateral positions. Spline interpolation was applied to fill in occluded markers and the data was swept for switched markers. The data streams produced by Shimmer and the Motion Analysis system were timestamped and time-aligned in post-processing. Figure 6.3 shows the marker placements.

6.1.3 Demographics and Data Collection Procedure

The algorithm was tested on healthy subjects performing typical knee and hip rehabilitation movements. The movement data of 20 participants (12 M, 8 F) were collected, with the average participant being 23 ± 4.5 years old. No participant had any lower back or leg injuries in the past six months. The experiment was approved by the University of Waterloo Research Ethics Board, and signed consent was obtained from all participants. Participant anthropometric data are summarized in Table 6.1, and are derived from motion capture data.

The subjects performed a series of exercises, as described in Table 6.2, designed to model physiotherapy exercises. Each participant performed two sets of ten motions for each motion type.

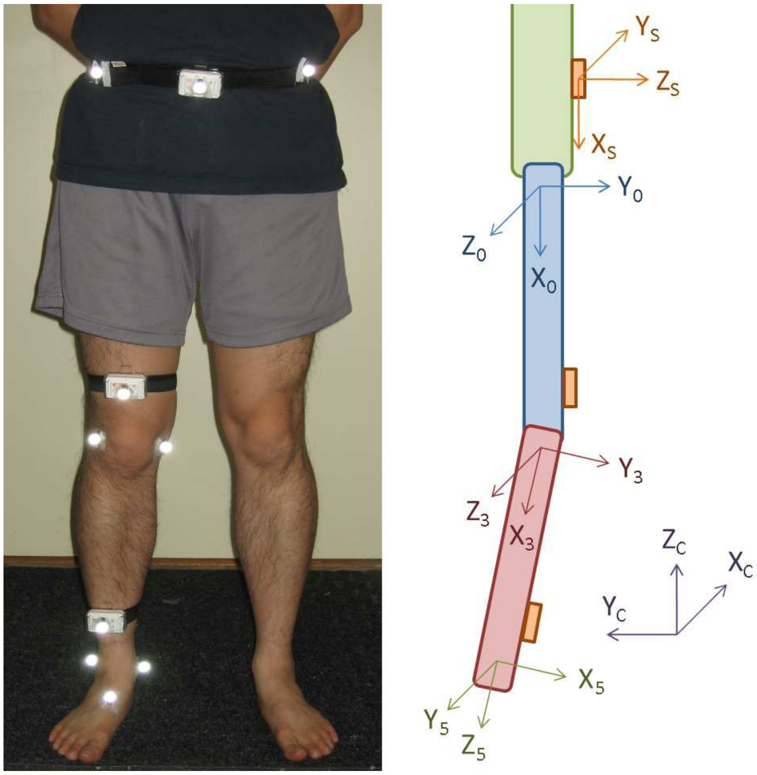


Figure 6.2: The hip and knee are modeled as a 3 and 2 DOF joint, respectively. Shimmer sensors, containing the accelerometers and gyroscopes, were positioned on the hip, knee and ankle of the participant. (Left) An image of the sensors on a participant's hip, knee and ankle. Motion capture markers placements are also shown. (Right) A profile image of the torso, upper and lower leg, rotated 90° from the left image. The boxes on the leg represent the sensor packages.

Table 6.1: Participant anthropometric data. All lengths in $[cm]$, all weights in $[kg]$. r_u is the hip-to-knee length. r_{usl} is the length along the thigh to the knee sensor. r_{usr} is the radius of the thigh at the knee sensor. r_l is the knee-to-ankle length. r_{lsl} is the length along the calf to the ankle sensor. r_{lsr} is the radius of the calf at the ankle sensor.

	Height	**Weight**	**Age**	$\boldsymbol{r_u}$	$\boldsymbol{r_{usl}}$	$\boldsymbol{r_{usr}}$	$\boldsymbol{r_l}$	$\boldsymbol{r_{lsl}}$	$\boldsymbol{r_{lsr}}$
Mean	168.7	65.2	23.3	44.8	34.9	7.2	38.2	23.4	6.5
SD	9.2	10.1	3.5	4.3	6.1	1.5	2.5	4.7	0.6
Max	183.0	90.0	31.0	57.3	47.0	12.0	43.4	32.1	8.2
Min	150.0	47.7	19.0	37.4	22.7	5.2	33.9	14.4	5.3

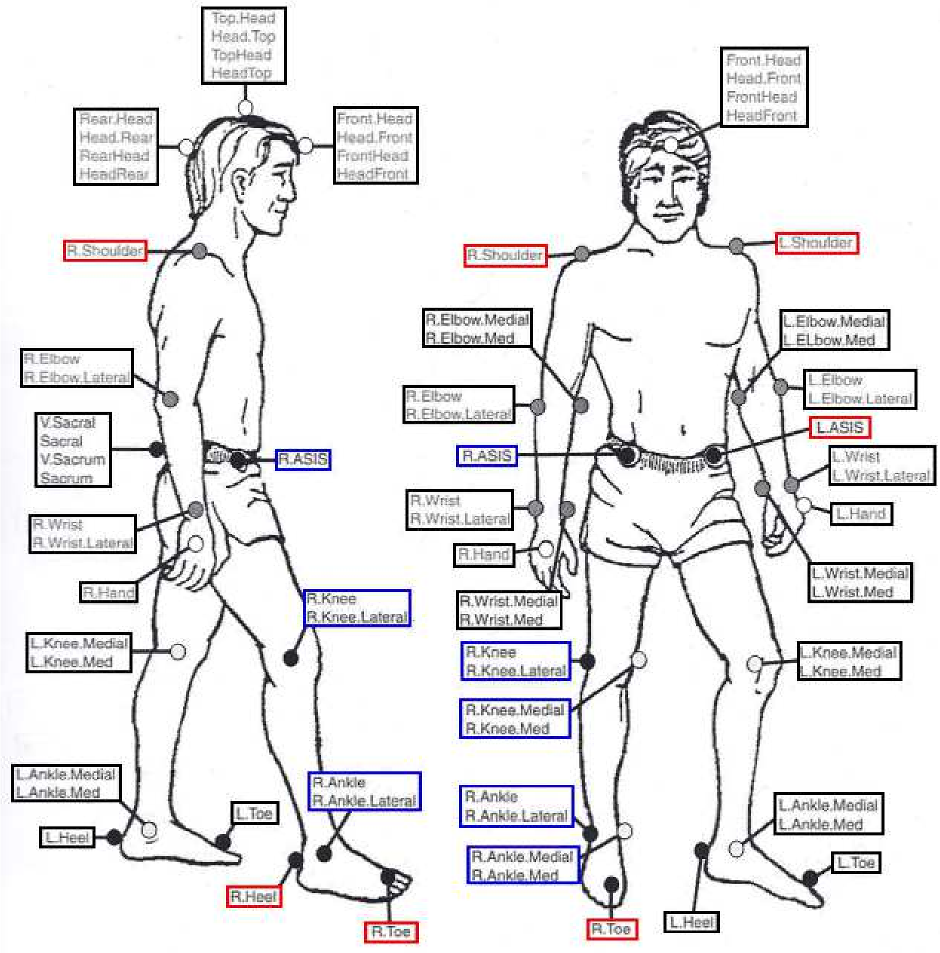


Figure 6.3: Motion capture markers used. Red boxes indicate markers that were collected, but were not directly used. Blue boxes indicate markers that were collected, and were used in calculations. Image is from the Motion Analysis Cortex manual [72], and is used with permission from Motion Analysis Corporation.

Table 6.2: Exercise descriptions for rehabilitation exercises.

Name	Initial Pose	Description
Knee flexion (KF)	Seated	Knee extension/flexion.
Sit to stand (SS)	Seated	Stand up from a seated posture.
Squats (SQ)	Standing	While keeping upper body upright, bend knees to lower torso vertical position.
Hip-knee flexion (HF)	Lying down	Starting with the leg straight, bend at the knee and hip to slide the heel towards the buttocks.
Straight leg raise (SL)	Lying down	Keeping the leg straight, bend at the hip and lift the entire leg up from the ground.

However, due to the nature of the rehabilitation motions examined, the motions tested in Table 6.2 were all sagittal plane motions. A second set of out-of-plane motions was collected with 1 male participant in order to verify the algorithm's capability to recover out-of-plane movement. The tested motions for this second set can be found in Table 6.3.

Table 6.3: Exercise descriptions for out-of-plane motions.

Name	Initial Pose	Description
Circle trace (CTS)	Standing	Trace out a circle in the transverse plane with leg while in a standing position, rotating only at hip.
Front-side leg raise (FSS)	Standing	Alternating between leg raises to the front (sagittal) and side (coronal).
Circle trace (CTL)	Lying down	Tracing out a circle in the transverse plane, with minimum knee bending.
Straight-diagonal leg raise (SDR)	Lying down	Alternating between front leg extension (sagittal) and side leg extension (away from sagittal).

6.2 Pose Estimation

The algorithm described in Chapter 4 was applied to the Shimmer data and tested against motion capture data. The EKF functions were implemented with the ReBEL MATLAB Toolbox [102]. The initial angles for the EKF were determined numerically by applying inverse kinematics to the Cortex data.

The initial EKF noise profiles (Q_t, R_t) for the rehabilitation motion dataset were tuned to a single participant. That is, the 5 exercise motions of the reference patient were examined, and a single Q_t and R_t matrix was derived across all 5 exercises. The tuning was done by minimizing RMS between the EKF-derived end-effector position and the end-effector position from the motion capture system. This noise profile was applied uniformly to all the exercises of the other 19 participants. For motions that were performed by other participants that may differ in speed, the auto-tuning mechanism described in Section 4.3 was used to adapt the noise model. The kinematic constraint limits q_c were aggregated from several anthropometric joint limit tables [75, 12]. The kinematic link lengths were obtained from the motion capture marker information. The potential field coefficient η and the influence distance ρ_0 were set to 1 and 75% of the joint limits, respectively.

6.2.1 State Estimation for Rehabilitation Motions

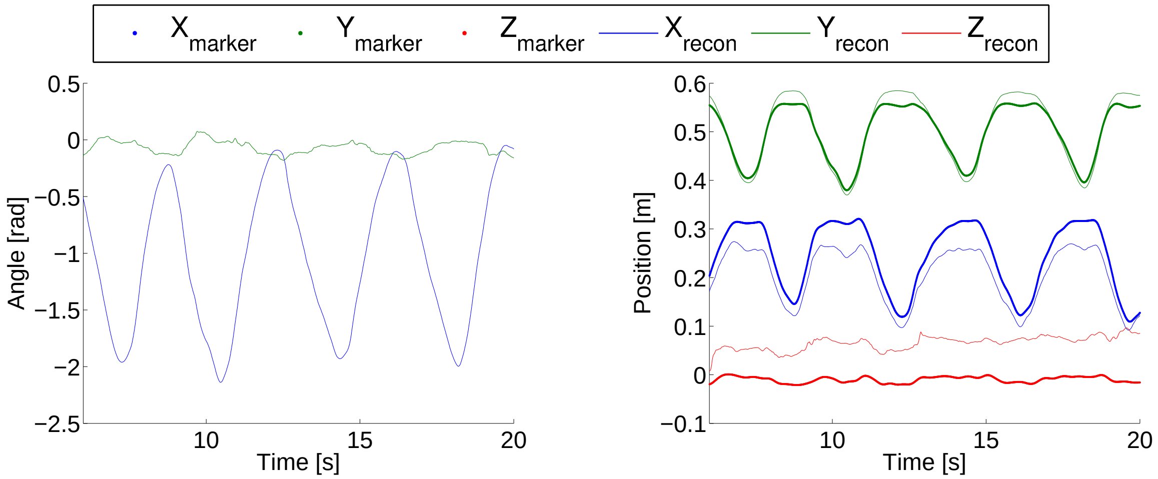
Prior to the accuracy analysis, two outlier motions were removed from the total set of 200 motion sets, due to a sensor error during data collection. To compare the results obtained from the EKF with the motion capture data, following estimation of the joint angles using the proposed algorithm, Equations 4.23 and 4.24 were used to compute the Cartesian location of the IMU markers. The estimated locations of the knee and ankle sensors were compared to the motion capture measured sensor locations, and the RMS computed. Table 6.4 presents the RMS for each Cartesian direction, as well as the average error for the thigh (upper) link, the calf (lower) link, and the total average error. The data were grouped together by motion type, with only the anthropometric joint limits as the kinematic constraints. The RMS data were converted to joint angles by applying the cosine rule to the link lengths and the RMS values. Table 6.4 also presents the joint angle errors for the thigh and the calf link.

Considering these results, it is evident that the largest errors exist in the z-axis (i.e., the motion out of the sagittal plane, c.f. Figure 6.2), where there is little actual motion. Without strong motion in that direction, the estimation suffers from drift, causing a large reported RMS. As can be seen from Figure 6.4, errors are also introduced by poor initial angle estimates, which can persist over the duration of the state recovery process.

In general, larger RMS were reported in the calf than the thigh. This is expected, as the end-effector distance errors from the thigh are also implicitly included in the calf. Once again, the impact of drift can be observed strongly in the calf z-axis. With only general anthropometric constraints, Table 6.4 shows a total mean RMS of 4.27 cm, with 3.46 cm for the thigh and 5.08 cm for the calf.

Table 6.4: EKF RMS results, by motion type, for the rehabilitation dataset, without any out-of-plane angle constraints. Cortex link length used. Results shown for each of the Cartesian dimensionality, as well as thigh (U), calf (L) and total (T) errors. The degree error for the hip (U) and knee (L) joint is also shown. The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL).

	RMS $[cm]$									**RMS** $[deg]$	
	X1	**Y1**	**Z1**	**X2**	**Y2**	**Z2**	**U**	**L**	**T**	**U**	**L**
KF	3.98	1.58	5.53	4.39	3.45	8.49	3.70	5.44	4.57	6.01	4.12
SS	1.59	2.28	2.44	2.19	4.16	2.78	2.10	3.04	2.57	7.60	4.90
SQ	1.59	2.29	4.84	2.81	4.34	5.78	2.91	4.31	3.61	10.82	8.45
HF	5.01	3.18	5.00	5.88	4.35	8.52	4.40	6.25	5.32	6.28	4.86
SL	5.72	2.96	3.93	6.21	3.53	9.44	4.20	6.39	5.30	5.91	5.53
AVG	3.56	2.46	4.35	4.28	3.98	6.97	3.46	5.08	4.27	7.35	5.59



(a) Knee joint.

(b) Calf sensor end-effector.

Figure 6.4: Pose estimation of leg extension motion, showing joint angle estimation and calculated end-effector position compared to the motion capture end-effector position. Drift can be seen in the z-axis (red), while initial angle error may sometimes persist in the state recovery.

Table 6.5: EKF RMS results, by motion type, for the out-of-plane dataset. Results shown for each of the Cartesian dimensionality, as well as thigh (U), calf (L) and total (T) errors. The degree error for the hip (U) and knee (L) joint is also shown. The motions performed were circle trace (CTS), front-side leg raise (FSS), circle trace (CTL) and straight-diagonal leg raise (SDR).

	RMS $[cm]$									**RMS** $[deg]$	
	X1	**Y1**	**Z1**	**X2**	**Y2**	**Z2**	**U**	**L**	**T**	**U**	**L**
CTS	2.09	7.49	6.99	2.82	12.22	9.71	5.77	8.25	7.01	8.89	6.08
FSS	2.16	5.44	4.44	2.59	10.67	8.55	4.16	7.27	5.71	6.32	7.42
CTL	2.30	1.57	3.58	5.22	3.30	7.56	3.45	5.36	4.41	5.21	6.49
SDR	3.44	1.91	4.90	4.00	2.47	5.88	3.60	4.12	3.86	5.39	1.59
AVG	2.50	4.10	4.98	3.66	7.17	7.93	4.25	6.25	5.25	6.45	5.40

In addition to the errors from the EKF estimates, errors are also introduced from the motion capture itself. Markers were only placed on the front of the hip and not the back. The hip rotational centre was assumed to be at the hip marker position, which is incorrect. This also applies to the location of the IMU sensors as well. This problem is less significant for the knee and ankle joint as the position of these two joints were estimated by considering multiple markers. Due to the nature of human joints, joint centres also shift during movement, contributing to additional error.

6.2.2 State Estimation for Out-of-plane Motions

The RMS for the out-of-plane motions were calculated in the same manner as the rehabilitation motions. Table 6.5 presents the Cartesian and degree error by motion type. With an total Cartesian RMS of 5.25 *cm* and an hip and knee joint error of 6.45° and 5.40° respectively, the algorithm is shown to work with similar effectiveness with out-of-plane motions when compared to the in-plane rehabilitation motions.

Although only healthy participants were examined, the proposed algorithm is well suited to use on the rehabilitation population. The proposed algorithm does not assume any *a priori* knowledge of the movement that is being performed and functions identically for the sagittal-plane dataset and the out-of-plane dataset. No parameter modifications were made to apply the algorithm to each of the healthy participants, covering a range of movement speed and pattern. It is possible that, with the rehabilitation population, state noise covariance would need to be modified, but the mechanics of the algorithm would remain identical.

6.2.3 Comparison to Existing Approaches

Pose Estimation

Two alternative algorithms were also implemented to serve as comparison benchmarks: Boonstra *et al.*'s implementation of the complementary filter [11] and Luinge and Veltink's implementation of the strapdown integrator [67]. Similar to the proposed algorithm, the two comparison algorithms rely on accelerometers and gyroscopes to produce orientation estimates. Both the accelerometer and the gyroscope are important, since the accelerometer returns accurate incline angles due to gravity, but is not reliable when the subject's motion acceleration is not negligible compared to gravity. The gyroscope can be integrated to obtain joint angles during faster movements, but suffers from integrational drift. Boonstra's algorithm considers only sagittal motion, so comparisons to the proposed algorithm are made to the sagittal-only results, where out-of-plane angles were locked to the initial joint values. The Cartesian RMS results can be found in Table 6.6a, and Table 6.6b for joint angle error.

Several minor modifications were made to Boonstra's algorithm for this comparison. Rather than using the filter parameters recommended by Boonstra, which may be optimized for their sensor suite, the filter parameters were optimized on the Shimmer devices in order to obtain the best possible performance. The sagittal incline angle was used to rotate the link lengths so that Cartesian RMS can be calculated for comparison against the proposed algorithm. Lastly, it was unclear how Boonstra combined the accelerometer-derived joint angle to the gyroscope-derived joint angle, so the average of the two was used.

From Table 6.6, it can be noted that Boonstra's algorithm does not perform as well as the proposed algorithm. The high-pass filter used in Boonstra does not perfectly eliminate the gyroscopic drift, which can be very significant. Without the accelerometer angle to average with, the reconstructed angles derived from the gyroscope would be very poor. This is the cause of numerous high-error entries for Boonstra in Table 6.6, emphasizing that the high-pass filter is not sufficient. Boonstra also notes that the high-pass filter may be removing movement information from the sensors, which would also hurt the angle estimates. Lastly, Boonstra's algorithm does not provide a methodical way to combine the measurement sources, whereas the EKF calculates a weighing factor, the Kalman gain (Equation 3.18), to combine all the measured data appropriately.

The second comparison algorithm considered was Luinge's implementation of the strapdown integration approach, where the movement and orientation of an object are estimated by rigidly attached IMUs, using a Kalman filter framework. The angles are determined by

Table 6.6: EKF RMS results by motion type, for the rehabilitation dataset, with out-of-plane angle constraints, comparing two other algorithms with the proposed. Results are given for the thigh link (U), calf link (L) and total error (T). The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL).

(a) RMS Cartesian error, reported in [cm].

	Boonstra			**Luinge, no bias**			**Luinge, bias**			**Proposed**		
	U	**L**	**T**	**U**	**L**	**T**	**U**	**L**	**T**	**U**	**L**	**T**
KF	3.13	6.83	4.98	2.78	4.53	3.66	2.51	3.82	3.16	2.51	3.48	2.99
SS	2.02	3.58	2.80	1.22	2.36	1.79	1.99	3.36	2.67	1.72	2.74	2.23
SQ	1.70	4.04	2.87	1.52	2.88	2.20	1.95	3.58	2.76	1.91	3.26	2.69
HF	6.18	8.24	7.21	4.12	4.86	4.49	4.06	4.79	4.43	3.90	4.50	4.20
SL	8.05	10.62	9.33	4.16	5.83	5.00	4.18	5.59	4.88	4.37	4.56	4.47
AVG	4.20	6.65	5.42	2.76	4.09	3.42	2.93	4.22	3.58	2.87	3.70	3.31

(b) RMS degree error, reported in [deg].

	Boonstra		**Luinge, no bias**		**Luinge, bias**		**Proposed**	
	U	**L**	**U**	**L**	**U**	**L**	**U**	**L**
KF	5.02	8.48	4.59	4.23	4.11	3.15	4.13	1.67
SS	7.26	9.48	4.45	5.72	7.25	6.94	6.00	3.74
SQ	6.36	13.07	5.60	7.60	7.08	8.86	7.08	5.39
HF	8.83	5.01	5.88	2.08	5.75	1.96	5.55	1.53
SL	11.20	6.00	5.88	4.89	5.90	4.06	6.17	1.12
AVG	7.70	8.45	5.29	4.90	6.04	5.01	5.81	2.73

gyroscopic integration, which is then used to rotate and isolate the gravity vector from the inertial acceleration. The Kalman filter combines the sagittal incline angle derived from the acceleration signal with the integrated angles derived from the gyroscope signal. The sensor noise is explicitly modeled by the Kalman filter. The state noise covariance matrix used was obtained by tuning, but the measurement noise covariance matrix was identical as the one used in the proposed algorithm, since the state model used was different, but the sensors and data were the same as the proposed.

Luinge's algorithm also includes the gyroscope sensor DC bias as a Kalman state variable to address drift. Two implementations of Luinge's algorithm were constructed for the results shown in Table 6.6, one with and one without gyroscope bias correction. The two implementations allow the analysis of the contribution of the kinematic model and the contribution of bias correction to the system performance separately. For the rehabilitation dataset, the gyroscope bias does not affect performance, since out-of-gravitation plane motion, which is most susceptible to drift, is restricted. Without a gyroscope bias model, EKF may not be able to properly estimate the gyroscope noise, causing a poorer estimation.

A full 3D DOF version of Luinge's algorithm was also implemented, to compare between Luinge's algorithm and the proposed algorithm in an unconstrained situation, using the out-of-plane motion dataset. Table 6.7 outlines the results from Luinge's algorithm in terms of Cartesian and joint angle error. Comparing the results for the proposed algorithm in Table 6.5 and Luinge's algorithm in Table 6.7, it can be seen that the performance in the sagittal plane is comparable, but the overall z-axis off-sagittal drift is much larger with Luinge's algorithm, resulting in a larger overall error reported for Luinge's algorithm. The inclusion of a kinematic model allows the proposed algorithm to constrain unfeasible motions that Luinge's algorithm does not, such as abduction motion in the knee. This allows the proposed algorithm to reduce drift.

Table 6.8 shows Luinge's algorithm with and without gyroscope bias, as well as the proposed algorithm in full 3D angle recovery without any sagittal constraints. Unlike the sagittal case, the inclusion of the gyroscope bias in Luinge's algorithm significantly improves performance in the case of unconstrained motion, particularly in the off-sagittal axis direction.

The performance of Luinge's algorithm is comparable to the proposed approach, showing the advantage of the principled integration of both sensor sources via the Kalman filter framework. However, with both comparison algorithms, only incline angles to the horizontal and heading angle to the vertical are estimated, while the proposed algorithm can estimate pose and report the results in the more clinically-relevant joint angle. Due to the

Table 6.7: EKF RMS results, by motion type, for the out-of-plane dataset, using Luinge's algorithm without gyroscope bias. Results shown for each of the Cartesian dimensionality, as well as thigh (U), calf (L) and total (T) errors. The degree error for the hip (U) and knee (L) joint is also shown. The motions performed were circle trace (CTS), front-side leg raise (FSS), circle trace (CTL) and straight-diagonal leg raise (SDR).

	RMS [cm]									**RMS** [deg]	
	X1	**Y1**	**Z1**	**X2**	**Y2**	**Z2**	**U**	**L**	**T**	**U**	**L**
CTS	21.85	17.05	23.32	48.55	29.51	41.68	20.74	39.91	30.33	32.38	43.89
FSS	1.49	2.63	2.23	3.16	6.59	4.42	2.12	4.72	3.42	3.21	5.95
CTL	4.46	2.21	3.31	5.69	2.76	5.56	3.32	4.67	4.00	5.04	3.02
SDR	4.72	2.06	2.42	11.66	5.12	5.83	3.06	7.54	5.30	4.60	10.17
AVG	8.13	5.99	7.82	17.26	11.00	14.37	7.31	14.21	10.76	11.31	15.76

Table 6.8: EKF RMS results by motion type, for the out-of-plane dataset, comparing Luinge's algorithm with the proposed. Results are given for the thigh link (U), calf link (L) and total error (T). The motions performed were circle trace (CTS), front-side leg raise (FSS), circle trace (CTL) and straight-diagonal leg raise (SDR).

(a) RMS Cartesian error, reported in [cm].

	Luinge, no bias			**Luinge, bias**			**Proposed**		
	U	**L**	**T**	**U**	**L**	**T**	**U**	**L**	**T**
CTS	20.74	39.91	30.33	5.61	11.34	8.47	5.77	8.25	7.01
FSS	2.12	4.72	3.42	3.26	8.15	5.71	4.16	7.27	5.71
CTL	3.32	4.67	4.00	4.10	8.47	6.29	3.45	5.36	4.41
SDR	3.06	7.54	5.30	3.92	7.79	5.86	3.60	4.12	3.86
AVG	7.31	14.21	10.76	4.22	8.94	6.58	4.25	6.25	5.25

(b) RMS degree error, reported in [deg].

	Luinge, no bias		**Luinge, bias**		**Proposed**	
	U	**L**	**U**	**L**	**U**	**L**
CTS	32.38	43.89	8.65	12.81	8.89	6.08
FSS	3.21	5.95	4.94	11.17	6.32	7.42
CTL	5.04	3.02	6.20	9.86	5.22	6.49
SDR	4.60	10.17	5.89	8.77	5.40	1.59
AVG	11.31	15.76	6.42	10.65	6.46	5.40

lack of kinematic constraints, the comparison algorithms also allow drift in DOFs that are not physically feasible, which degrades the quality of the angle estimates.

Comparison to Goniometry

The error magnitudes obtained by the proposed algorithm are comparable to errors reported using goniometric systems. Chao [19] reported a sagittal mean error of 1.4°, when comparing an electrogoniometer with x-ray data. Jagodzinski *et al.* [45] examined hip and knee goniometeric measurement accuracies against radiology images of the leg, and noted a mean degree error of 3.92° with a standard goniometer and 1.22° with a long-arm goniometer, with 3.3° of inter-rater variance between the two examiners. Edwards *et al.* [24] reports that mean error of goniometer measurement of knee angle ranged from 0° at 90° knee flexion, to 3° error at 120° flexion. The average error rose to an average of 5° error with unaided visual assessments. The corresponding joint error for the proposed algorithm reports a sagittal error of 2.73°, which makes it better than an unaided joint assessment, and comparable to existing goniometric methods. This makes the proposed algorithm a viable alternative to the existing goniometer methods as the sensors are lightweight and do not require a brace such as those required by an electrogoniometer. It also provides joint angle estimates in both static and dynamic situations, providing exercise velocity and timing information that a goniometeric system would not be able to supply.

6.3 Segmentation and Identification

The joint angles obtained from Section 6.2 were pre-processed with a low-pass filter to reduce state reconstruction noise caused by any filter resets. The HMM functions were implemented with Murphy's HMM MATLAB Toolbox [73].

An 8-state left-right HMMs were used, trained using the Baum-Welch algorithm. $\boldsymbol{\pi_H}$ was initialized with $[1; 0; 0; ...]$, as typical for left-right HMMs. The initial $\boldsymbol{\mu_H}$ was calculated by separating the templates into 8 sequential equally sized segments, and calculating the mean for each DOF within each segment. The initial $\boldsymbol{\Sigma_H}$ for each state was set to a diagonal matrix of 10, and allowed to converge after several iterations, as some $\boldsymbol{\Sigma_H}$ values suggested by separating the data into 8 segments and calculating the variance resulted in numerical issues with the training algorithm. Ground truth manual segmentation was determined by a human observer using video playback of the motion capture data.

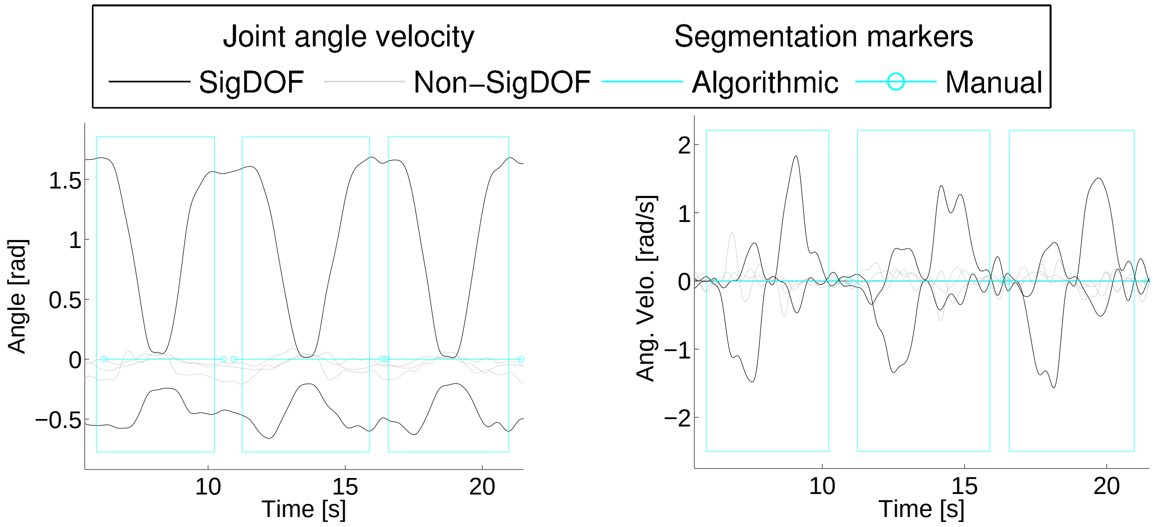
An algorithmic segmentation point was declared correct if it falls within $\pm t_{error}$ of a manual segment point. A false positive error was declared if an algorithmic segment point was identified when there is not a corresponding manual segmentation point. A false negative error was declared if a segment point was not found algorithmically for a manually identified segment point. Each segment's two segment points are awarded points separately. That is, a completely correct segment could receive +2 correct, whereas a partially correct segment could receive +1 correct and +1 false negative. All motion sequences contain only a single motion type, so if the algorithm labels a given segment incorrectly, it is reported independently from the segmentation results.

6.3.1 Segmentation Results

The data streams used for the feature selection were the 5 joint angles from the EKF (Section 6.2). For each subject, the first of the two collected motion sets was used for the template training, then both motion sets were used separately for testing the proposed algorithm. In the first set of experiments, the templates used were subject specific, so the templates generated from one subject's motions was used to segment only their own motion.

Figures 6.5 (sit-to-stand), 6.6 (squats), and 6.7 (hip flexions while supine) show various motions segmented by the algorithm. The boxes indicate the algorithmic segmentation points, while the circles indicate manual segmentation points. The top plot shows angle data; the bottom plot shows angular velocity. The dark lines are DOFs denoted as significant across all the templates, whereas the lighter lines are not significant.

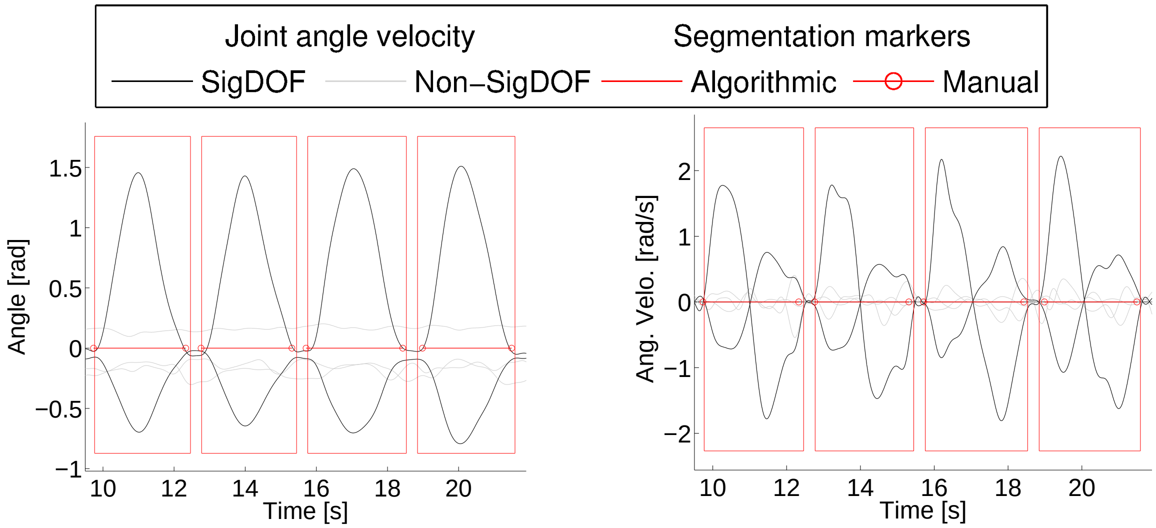
The segmentation results for the feature-guided HMM with $t_{error} = 0.2\ s$ can be found in Table 6.10. This table shows the segmentation results on both the training and the testing set. The algorithm correctly determines 76% of the segments for data not used during training. If $t_{error} = 0.3\ s$, shown in Table 6.11, the accuracy increases to 84%, implying that a number of the algorithmic segments are just outside of the manual segment error bounds. Several different factors contribute to this result. Since there are no standardized guidelines on how a motion sequence should be segmented, manual segmentation is typically left up to the perception of the human observer. Although manual segment points are typically used as ground truth, there may be inconsistencies between observers or even between exemplars for the same observer. It can be difficult to visually determine when a segment has started or ended, due to stray motion or tremors at segment ends, or due to poor camera angle during video replay, causing manual segments to be declared before or after the actual end of motion. The observer may also be segmenting on DOFs that are not the most significant. At times, the algorithm overestimates the bounds of the segments, as the motion may have been perceived to have ended due to slow velocity before the actual



(a) Joint angle.

(b) Angular velocity.

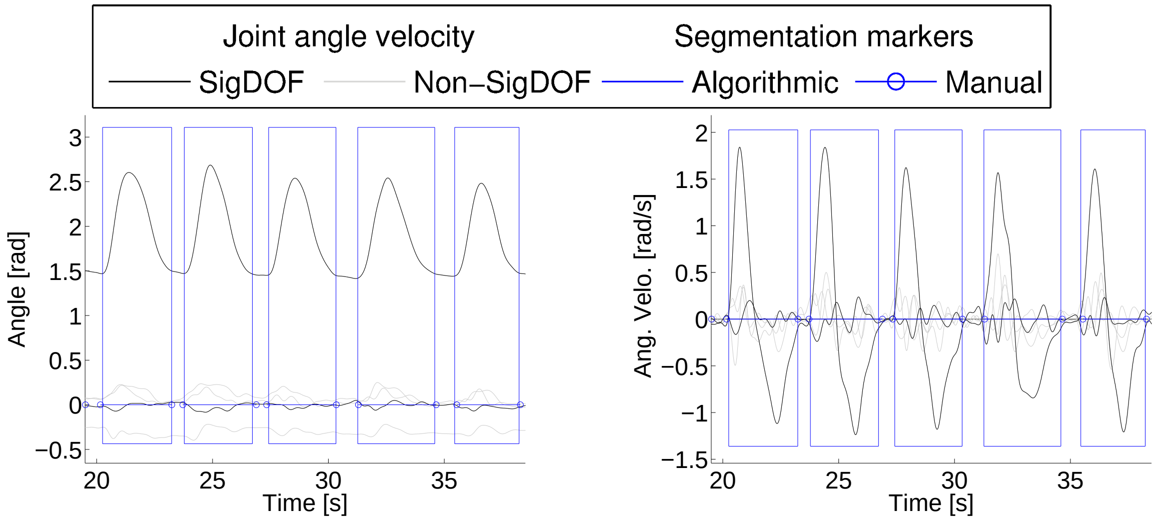
Figure 6.5: Joint angles and angular velocity of a subject performing sit-to-stand. The circles denote the manual segments, whereas the rectangles denote the algorithmic segments. In this figure, only one of the six segment points is declared a correct segment under $t_{error} = 0.2s$, even though the algorithm located all the motions. Reaction speed of the human observer is a common source of the mismatch between the automatic and the manual segmentation, as the manual segments sometimes lag behind the time when the participant ceases motion. This can be seen most strongly between the second and third segment. Examining the space between the first and second segment, it can be noted that the participant does not actually stop moving between repetitions, making the location of the true segment point ambiguous, especially if the viewing angle is not optimal.



(a) Joint angle.

(b) Angular velocity.

Figure 6.6: Joint angles and angular velocity of a subject performing squats. The circles denote the manual segments, whereas the rectangles denote the algorithmic segments.



(a) Joint angle.

(b) Angular velocity.

Figure 6.7: Joint angles and angular velocity of a subject performing hip flexions. The circles denote the manual segments, whereas the rectangles denote the algorithmic segments. Despite significant motion in the other DOFs, the algorithm successfully segments the motion under examination.

ZVC. If segmentation accuracy is ignored and only identification accuracy is examined, as reported in Table 6.9, the algorithm reports 95% accuracy.

Utilizing only angle data in the HMM states sometimes leads the identification process to confuse two similar actions. For example, squats and the sit-to-stand both involve a large change in the knee sagittal angle and a small change in the ankle sagittal angle. Using joint data only, the squat position is sometimes confused with the seated position, causing misidentification. Adding the acceleration signal introduces strong additional orientation-based data into the HMM vectors, reducing this misidentification. These results can be found in Tables 6.10 and 6.11. These results show that the false positives, which include template misidentification, decrease with the addition of the accelerometer terms. Including these accelerometer terms raises the segmentation accuracy, particularly for the squat and sit-to-stand motions. These results also demonstrate the capability of the proposed algorithm to handle a variety of input time series data, including both joint angle and Cartesian acceleration measurements. Table 6.9 shows that the identification accuracy also increases to 100%.

Examining these results shows that the sit-to-stand motion performs particularly poorly in terms of segmentation accuracy. A detailed examination of the sit-to-stand results shows that the segmentation routine produces results comparable to the other motions, but a mismatch between the locations of the manual and algorithmic segmentations causes the algorithm to report a lower accuracy (see Figure 6.5). We believe this is due to issues with the viewing angle during manual segmentation, especially if the participant does not perform a complete stop between repetitions. Comparing $t_{error} = 0.2\ s$ to $t_{error} = 0.3\ s$, the sit-to-stand accuracy reported for sit-to-stand for both datasets in Tables 6.10 and 6.11 rises to from 49% and 52% to 65% and 77% respectively. If sit-to-stand is not included, the total accuracy for these two tables rises from 76% and 77% to 82% and 83%, respectively.

6.3.2 Inter-personal Template Segmentation Results

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

The participant data were randomly separated into groups of three or five, and templates were generated from these groups. The accelerometer data were rotated before

Table 6.9: Identification results of the proposed algorithm, comparison when using joint angles only (JA) or joint angles and accelerometer data (JA+A). The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL). Separate results are reported for trained and untrained data.

	Trained data			**Untrained data**		
	Total	**JA**	**JA+A**	**Total**	**JA**	**JA+A**
KF	406	404 (100%)	406 (100%)	410	384 (94%)	410 (100%)
SS	398	388 (97%)	398 (100%)	398	364 (91%)	398 (100%)
SQ	416	410 (99%)	416 (100%)	412	376 (91%)	412 (100%)
HF	418	400 (96%)	418 (100%)	408	396 (97%)	408 (100%)
SL	406	406 (100%)	406 (100%)	410	410 (100%)	410 (100%)
Total w/o SS	1646	1620 (98%)	1646 (100%)	1640	1566 (95%)	1640 (100%)
Total	2044	2008 (98%)	2044 (100%)	2038	1930 (95%)	2038 (100%)

HMM training to align the mounting orientation of the sensors between participants. All other parameters were identical to the ones used in Section 6.3.1.

Table 6.13 shows the mean and standard deviation of the accuracy percentages over all the template groups. The data used for template training were not used for the segmentation testing. The algorithm demonstrates good robustness when using participant-independent rather than participant-specific templates, dropping approximately 5% for overall accuracy for $t_{error} = 0.2$ s and 6% for overall accuracy for $t_{error} = 0.3$ s compared to the single-template results from Section 6.3.1. Identification accuracy drops by 1%, as per Table 6.12. Similar to the results in Section 6.3.1, the sit-to-stand results were noticeably worse. With exception of the sit-to-stand, the 5-template set outperformed the 3-template sets in all the motions. This was to be expected, since the 5-template set includes a wider variety of exemplar motions, and thus was more able to account for inter-participant variability.

6.3.3 Comparison to Other Segmentation Algorithms

In order to compare the proposed approach to existing work, a ZVC method, described in Fod *et al.* [30] was implemented. Segmentation points are declared when the velocity crosses zero, thus a segment window is formed with two consecutive ZVCs. Segment windows that are too small are rejected. To reduce the amount of spurious ZVCs, only

Table 6.10: Segmentation results of the proposed algorithm, comparison when using joint angles only (JA) or joint angles and accelerometer data (JA+A). The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL). Results shown is the algorithm's performance for $t_{error} = 0.2\ s$, and the number of segment point estimates that are correct, false positive (FP) and negative (FN). Separate results are reported for trained and untrained data.

(a) Trained data

		JA			**JA+A**		
Motion	**Total**	**Correct**	**FP**	**FN**	**Correct**	**FP**	**FN**
KF	406	382 (94%)	2	22	382 (94%)	0	22
SS	398	211 (53%)	42	187	264 (66%)	26	134
SQ	416	337 (81%)	8	79	343 (83%)	2	73
HF	418	330 (79%)	20	88	342 (82%)	2	76
SL	406	353 (87%)	6	53	353 (87%)	4	53
Total w/o SS	1646	1402 (85%)	36	242	1420 (86%)	8	224
Total	2044	1613 (79%)	78	429	1684 (82%)	34	358

(b) Untrained data

		JA			**JA+A**		
Motion	**Total**	**Correct**	**FP**	**FN**	**Correct**	**FP**	**FN**
KF	410	371 (91%)	26	39	389 (95%)	0	21
SS	398	193 (49%)	72	205	207 (52%)	62	191
SQ	412	294 (71%)	36	118	328 (80%)	0	84
HF	408	331 (81%)	14	77	320 (78%)	2	88
SL	410	349 (85%)	12	61	324 (79%)	12	86
Total w/o SS	1640	1345 (82%)	88	295	1361 (83%)	14	279
Total	2038	1538 (76%)	160	500	1568 (77%)	76	470

Table 6.11: Segmentation results of the proposed algorithm, comparison when using joint angles only (JA) or joint angles and accelerometer data (JA+A). The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL). Results shown is the algorithm's performance for $t_{error} = 0.3\ s$, and the number of segment point estimates that are correct, false positive (FP) and negative (FN). Separate results are reported for trained and untrained data.

(a) Trained data

		JA			**JA+A**		
Motion	**Total**	**Correct**	**FP**	**FN**	**Correct**	**FP**	**FN**
KF	406	389 (96%)	2	15	389 (96%)	0	15
SS	398	276 (69%)	20	122	325 (82%)	8	73
SQ	416	376 (90%)	8	40	382 (92%)	2	34
HF	418	363 (87%)	18	55	380 (91%)	0	38
SL	406	379 (93%)	2	27	378 (93%)	2	28
Total w/o SS	1646	1507 (92%)	30	137	1529 (93%)	4	115
Total	2044	1783 (87%)	50	259	1854 (91%)	12	188

(b) Untrained data

		JA			**JA+A**		
Motion	**Total**	**Correct**	**FP**	**FN**	**Correct**	**FP**	**FN**
KF	410	375 (92%)	26	35	393 (96%)	0	17
SS	398	257 (65%)	54	141	308 (77%)	20	90
SQ	412	328 (80%)	36	84	362 (88%)	0	50
HF	408	364 (89%)	12	44	351 (86%)	0	57
SL	410	381 (93%)	2	29	354 (86%)	2	56
Total w/o SS	1640	1448 (88%)	76	192	1460 (89%)	2	180
Total	2038	1705 (84%)	130	333	1768 (87%)	22	270

Table 6.12: Identification results presented as accuracy percentage with inter-subject templates. The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL).

	3-template set		**5-template set**	
	Mean	**SD**	**Mean**	**SD**
KF	99.1%	1.9%	99.4%	0.9%
SS	99.4%	1.2%	98.3%	1.9%
SQ	99.7%	0.2%	98.8%	0.9%
HF	99.2%	1.2%	99.0%	1.2%
SL	100.0%	0.1%	100.0%	0.0%
Total w/o SS	99.5%	0.5%	99.3%	0.5%
Total	99.5%	0.4%	99.1%	0.8%

Table 6.13: Segmentation results under different t_{error} values, with inter-subject templates. The motions performed were knee flexion (KF), sit to stand (SS), squats (SQ), hip-knee flexion (HF) and straight leg raise (SL).

	$\boldsymbol{t_{error}} = \mathbf{0.2}\ s$				$\boldsymbol{t_{error}} = \mathbf{0.3}\ s$			
	3-template set		**5-template set**		**3-template set**		**5-template set**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
KF	85.3%	2.0%	86.6%	1.6%	86.7%	2.1%	88.0%	1.8%
SS	53.1%	4.1%	53.2%	8.8%	70.7%	6.5%	69.8%	11.3%
SQ	73.6%	2.4%	74.6%	3.4%	81.2%	2.8%	82.6%	3.2%
HF	74.3%	8.8%	74.4%	11.9%	82.4%	7.6%	82.4%	12.4%
SL	79.5%	2.1%	83.8%	1.1%	85.7%	2.7%	90.6%	1.1%
Total w/o SS	78.9%	1.2%	81.0%	1.5%	84.3%	1.7%	86.6%	2.2%
Total	73.1%	1.5%	74.6%	3.0%	81.2%	2.0%	82.8%	3.5%

ZVCs that occurred on the significant DOFs were considered. Since ZVC produces segment points at both the end of flexion and the end of extension, every two consecutive ZVC segment windows were combined.

A fixed-sliding window HMM [80] was also implemented. HMM construction for the fixed sliding window was identical to the feature guided HMM. The fixed window length was computed from the lengths of the exemplar templates. Segmentation points were declared on local maximas of the likelihood, as long as it is above some likelihood threshold.

Lastly, a DTW algorithm [87] was also implemented. An arbitrary exemplar is selected to be the template motion. A mapping matrix, used to time-align two time series signals, is constructed by taking the Euclidean distance between each time step. An unconstrained path was used, since motion data often contains pauses, which is less suited for a greedy algorithm [97]. The warping path width was constrained to 8 timestamps to prevent singularities. A segment candidate is considered a primitive candidate as long as the Euclidean distance between a given template and the observation data was maintained below some empirically-derived threshold. The threshold is determined by taking the distance between all the exemplar motions against the template exemplar, and taking the average distance. If the observation data is of a different length then that of the template, a penalty distance is added, proportional to the difference in length. Initial tests with a bottom-up DTW [49] resulted in 50% accuracy, with average runtime of 700 seconds, indicating that basic DTW is not suitable for this application. To reduce the DTW computation time and improve accuracy, a modified version of DTW was implemented. The feature-searching component (Figure 5.5.2-6) from the proposed algorithm was used to determine the potential segment points, and the HMM component in the proposed algorithm was replaced with DTW. That is, potential segment points were located by considering the sequence of velocity peaks and zeros in the observation data, while rejecting short ZVC intervals or small velocity peaks. When a sequence of velocity features matches that of a given template, a segment potential is declared and assessed by the DTW. This also allows the HMM and the DTW to be compared in a more straight forward fashion.

The 5 joint angles from the EKF and the accelerometer signal were used for template training and observation input for the fixed-window HMM and the DTW. Only the joint angle data were used for the ZVC algorithm as the accelerometer data would introduce noisy ZVCs and degrade the segmentation quality.

The feature-guided HMM outperforms all the comparison algorithms. Table 6.14 shows the feature-guided HMM's error metrics compared against ZVC , fixed-window HMM and feature-guided DTW, at different t_{error} levels. The ZVC algorithm does not have a identification component, but the data are presented together for ease of comparison.

Table 6.14: Identification and segmentation results comparing the proposed to other algorithms, with identification accuracy and percentage, as well as the algorithm's performance within a given error bound t_{error}, and the number of segment point estimates that are correct, false positive (FP) and negative (FN). Separate results are reported for trained and untrained data.

(a) Trained data. Total = 2044.

	Ident.	$\boldsymbol{t_{error} = 0.2}$ s			$\boldsymbol{t_{error} = 0.3}$ s		
	Correct	**Correct**	**FP**	**FN**	**Correct**	**FP**	**FN**
ZVC	-	1256 (61%)	526	786	1414 (69%)	432	628
Fixed HMM	2038 (99.7%)	483 (24%)	948	1559	710 (35%)	644	1332
Feature DTW	2044 (100%)	1658 (81%)	42	384	1822 (89%)	14	220
Feature HMM	2044 (100%)	1684 (82%)	34	358	1854 (91%)	12	188

(b) Untrained data. Total = 2038.

	Ident.	$\boldsymbol{t_{error} = 0.2}$ s			$\boldsymbol{t_{error} = 0.3}$ s		
ZVC	-	1345 (66%)	424	691	1513 (74%)	344	523
Fixed HMM	2028 (99.5%)	397 (20%)	958	1639	572 (28%)	742	1464
Feature DTW	2036 (99.6%)	1556 (76%)	96	482	1736 (85%)	42	302
Feature HMM	2038 (100%)	1568 (77%)	76	470	1768 (87%)	22	270

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

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

It does not appear that DTW warping irregularities is a significant problem for this data set, as feature-guided DTW performs comparably to the feature-guided HMM in both segmentation and identification. Having a multi-tier segmentation algorithm and a way to reduce the search space for potential segment points greatly increases the accuracy of the segmentation algorithm. Due to variations in human motion, it is critical that the algorithm employed can handle spatial and temporal intra- and interpersonal variances between the motion template and observations. HMMs model spatial variations explicitly as state variance, and temporal variations within the state transition matrix, allowing it to identify the observation motion, even if it is carried out faster or slower than the exemplar template. ZVC does not account for these variabilities at all.

6.3.4 Timing Results

Table 6.15 shows the timing results for each of the examined algorithms, averaged over the 198 motion sequences examined. The exemplar length is the average length of each observation sequence. The template construction time is the average time taken to construct all relevant template data for a given subject. The segmentation time is the average time required to segment one set of observation data.

Due to its simplicity, the ZVC algorithm requires no template training time and very little segmentation time. However, as noted in the previous section, this algorithm is inaccurate. Fixed-window HMM uses the Baum-Welch algorithm to train the HMMs based on exemplar data, and requires a significant amount of training time. Its segmentation time is also very long, as it needs to run the forward algorithm numerous times at each time

Table 6.15: Timing results compared to other algorithms, in [s]. Average exemplar length is 37.14 $s \pm 11.03$ s.

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

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

It should be noted that these results were collected from algorithms implemented in MATLAB, which consists of a large overhead due to its high-level structure. These timing results should be examined relative to each other, as opposed to being an absolute assessment. As such, Table 6.15 shows that the feature-guided HMM significantly decreased the cost of utilizing an HMM algorithm as it performed better than the fixed-window HMM. When all other factors were held equal, HMM is seen to outperform DTW, as can be seen by the feature-guided version of both algorithms.

6.4 Summary

The joint angle estimation algorithm proposed in Chapter 4 was verified on a set of movement data from healthy subjects. When compared to motion capture, the approach achieves an average RMS error of 4.27 *cm* for unconstrained motion, with an average joint error of 6.5°. The average RMS error is 3.31 *cm* for sagittal planar motion, with an average joint error of 4.3°.

The segmentation and identification algorithm proposed in Chapter 5 was also verified, on the joint angle estimates produced by the pose recovery algorithm proposed in Chapter

[4] The segmentation algorithm reports 78% accuracy when the template training data comes from the same participant, and 74% for a generic template.

Chapter 7

Conclusions and Future Work

7.1 Summary of Findings

This thesis proposes a rehabilitation exercise monitoring system that provides detailed joint angle measurements and automatically identifies the exercises performed by the patient over each rehabilitation session. The proposed system was tested on 20 healthy subjects, performing 5 rehabilitation exercises.

On-line pose estimation from body-worn sensors. This thesis proposed an algorithm for estimating lower body pose based on the extended Kalman filter. Inertial measurement units are employed as sensory input, with constant-acceleration and forward kinematics as the state and measurement models, respectively. These sensor units are well suited for rehabilitation as they are small and lightweight, and do not obstruct the rehabilitation process. Kinematic constraints, based on anthropometric joint angle ranges, are applied to mitigate the effects of gyroscopic drift. This is introduced to the state estimates via a potential field, which exerts a virtual acceleration that varies in strength proportional to the distance between the estimated joint angles and the joint limits. The noise covariances are adjusted on-line to adapt the Kalman filter to each participant and movement profile. The proposed system achieves an overall average RMS error of 4.27 *cm*, and is capable of estimating joint angles accurately for arbitrary 3D motion of the leg.

On-line exercise segmentation and identification. This thesis proposed a feature-guided HMM motion segmentation algorithm to segment human motion data during on-line measurement. The proposed approach reduces the comparison space of the observation data by identifying significant DOFs, velocity peaks and ZVCs to estimate locations of potential segment points. When such potential segment points are located, the sequence under examination is checked against HMM templates to find the most accurate segmentation and identify the motion. The algorithm is shown to be able to segment data with high accuracy with subject-specific templates. It was also shown to be robust when inter-person templates are utilized, demonstrating its ability to handle both spatial and temporal variability introduced by intra and inter-personal differences. The algorithm is capable of handling any time series input stream, including joint angles or Cartesian linear acceleration data. The ability to segment on-line enables immediate feedback to the physiotherapist and rehabilitation participant, allowing the participant to correct incorrect motions to maximize rehabilitation efficacy. The algorithm reports 78% accuracy when the template training data comes from the same participant, and 74% for a generic template.

7.2 Future Work

7.2.1 Data Collection on Rehabilitation Patients

For future work, motion data will be collected from joint replacement physiotherapy patients, in order to assess the system's accuracy against a test population whose motions may have different noise characteristics from a healthy population. Patients undergoing physiotherapy may exhibit significantly different movement profiles, including increased variability and tremor, which may increase the difficulty of joint angle recovery and segmentation.

7.2.2 Pose Estimation

For the pose estimation, improvements to the noise covariance estimation are required to improve the robustness of the algorithm. Increasing the noise covariance when the Kalman filter performs poorly allows the filter to continue to operate instead of diverging, but implies that the filter was running with subpar parameters. A more targeted auto-tuning system would also improve the gyroscopic drift issue. Although the Kalman filter was shown to perform with less reported error when compared to previous work, drift issues still remain. A better noise covariance estimation would allow the Kalman filter to account for the inaccuracies in the sensor calibration, produce more accurate joint angle estimates, and potentially eliminate the need for the joint constraint potential field.

Currently, the Kalman filter utilizes link lengths derived from motion capture. A substitute for motion capture values is to use anthropometric tables, but anthropometric tables are constructed for a specific population and may not be applicable to all populations. A method for link length parameter estimation would remove the dependency on motion capture or anthropometric tables, and provide a better estimation routine than using a fixed link length.

7.2.3 Motion Segmentation and Identification

For the motion segmentation and identification, a method to dynamically update the template could improve segmentation results. Motion templates generated from healthy participants or physiotherapist may not correspond well to the motions produced by patients undergoing physiotherapy. Template adaptation would allow the templates to be modified by incorporating information from the observation itself, enabling the segmentation and identification of a wider range of motions.

The application of this thesis is to exercise motions, and exercise motions tend to be cyclic in the joint velocity, allowing for velocity crossings and velocity peaks to be used. However, more powerful or sophisticated features may need to be proposed for general motions sets. Currently, only relative positions of peaks are used, and timing data use was minimal. Improved feature identification would further decrease the number of forward algorithm calls, and thus improve runtime as well.

References

[1] J. A. Afonso, H. R. Silva, P. M. Oliveira, J. H. Correia, and L. A. Rocha. Design and implementation of a real-time wireless sensor network. In *Proceedings of the International Conference on Sensor Technologies and Applications*, pages 496–501, 2007.

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

[3] M. Ayoade, L. Morton, and L. Baillie. Investigating the feasibility of a wireless motion capture system to aid in the rehabilitation of total knee replacement patients. In *Proceedings of the International Conference on Pervasive Computing Technologies for Healthcare*, pages 404–407, 2011.

[4] S. Baby and V. Krüger. Primitive based action representation and recognition. In Arnt-Brre Salberg, Jon Hardeberg, and Robert Jenssen, editors, *Image Analysis*, volume 5575 of *Lecture Notes in Computer Science*, pages 31–40. Springer Berlin / Heidelberg, 2009.

[5] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, pages 185–194, 2004.

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

[7] J. J. Baskett, J. B. Broad, G. Reekie, C. Hocking, and G. Green. Shared responsibility for ongoing rehabilitation: a new approach to home-based therapy after stroke. *Clinical Rehabilitation*, 13:23–33, 1999.

[8] J. Bergmann, R. Mayagoitia, and I. Smith. A portable system for collecting anatomical joint angles during stair ascent: a comparison with an optical tracking device. *Dynamic Medicine*, 8:1–7, 2009.

[9] J. Bergmann and A. McGregor. Body-worn sensor design: What do patients and clinicians want? *Annals of Biomedical Engineering*, 39:2299–2312, 2011.

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

[11] M. C. Boonstra, R. M. A. van der Slikke, N. L. W. Keijsers, R. C. van Lummel, M. C. de Waal Malefijt, and N. Verdonschot. The accuracy of measuring the kinematics of rising from a chair with accelerometers and gyroscopes. *Journal of Biomechanics*, 39:354–358, 2006.

[12] R. S. Bridger. *Introduction to Ergonomics, Third Edition*. CRC Press, Boca Raton, FL, 2008.

[13] S. P. Buerger, J. J. Palazzolo, H. I. Krebs, and N. Hogan. Rehabilitation robotics: adapting robot behavior to suit patient needs and abilities. In *Proceedings of the American Control Conference*, volume 4, pages 3239–3244, 2004.

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

[15] S. Calinon, F. D'halluin, E. L. Sauser, D.G. Caldwell, and A. G. Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics Automation Magazine*, 17(2):44–54, 2010.

[16] Canadian Joint Replacement Registry. Total hip and total knee replacements in canada, 2005.

[17] F. Cavallo, M. Aquilano, L. Odetti, M. Arvati, and M.C. Carrozza. A first step toward a pervasive and smart zigbee sensor system for assistance and rehabilitation. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics*, pages 632–637, 2009.

[18] J. Cham. Piled higher and deeper (phd). www.phdcomics.com, 1997.

[19] E. Y. S. Chao. Justification of triaxial goniometer for the measurement of joint rotation. *Journal of Biomechanics*, 13:989–1006, 1980.

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

[21] C. A. Cott, R. M. A. Devitt, L. Falter, L. J. Soever, and L. A. Passalent. Barriers to rehabilitation in primary health care in ontario: Funding and wait times for physical therapy services. *Physiotherapy Canada*, 3:173–183, 2007.

[22] Crossbow Technology. Micaz mesh network radio module datasheet.

[23] W. Dong, I. M. Chen, K. Y. Lim, and Y. K. Goh. Measuring uniaxial joint angles with a minimal accelerometer configuration. In *Proceedings of the International Convention on Rehabilitation Engineering and Assistive Technology*, pages 88–91, 2007.

[24] J. Z. Edwards, K. A. Greene, R. S. Davis, M. W. Kovacik, D. A. Noe, and M. J. Askew. Measuring flexion in knee arthroplasty patients. *Journal of Arthroplasty*, 19:369–372, 2004.

[25] J. Eriksson, M. J. Matarić, and C. J. Winstein. Hands-off assistive robotics for post-stroke arm rehabilitation. In *Proceedings of the IEEE International Conference of Rehabilitation Robotics*, pages 21–24, 2005.

[26] R. J. Farris, H. A. Quintero, and M. Goldfarb. Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(6):652–659, 2011.

[27] P. Fergus, K. Kafiyat, M. Merabti, A. Taleb-bendiab, and A. El Rhalibi. Remote physiotherapy treatments using wireless body sensor networks. In *Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 1191–1197, 2009.

[28] F. Ferraris, U. Grimaldi, and M. Parvis. Procedure for effortless in-field calibration of three-axial rate gyro and accelerometers. *Sensors and Materials*, 7:311–330, 1995.

[29] J. R. Flanagan and A. K. Rao. Trajectory adaptation to a nonlinear visuomotor transformation: evidence of motion planning in visually perceived space. *Journal of Neurophysiology*, 74(5):2174–2178, 1995.

[30] A. Fod, M. J. Matarić, and O. C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 2002.

[31] V. F. S. Fook, S. Z. Hao, A. A. P. Wai, M. Jayachandran, J. Biswas, L. S. Yee, and P. Yap. Innovative platform for tele-physiotherapy. In *Proceedings of the International Conference on e-health Networking, Applications and Services*, 2008.

[32] E. Garcia, M.A. Jimenez, P.G. de Santos, and M. Armada. The evolution of robotics research. *IEEE Robotics Automation Magazine*, 14(1):90–103, 2007.

[33] M. A. Giese and T. Poggio. Synthesis and recognition of biological motion patterns based on linear superposition of prototypical motion sequences. In *Proceedings of the IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pages 73–80, 1999.

[34] C. Gong, J. Yuan, and J. Ni. Nongeometric error identification and compensation for robotic system by inverse calibration. *International Journal of Machine Tools and Manufacture*, 40(14):2119–2137, 2000.

[35] A. E. Gross. Commentary on an article by trevor g. russell, md, et al.: internet-based outpatient telerehabilitation for patients following total knee arthroplasty. a randomized controlled trial. *Bone*, 93:E6 1–2, 2011.

[36] A. Hadjidj, Y. Challal, and A. Bouabdallah. Toward a high-fidelity wireless sensor network for rehabilitation supervision. In *Proceedings of the 36th IEEE Conference on Local Computer Networks*, pages 458–465, 2011.

[37] J. Hidler, D. Brennan, I. Black, D. Nichols, K. Brady, and T. Nef. Zerog: overground gait and balance training system. *Journal of Rehabilitation Research and Development*, 48:287–98, 2011.

[38] J. Hidler and P. S. Lum. The road ahead for rehabilitation robotics. *Journal of Rehabilitation Research and Development*, 48:vii–x, 2011.

[39] W. T. Higgins. A comparison of complementary and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11:321–325, 1975.

[40] J. A. Howe, E. L. Inness, A. Venturini, J. I. Williams, and M. C. Verrier. The community balance and mobility scale-a balance measure for individuals with traumatic brain injury. *Clinical Rehabilitation*, 20:885–895, 2006.

[41] J. Hu, M. K. Brown, and W. Turin. Hmm based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.

[42] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1398–1403, 2002.

[43] W. Ilg, G. H. Bakir, J. Mezger, and M. A. Giese. On the representation, learning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics*, 1(4):613–636, 2004.

[44] M. Ito and J. Tani. On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, 12(2):93–115, 2004.

[45] M. Jagodzinski, V. Kleemann, P. Angele, V. Schnhaar, K.W. Iselborn, G. Mall, and M. Nerlich. Experimental and clinical assessment of the accuracy of knee extension measurement techniques. *Knee Surgery, Sports Traumatology, Arthroscopy*, 8:329–336, 2000.

[46] B. Janus and Y. Nakamura. Unsupervised probabilistic segmentation of motion data for mimesis modeling. In *Proceedings of the 12th International Conference on Advanced Robotics*, pages 411–417, 2005.

[47] E. J. R. Justino, F. Bortolozzi, and R. Sabourin. Off-line signature verification using hmm for random, simple and skilled forgeries. In *Proceedings of the 6th International Conference on Document Analysis and Recognition*, pages 1031–1034, 2001.

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

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

[50] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of the 1st SIAM International Conference on Data Mining*, 2001.

[51] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 500–505, 1985.

[52] H. Kim, L. Miller, I. Fedulow, M. Simkins, G. Abrams, N. Byl, and J. Rosen. Kinematic data analysis for post stroke patients following bilateral versus unilateral rehabilitation with an upper limb wearable robotic system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 99:(pre–printing), 2012.

[53] A. D. King. Inertial navigation–forty years of evolution. *General Electric Company Review*, 3:140–149, 1998.

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

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

[56] H. Krebs, M. Ferraro, S. Buerger, M. Newbery, A. Makiyama, M. Sandmann, D. Lynch, B. Volpe, and N. Hogan. Rehabilitation robotics: pilot trial of a spatial extension for mit-manus. *Journal of NeuroEngineering and Rehabilitation*, 1(1):5–20, 2004.

[57] D. Kulić, W. Takano, and Y. Nakamura. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Transactions on Robotics*, 25(5):1158–1166, 2009.

[58] M. D. Landry, S. B. Jaglal, W. P. Wodchis, N.S. Cooper, and C. A. Cott. Rehabilitation services after total joint replacement in ontario, canada: can 'prehabilitation' programs mediate an increasing demand? *International Journal of Rehabilitation Research*, 30:297–303, 2007.

[59] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[60] G. X. Lee, K. S. Low, and T. Taher. Unrestrained measurement of arm motion based on a wearable wireless sensor network. *IEEE Transactions on Instrumentation and Measurement*, 59(5):1309–1317, 2010.

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

[62] J. F. S. Lin and D. Kulić. Human pose recovery using wireless inertial measurement units. *Physiological Measurement*, 33(12):2099–2115, 2012.

[63] J. F. S. Lin and D. Kulić. On-line segmentation of human motion for automated rehabilitation exercise analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, (in submission).

[64] K. Lorincz, B. R. Chen, G. W. Challen, A. R. Chowdhury, Shyamal P., P. Bonato, and M. Welsh. Mercury: A wearable sensor network platform for high-fidelity motion analysis. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 2009.

[65] K. S. Low, G. X. Lee, and T. Taher. A wearable wireless sensor network for human limbs monitoring. In *Proceedings of the Instrumentation and Measurement Technology Conference*, pages 1332–1336, 2009.

[66] E. C. Lu, R. H. Wang, D. Hebert, J. Boger, M. P. Galea, and A. Mihailidis. The development of an upper limb stroke rehabilitation robot: identification of clinical practices and design requirements through a survey of therapists. *Disability and Rehabilitation: Assistive Technology*, 6(5):420–431, 2011.

[67] H. Luinge and P. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and Computing*, 43:273–282, 2005.

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

[69] A. Lymberis. Smart wearables for remote health monitoring, from prevention to rehabilitation: current research and development, future challenges. In *Proceedings of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine*, pages 272–275, 2003.

[70] L. Mertz. The next generation of exoskeletons: Lighter, cheaper devices are in the works. *IEEE Pulse*, 3(4):56–61, 2012.

[71] T. B. Moeslund, A. Hilton, and V. Krger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[72] Motion Analysis Corporation. *Cortex 2.0 Reference Manual*, 2010.

[73] K. Murphy. Bayes net toolbox for matlab. code.google.com/p/bnt/, 1998. Last accessed on 2012/04/27.

[74] C. Myers, L. Rabiner, and A. Rosenberg. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 173–177, 1980.

[75] C. C. Norkin and D. J. White. *Measurement of Joint Motion: A Guide to Goniometry, 4th Edition.* F. A. Davis Company, Philadelphia, PA, 2009.

[76] A. M. Okamura, M. J. Mataric, and H. I. Christensen. Medical and health-care robotics. *IEEE Robotics Automation Magazine*, 17(3):26–37, 2010.

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

[78] J. C. Perry, J. Rosen, and S. Burns. Upper-limb powered exoskeleton design. *IEEE/ASME Transactions on Mechatronics*, 12(4):408–417, 2007.

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

[80] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

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

[82] D. Roetenberg, C. T. M. Baten, and P. H. Veltink. Estimating body segment orientation by applying inertial and magnetic sensing near ferromagnetic materials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15:469–472, 2007.

[83] D. Roetenberg, H. Luinge, and P. J. Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Technical report, Xsens Technologies, 2009.

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

[85] T. G. Russell, P. Buttrum, R. Wootton, and G. A. Jull. Internet-based outpatient telerehabilitation for patients following total knee arthroplasty: A randomized controlled trial. *Bone and Joint Surgery*, 93:113–120, 2011.

[86] T. Sakaki, S. Okada, Y. Okajima, N. Tanaka, A. Kimura, S. Uchida, M. Taki, Y. Tomita, and T. Horiuchi. Tem: Therapeutic exercise machine for hip and knee joints of spastic patients. In *Proceedings of the International Conference on Rehabilitation Robotics*, 1999.

[87] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Speech and Signal Processing*, 26(1):43–49, 1978.

[88] A. Salarian, F. B. Horak, C. Zampieri, P. Carlson-Kuhta, J. G. Nutt, and K. Aminian. itug, a sensitive and reliable measure of mobility. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18:303–310, 2010.

[89] R. B. Salter, D. F. Simmonds, B. W. Malcolm, E. J. Rumble, D. MacMichael, and N. D. Clements. The biological effect of continuous passive motion on the healing of full-thickness defects in articular cartilage. an experimental investigation in the rabbit. *The Journal of Bone and Joint Surgery. American Volume*, 62:1232–51, 1980.

[90] Y. Sankai. Hal: Hybrid assistive limb based on cybernics. In *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 25–34. Springer Berlin / Heidelberg, 2011.

[91] C. Schabowsky, S. Godfrey, R. Holley, and P. Lum. Development and pilot testing of hexorr: Hand exoskeleton rehabilitation robot. *Journal of NeuroEngineering and Rehabilitation*, 7(1):36, 2010.

[92] M. A. Schwartz. The importance of stupidity in scientific research. *Journal of Cell Science*, 121(11):1771, 2008.

[93] M. W. Spong, S. Hutchinson, and M Vidyasagar. *Robot Modeling and Control*. John Wiley and Sons, Inc., New York, NY, 2006.

[94] E. Strickland. Good-bye, wheelchair. *IEEE Spectrum*, 49(1):30–32, 2012.

[95] M. Tousignant, P. Boissy, H. Corriveau, and H. Moffet. In home telerehabilitation for older adults after discharge from an acute hospital or rehabilitation unit: A proof-of-concept study and costs estimation. *Disability and Rehabilitation: Assistive Technology*, 1(4):209–216, 2006.

[96] M. Tousignant, P. Boissy, H. Corriveau, H. Moffet, and F. Cabana. In-home telerehabilitation for post-knee arthroplasty: A pilot study. *International Journal of Telerehabilitation*, 1:9–16, 2009.

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

[98] J. J. Jr. Uicker, J. Denavit, and R. S. Hartenberg. An iterative method for the displacement analysis of spatial mechanisms. *Journal of Applied Mechanics*, 31(2):309–314, 1964.

[99] A. Valtazanos, D. K. Arvind, and S. Ramamoorthy. Comparative study of segmentation of periodic motion data for mobile gait analysis. In *Wireless Health*, pages 145–154, 2010.

[100] V. van Acht, E. Bongers, N. Lambert, and R. Verberne. Miniature wireless inertial sensor for measuring human motions. In *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6278–6281, 2007.

[101] John Varkey, Dario Pompili, and Theodore Walls. Human motion recognition using a wireless sensor-based wearable system. *Personal and Ubiquitous Computing*, pages 1–14, 2011.

[102] E. A. Wan and A. T. Nelson. *Kalman Filtering and Neural Networks*, chapter Dual EKF Methods, pages 123–173. John Wiley and Sons, Inc., New York, NY, 2001.

[103] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585 – 601, 2003.

[104] G. Welch and G. Bishop. An introduction to kalman filters. Technical report, University of North Carolina at Chapel Hill, 2006.

[105] D. Wigdor, S. Williams, M. Cronin, R. Levy, K. White, M. Mazeev, and H. Benko. Ripples: utilizing per-contact visualizations to improve user interaction with touch displays. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 3–12, 2009.

[106] R. Williamson and B. J. Andrews. Detecting absolute human knee angle and angular velocity using accelerometers and rate gyroscopes. *Medical and Biological Engineering and Computing*, 39:294–302, 2001.

[107] J. M. Winters, Y. Wang, and J. M. Winters. Wearable sensors and telerehabilitation. *IEEE Engineering in Medicine and Biology Magazine*, 22:56–65, 2003.

[108] L. Yardley, N. Beyer, K. Hauer, G. Kempen, C. Piot-Ziegler, and C. Todd. Development and initial validation of the falls efficacy scale-international. *Age and Ageing*, 34(6):614–19, 2005.

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

[110] H. Zhou, T. Stone, H. Hu, and N. Harris. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical Engineering and Physics*, 30:123–133, 2008.

[111] R. Zhu and Z. Zhou. Realtime articulated human motion tracking using tri-axis inertial/magnetic sensors package. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12:295–302, 2004.