2014

Acquisition and distribution of synergistic reactive control skills

Matthew Field
University of Wollongong

Acquisition and distribution of synergistic reactive control skills

A thesis submitted in fulfilment of the

requirements for the award of the degree

Doctor of Philosophy

from

University of Wollongong

by

Matthew Field, B.Eng(Electrical)

School of Electrical, Computer and Telecommunications Engineering

October 2014

Dedicated to my family...

Acknowledgements

I would like to thank first and foremost my family and friends for their continued support throughout my studies.

I must thank my supervisor David Stirling for his mentorship over the years. The continued guidance he has provided has been of great influence on me. I also thank my co-supervisors Zengxi Pan and Fazel Naghdy for their professional scientific advice. I have learned a great deal about engineering from each of my supervisors.

I send my thanks to all my fellow research students who shared good conversations, social sport and kindly helped by participating in experiments. I could not have enjoyed the studies as much as I did without their company.

I would like to thank Marnie, Karl and Damien for their assistance, ideas and enthusiasm for our experimental set up. I enjoyed the experiments conducted with them and hope that collaborations continue on with our research group into the future.

Next I thank Scott and Jeremy for their enthusiasm for applied motion capture and their support for our experiments.

I also appreciate the support from all of the engineering staff including administration and teaching work from Sasha, Brian, and Roslyn and the technical workshop staff.

Certification

I, Matthew Field, declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is entirely my own work unless otherwise referenced or acknowledged. This manuscript has not been submitted for qualifications at any other academic institute.

Matthew Field

Date: 7 September 2014

Abstract

Learning from demonstration is an efficient way to attain a new skill. In the context of autonomous robots, using a demonstration to teach a robot accelerates the robot learning process significantly. It helps to identify feasible solutions as starting points for future exploration or to avoid actions that lead to failure. But the acquisition of pertinent observations is predicated on first segmenting the data into meaningful sequences. These segments form the basis for learning models capable of recognising future actions and reconstructing the motion to control a robot. Furthermore, learning algorithms for generative models are generally not tuned to produce stable trajectories and suffer from parameter redundancy for high degree of freedom robots.

This thesis addresses these issues by firstly investigating algorithms, based on dynamic programming and mixture models, for segmentation sensitivity and recognition accuracy on human motion capture data sets of repetitive and categorical motion classes. A stability analysis of the non-linear dynamical systems derived from the resultant mixture model representations aims to ensure that any trajectories converge to the intended target motion as observed in the demonstrations. Finally, these concepts are extended to humanoid robots by deploying a factor analyser for each mixture model component and coordinating the structure into a low dimensional representation of the demonstrated trajectories. This representation can be constructed as a correspondence map is learned between the demonstrator and robot for joint space actions.

Applying these algorithms for demonstrating movement skills to robot is a further step towards autonomous incremental robot learning.

Publications Arising from the Thesis

1. M. Field, D. Stirling, Z. Pan, and F. Naghdy. Learning trajectories for robot programing by demonstration using a coordinated mixture of factor analyzers. Cybernetics, IEEE Transactions on, PP(99):1-1, 2015.

2. M. Field, D. Stirling, Z. Pan, M. Ros, and F. Naghdy. Recognizing human motions through mixture modeling of inertial data. Pattern Recognition, 48(8):2394-2406, 2015.

3. M. Field, D. A. Stirling, M. Ros and Z. Pan, "Inertial sensing for human motor control symmetry in injury rehabilitation," In Advanced Intelligent Mechatronics, 2013. AIM 2013. IEEE International Conference on, pages 1470-1475, July 2013.

4. M. Field, Z. Pan, D. A. Stirling and F. Naghdy, "Human motion capture sensors and analysis in robotics," Industrial Robot, vol. 38, (2) pp. 163-171, 2011.

5. D. A. Stirling, F. Naghdy, G. Naghdy, M. Field, R. Arunglabi and D. Kilpatrick, "Objective functional capacity assessment using inertial sensor," in Proceedings 2011 1st IEEE International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB), 2011, pp. 272-277.

6. M. Field, Z. Pan, D. A. Stirling and F. Naghdy, "Motion capture in robotics review," in 2009 IEEE International Conference on Control and Automation, 2009, pp. 1697-1702.

7. M. Field, D. A. Stirling, F. Naghdy and Z. Pan, "Motion segmentation for humanoid control planning," in Australasian Conference on Robotics and Automation, 2008.

8. M. Field, D. A. Stirling, F. Naghdy and Z. Pan, "Mixture model segmentation for gait recognition," in The 2008 ECSIS Symposium on Learning and Adaptive Behaviour in Robotic Systems, 2008, pp. 3-8.

9. M. Field, D. A. Stirling, F. Naghdy and Z. Pan, "Empirical modelling of human gaits for biped robots," in Australasian Conference on Robotics and Automation, 2007, pp. 1-7.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The design of autonomous systems is generally contingent on expert knowledge of the problem domain in question. The acquisition, interpretation and deployment of an expert skill is a continual challenge in the fields of control, automation, machine learning and robotics. A central issue hindering the design cycle of autonomous systems is the time and effort required to encode a new capability. Not only does it require a technical specialist in a particular programming framework but also detailed knowledge of the problem domain in order to outline a suitable action for each state of the system. In many cases, the set of decision rules derived by the expert through introspection are found to be incomplete, partially inaccurate or sub-optimal [4]. Alternatively, by observing a set of expert demonstrations, the state-action relationships can be extracted from the data automatically using machine learning or data mining techniques. This approach is commonly referred to as programming by demonstration (PbD) [5], behavioural cloning [6] or apprenticeship learning [7].

The main advantage of PbD is that it eliminates the need for technical knowledge in the programming of control algorithms. That is to say, it enables a lay person to endow a robot with a new capability. However, multiple challenges arise when constructing a framework within the PbD paradigm. A crucial factor in the success

of the resultant program is how *robust* it is to disturbances or changes in the task context. When executing an assigned task, the demonstrated path for a robot may be obstructed by unforeseen environmental constraints or perturbed by external forces. For the transference of motion skills to a robot there is a *correspondence* problem due to differences in the kinematic structure between the demonstrator and the target system. For instance, in the context of minimally invasive surgery the fine movements of the surgeon are translated to the actuation commands of a miniature robot. Lastly, the *transparency* of a system gauges how easily the structure of a learnt program can be interpreted and subsequently fine-tuned. In learning a control system for high risk applications the structure of the program should be checked for the possibility of critical failures. If the program is structured in an interpretable way it can be evaluated by an expert. The eventual success of the transfer of expert skill can be assessed by addressing these general concepts in the composition of control programs.

A skill, for the purposes of this work, is the cumulation of decisions or actions an expert takes to solve a set of similar problems at minimum cost, where the cost function may constitute a measure of time, energy or target deviation. The acquisition of a skill entails the collection of state-action examples and the derivation of an associated cost function. Research into the robotic learning of motion skills has focused on two levels of control. High-level procedural skills which describe the logical sequence of gross decision-making and are typically learnt through inductive reasoning techniques [8]. Low-level trajectory skill which handles the continuous control of a robot motion and where each action may involve the selection of a particular velocity at each time interval compensating for disturbances and constraints [9].

Recent research in PbD has proposed motion skill descriptions as a composition of fundamental motion primitive units [10]. Each motion primitive is interpreted either as an observed trajectory as part of skill library or as an indivisible component of a

particular trajectory. This approach is, in part, inspired by the neurophysiological evidence for mirror neurons [11]. These neurons are considered important for imitation because they activate when the body performs a particular action and also when observing the same action performed by a different body. They are thereby considered to play a role in both the recognition of a movement as well as the activation of the required muscles to perform the same movement. By expressing a trajectory in this discretized motion primitive form inductive procedural skill learning techniques can be exploited for motion planning and inferring motion intention. Furthermore, by deriving the motion primitive modules directly from exemplars one can leverage the inherent human knowledge of the task without unintended biases from manually derived controllers.

This thesis is primarily focused on the transfer of modularized reactive control skills to anthropomorphic robots via demonstrations. The approach presented here expands on recent research into the use of motion primitives in robot programming by demonstration. In the remaining sections of this introduction the overall aim and associated research questions are stated with further elaboration of the context of the research. A description of the contributions of the thesis with respect to the latest research is then presented and is then followed by an outline of the dissertation.

1.1 Aim and Research Questions

A control skill implies there is an ability to adapt to changing circumstances. In a teleoperated system the objective is rather to directly track an observed path. However, this extinguishes any provision for adaptation of a trajectory and any permitted tracking variance is therefore constant along the path. If additional observations are gathered, as shown in Figure 1.1, the variance along a trajectory can be represented and regions of restricted variance may indicate crucial task constraints that were

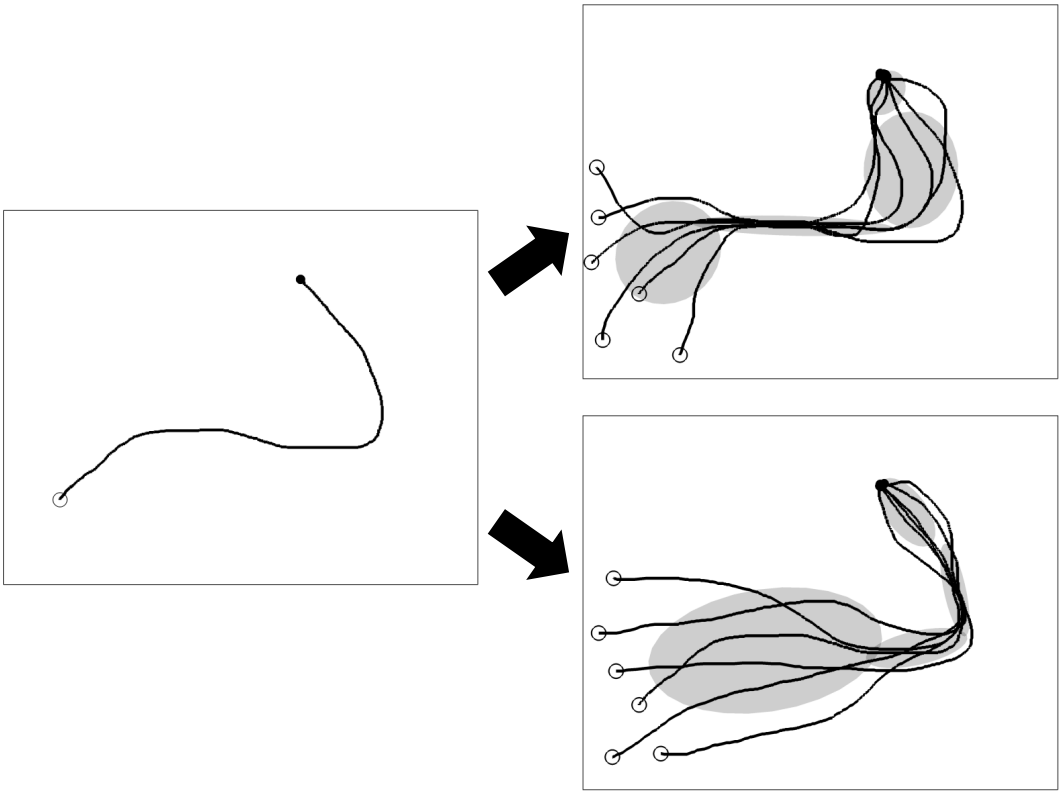

Figure 1.1: An illustration of the significance of learning from multiple demonstrations of a two dimensional path. After a single demonstration (left) one is constrained to replicate the trajectory, from the starting location (circle) to the terminating point. Further demonstrations provide information about the variance along this trajectory (right) thereby allowing variation in the reproduction. Ellipses correspond to a localised single standard deviation and where regions of minimal variance are commonly related to the task constraints.

otherwise unknown or must be explicitly programmed.

Ultimately the aim of this research is to automatically derive a repertoire of motion skill models from observed data to be managed in a skill vocabulary which, upon request of a particular task, can generate stable robot motion trajectory capable of adapting due to constraints. This aim entails a complex system which can be parcellated into a number of major components summarized in Figure 1.2.

Firstly, a manifestation of motion skill is observed with a suite of sensor technologies which measure the trajectory of the system. As a continuous stream of data representing the demonstrated action is observed, this motion needs to be organised

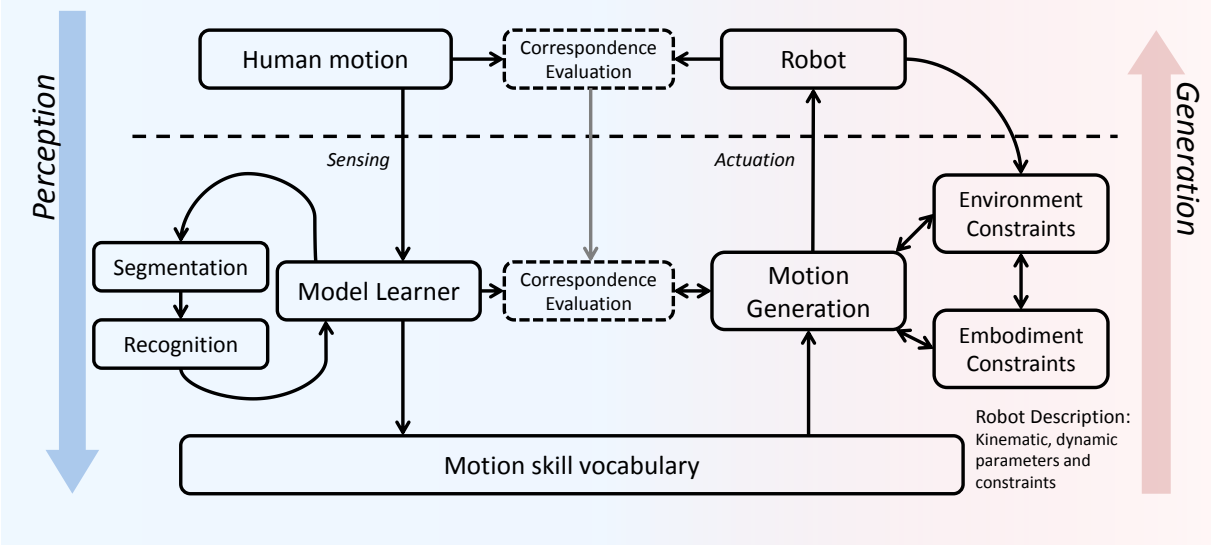

Figure 1.2: Overview of a robot programming by demonstration architecture.

into a framework based on the relevance to a set of tasks. A crucial component of the perception of motion is the formation of an internal model which efficiently represents the motion and is used as a discriminative tool to recognize the same motion in future observations. The perceptual model approach in this thesis, as depicted in Figure 1.2, is to form a model of the motion with unsupervised learning algorithms and use this representation of the data to separate the trajectories into time-series segments. These sequences are subsequently classified as either related to previous observations or as an initial observation of a new skill. The key factors of the perceptual model are therefore the parsimony of the representation, that is, the ratio between the number of parameters of the model to the number of observations the model represents and the discriminative power of the model.

To use observed skills to control a robot a mechanism for reconstructing the trajectory from the representative model is needed. This may take the form of extracting the key way-points for the task as displayed in the data or a model of the desired velocity at each point in the workspace. However, in generating motion for a robot frame one cannot always expect to reconstruct the exact path. There are practical

constraints which affect the feasibility of a planned robot motion. The trajectory may be inhibited by environmental factors such as an obstacle impeding the path or uncertainties, due to sensor variance, in the desired target location. In addition there are often constraints due to the robot embodiment model. A motion may be limited by discrepant degrees of freedom, joint actuator limits, kinematic singularities or the calculation of self-collisions. At the same time, any modification of the motion due to the satisfaction of these constraints, must match the demonstration as closely as possible through a similarity evaluation or imitation metric. The problem therefore, consists of balancing the satisfaction of a set of constraints with a faithful emulation of the demonstrated motion.

There are clearly many components to the implementation of a PbD system with minimal human involvement. The general research questions considered in this thesis are aligned with the concepts of motion perception and generation:

- How can a robot autonomously segment and recognize similar trajectories from continuous data and incrementally update a representative model?

- How can an autonomous agent generate actions from the learned internal model in a useful manner for its corresponding embodiment?

These fundamental questions, however, can be partitioned into several specific problems, which, in themselves, are complex in the field of machine learning.

The design of perceptual models has focused on maximising the reliability of discriminating between classes of motions. The classes of interest are typically defined in advance and significant manual preparation of the data is required to train a model. An autonomous robot observer, however, is gathering data continuously and must segment, interpret and update, if necessary, the set of available skills. The question is then, how can a system organize data when little or no contextual information is

available and how do current algorithms perform under these circumstances? Other aspects to consider concern the metric of the extent to which a model is representative of the data, how many features are required to capture the fundamental motion or which feature transformations can efficiently compress the data.

Similarly, the generation of motion from perceptual models also faces more specific questions. For instance, in motion generation models depending on the system in question many conditions may constrain the resulting motion. There must be a systematic method to incorporating all constraints in a generating cost function which allows flexibility in the path when it exists in the demonstrations but satisfies important constraints concerning the physical model of the robot. Using known constraints and the model information is it possible to combine forward planning with the motion generation methods?

1.2 Contribution

The contributions presented in this thesis address the aforementioned research questions by presenting:

1. *A system for the segmentation and classification of continuous observed motion.* An incremental segmentation and recognition approach is proposed for models starting with limited prior information and assumptions. The segmentation algorithm is unique in that it uses the model information to partition the data space by analysing the frequency of motion primitives and combining similar sequences of motion primitives taking advantage of repeated demonstrations.

2. *An algorithm for generating a stable and robust motion from an internal model which has a high similarity with the observed data.* The approach presented uses an explicit similarity measure to optimize the correlation with the observations

subject to constraints. A forward planning module is considered to avoid violating static constraints that inhibit the reconstruction of a trajectory. Control laws are proposed to ensure that the learnt motion primitive model generates stable motion for terminal trajectories and oscillatory trajectories.

3. *System for stable humanoid motion generation in a low dimensional embedding which considers the correspondence constraints of the target body.* A robot system is controlled through a low dimensional embedded coordinate frame learnt from a series of demonstrations and to which robot constraints are mapped. The dynamics along the manifold coordinates are tuned to produce stable trajectories including oscillatory motion such as a gait. The motion primitive control methods are shown to be an effective mechanism in translating skills to humanoids.

1.3 Outline of the Thesis

A description of the contents of the remaining chapters is given below:

In Chapter 2, a literature review is presented which initially describes the concept of human motor control and the research into proposed control mechanisms that explain human motion. It then proceeds to evaluate the current state of motion sensors and the suitability in the fields of gesture recognition and robot PbD. The interpretation and modeling of motion data is reviewed but with the emphasis on time-series segmentation, motion recognition and models which generate or reconstruct the observations. The focus is then turned to robotics where the concepts and general solutions to the correspondence problem are reviewed. Finally the recent research in robot PbD is detailed.

In Chapter 3, the problem of time-series segmentation as applied to areas of human motion analysis is introduced with real experiments in detecting meaningful posture changes. Firstly, a number of preliminaries pertaining to the theory of motion primitives and how they are applied in the context of segmentation and classification are outlined. This is followed by descriptions of the experimental sensor system and kinematic parameter notation. Lastly, two experimental case studies exemplify the application of these concepts to real problems in interpreting human motion. The case studies involve the tracking of injury rehabilitation and the automatic assessment of human posture in clinical tests.

In Chapter 4, the proposed methods of time-series segmentation and recognition of trajectories are presented as the perceptual component of the imitation learning system. The algorithm proposed is based on interpreting the model structure from unsupervised learning and using this model description to partition into sets of similar trajectories. In this case, the experimental data includes an arbitrary set of full body motion capture actions. Initially the approach is assessed on how accurately the classes of motion are automatically parcellated without prior class label examples. It is further evaluated when example labels are supplied and is compared with other common classification techniques.

In Chapter 5, the proposed modelling methods are extended to the problem of motion generation and the subsequent mapping to a robot. A motion generation method is examined and tested on an example robot which tracks a 2 dimensional path. The proposed modifications of the algorithm produce stable oscillations and an analysis of the stability of the resultant system is provided. The evaluation metrics tested in this domain the similarity to demonstrated paths, the avoidance of obstacles, forward planning for static obstacles.

In Chapter 6, the robot programming approach is further extended into the field

of humanoid robotics. The problems addressed are the correspondence between the human and robot, the extension of the models into higher degrees of freedom with the use of dimensionality reduction, and the incorporation of constraints into the motion generation such as angular limits and dynamical system stability. A validation of the approach is presented with the transfer of full body motion to a humanoid robot, where the robot is controlled from a learned dynamical system in a reduced dimensional space.

Finally, Chapter 7 summarises the major outcomes of the work in the thesis, and makes recommendations for future work.

Chapter 2

Literature Review

Autonomous control systems are expected to perform in a reliable and robust manner across the assigned state space. Traditional control techniques involve modelling the system dynamics, designing a theoretical controller and implementing the control with minor manual or adaptive tuning. As systems become more complex with increasing multidimensional control variables, the system dynamics may not be completely known or require advanced mathematical models that are still limited by assumptions. For this reason research into new control schemes often make use of machine learning techniques to learn and adapt through interaction with the process.

A common approach to deriving controllers is to reconstruct a skill from monitoring human demonstrations, a technique otherwise known as imitation learning using machine learning. Humans can control many highly complicated processes with sufficient training and practice. Human control, when explicitly summarised by experts, is often very procedural but also involves an important element of qualitative judgement embedded as a sub-cognitive skill [12]. Skills are thereby not accurately expressed solely through introspection. Rather learning the procedure of a skilled action from a series of observations provides an avenue for evidence-based model construction. Previous research in this area has focused on problem domains such as automatic

control of flight simulators [6], operation of cranes [13] and humanoid robot imitation [14].

A principal focus of this thesis is to use the display of human movement as the manifestation of a psychomotor skill and to reconstruct the movement in an autonomous robot. The motivation for pursuing this problem is that the derivation of robust controllers from human demonstration provide observations of fundamental components that can be modulated to address new contexts or act as starting points for learning agents in an exploratory reinforcement learning setting. Obtaining the fundamental components that reconstruct a motion would further enable a functional framework for the abstraction of symbols and knowledge-based rules that contribute to understanding and manipulating synergistic robot motion.

The focus of this chapter is then to establish a background context of major topics behind robot programming by demonstration. Initially, the concepts from human motor control which inform robot imitation are introduced before describing mechanisms of motion sensing technology which drive the data acquisition process for robot teaching. The relevant research in time series modelling, segmentation and classification is reviewed for background on how human motion can be perceived and represented [15]. Finally, the chapter summarises literature in programming by demonstration focusing on methodologies for generating time series from a PbD model.

2.1 Human Perception and Motor Control

The study of human motor control is highly active research area. The contributing factors of motor control commands generally include the motor cortex (primary, premotor and supplementary areas), sensory feedback and central pattern generators (CPG).

The motor cortex function includes planning, control and execution of voluntary movements. A particularly relevant neuronal construct, mirror-neurons, first observed in macaque monkeys [11], are located in the pre-motor cortex. The mirror neuron concept was proposed to explain why a cluster of neurons that activate when executing motor commands for a certain action also activated while the body is stationary but is observing the same motion executed by another body, hence the reference to a mirror. This has counted as evidence for a mechanism for cataloguing and recognizing similar motion that is linked to a neuronal region which activates while generating motor commands to enact this same motion. Mirror neurons are often cited as a prominent factor in human behavioural imitation suggesting a strong link between behaviour observation and generation. Currently there is no conclusive evidence on the manner of their operation in humans yet many research studies have shown a strong correlation between action and observation neuronal centers with functional magnetic resonance imaging (fMRI) [16]. Understanding these links at a neuronal level is far from complete. Nonetheless, the concept has become a source for inspiration in computational imitation learning in robotics [17], particularly in designing data-driven models that are used for both motion recognition and generation.

Sensory feedback systems are the source of perception and provide the motor cortex with environmental information. Without this sensor feedback the motor control system would be operating in an open loop, relying on precise prior information to achieve the motion intention. There are at least three fundamental feedback loops guiding motor control. Proprioception involves the awareness of the relative position of neighbouring body parts and the strain on muscles and joints. Vision is the primary sense for providing awareness of context, apparent self-motion or vection and supplements proprioceptive feedback of the body. The vestibular system provides information on the direction of gravity and other external accelerations through the

otolith, which is thought to contribute to maintaining a stable, upright orientation for the head and upper body [18]. The relative significance of each sensory response under various conditions and how they are ascribed importance in motor control is a common topic in physiological research. It has been shown that the response is adaptive to the extent that with the appropriate interface and training these senses can be substituted [19], a concept referred to as neuroplasticity.

Another source of motion control are central pattern generators (CPG), which are neural networks in the spinal cord producing rhythmic motor commands. They have been shown to be autonomous networks capable of producing muscle activation patterns for locomotion, when stimulated independently to nervous system [20], but normally receive stimulation to modulate the motion. Such feed-forward systems are highly sensitive to disturbances leading to suggestions that humans incorporate internal sensory models resulting in a feedforward/feedback system [21]. [22]

Ultimately neurological research is concerned with understanding the complete perception-action loop. With the research gathered on motor control one can begin proposing control models integrating biomechanical models with feed-forward (CPG), sensor feedback and path planning control. Van der Kooij et al. [23] modelled sensory integration with optimal state estimators using an internal model of body and sensory dynamics. Constrained model variations, such as vestibular loss, produced predictions that corresponded well with experimental results. Such techniques are intended to explain how humans synthesise postural responses to the environment and predict further actions. This provides potential for extension into generating stable motor commands in more complex tasks.

2.2 Motion Sensing Technologies

Motion sensors are a vital component of perception in order interact, co-operate, or teach intelligent robots. In recent years tracking technology has become increasingly miniaturized, and along with computing power, accessible. Measurement error has progressively been reduced with advanced tracking algorithms and post-processing. Several fundamentally different physical principles have been used to measure position and orientation. Mechanisms vary from using a multiplexed reading of orthogonal magnetic fields from inductive coils, accelerometers and gyroscopes, intensity of ultrasonic pulses, the mechanical orientation of joints, or a reconstruction of the position of visible markers detected with multiple cameras. Thus, the efficacy of motion sensing systems varies according to physical susceptibilities, such as occluded trackers, constrained motion or magnetic disturbances.

In this section, we review the available tracking technologies as categorised by the working principle and discuss their advantages.

- *Optical*: Detection with passive optical markers, or reflective indicators, deploys multiple high-speed cameras fixed around the measurement area to triangulate a precise marker position. The use of infrared lighting allows the capture of high contrast images of the reflective markers up to 2kHz. At least two cameras at a time must capture a marker otherwise there are occlusion errors. Markers cannot be differentiated from each other until a post-processing algorithm calculates the correct path. This results in a set of unlabelled points in a three dimensional workspace that correspond to the kinematic structure of the subject which localise the joint centres down to millimetre precision. Redundancy of markers is often used to overcome occlusions, which reduces the probability of errors, but increasing the number of markers also increases the processing

latency.

Active optical markers act as a light source instead of a reflector and are typically deployed as infrared emitting diodes (IRED). The light emitted is multiplexed and therefore the frequency of the camera speed is divided by the number of sensors to detect. Although this introduces a limitation on measurement frequency, less post-processing is required since each individual LED can be identified. The capture area is also limited by the arrangement of cameras and field of view, but is higher than for reflective systems because of the diminishing light intensity over distance.

Ideally motion capture would only use one set of camera(s) from one angle without requiring any body markers. This topic of markerless motion capture is an ongoing research area that relies upon image segmentation and processing techniques to find a human posture which may be matched to a human template [24]. Common approaches employ background scene subtraction techniques to extract a silhouette and various machine learning algorithms [25][26] to classify human postures. Combining video with infrared depth information, in the Kinect system, has resulted in highly cost-effective systems for real-time gesture recognition [27]. Although these systems have enabled a diverse range of applications they are still challenged by unconstrained problem domains where distance limitations and occlusions introduce debilitating artefacts [28].

- *Inertial*: Inertial motion capture relies on acceleration and rotational velocity measurements from triaxial accelerometers and gyroscopes. Each inertial sensor positioned at strategic points on the body measures precise orientation to within 2° RMS [29]. This is achieved with estimation techniques such as Kalman filters [30] fusing the angular rate with the inclination (gravity vector) detected by accelerometers and, for some sensors, magnetometers for more reliable heading

data. Assuming a certain configuration for the sensors and calibrating for actor dimensions an accurate posture can be resolved by applying forward kinematics from the set of sensed orientations. An advantage this system has over optical methods is the flexibility of recording in many environments unrestricted by a field of view of a camera. Although a drawback is that in estimating the base position of the body by double integration of accelerations, a cumulative error arises, referred to as drift. Modern inertial motion capture systems rely upon ground contact force detection, indicated by sudden foot accelerations, to update reference position. Other limitations include the need for post-processing in general environments, when the ground support is varying.

- *Magnetic*: Electromagnetic fields are established through precise current pulses in mounted transmitting antennae. Each magnetic field including the earth magnetic field is measured giving an estimation of joint position, angles and global orientation. AC electromagnetic systems are highly distorted by neighbouring metallic objects but recent DC magnetic field systems exhibit significantly reduced distortion. A tri-axial transmitter produces DC pulses sequentially to each axis and the receiving antennae, mounted on strategic positions on the body, measure the magnetic field along each axis. The earth magnetic field is measured when no pulse is present and subtracted when measuring the orientation. This results in 6DOF position and orientation information for each sensor up to a range of 10 ft from the transmitter [31]. A range of sensor configurations have been employed in robotics and action recognition tasks [32]. Advantages of this approach include the flexibility in locating the sensors on the body, there are no occlusion issues.

- *Mechanical*: The simplest method of capturing pose is to measure orientation directly using electromechanical potentiometers measuring the orientation dis-

placement of each joint. This approach is effective in many cases since it is not affected by external forces or occlusions, measurements can be fast and the equipment portable. Mechanical motion capture system was used for movement imitation in the SARCOS humanoid robot [9]. The main disadvantage is that motion is usually constrained by the rigidity of the wearable equipment and difficult to adjust for different body types. Another problem is in detecting the true position and orientation of the entire frame. This mechanism cannot detect events such as jumping or turning, only the relative angle between limbs.

- *Acoustic*: By attaching ultrasonic transmitters and microphones at specific locations on a moving body an estimate of position can be determined through the intensity of acoustic pulses. The pulses are multiplexed so that each microphone measures the pulse intensity from each transmitter to estimate the relative distances between all sensor points. A hybrid inertial and acoustic sensor suit which could be used in many different environments has been developed [33]. A complication arising from this arrangement is self-occlusion, that is, parts of the moving body blocking a direct path to receiving microphones. It is especially difficult with partial occlusions since the reduced intensity should not be related to distance.

2.3 Segmentation and modelling of human motion

The autonomous organisation and interpretation of a set of trajectories is dependent upon identifying significant transitory events that partition these trajectories into categories. These segments enable further reasoning about the task being demonstrated and to concentrate on modelling the periods of interest.

2.3.1 Temporal Segmentation

Time series segments are time periods within a continuous trajectory for which an event has occurred, as delineated by time stamp markers. This therefore separates meaningful events that are clustered into categories and further modelled to facilitate PbD. The motivation behind time series segmentation is clearly the reduction in manual work of annotating a large database. However long it takes to record the demonstrations, it takes at least as much time to review and annotate, often much longer. In many scenarios, building in a mechanism for a demonstrator to segment the data during the demonstrations is not practical, and when it is feasible it introduces additional distraction for the demonstrator, one that should ideally be automated.

Explicitly recording the time and identity of actions whilst using wearable sensors has been shown to increase the accuracy of classification, however, there is always a trade-off with the interruptive nature of experience based annotation [34]. If possible, it is advantageous to use application specific sensors and domain knowledge to detect these change points. Ward et al. [35] used the relative sound intensity between microphones positioned on the arm to indicate behavioural transitions in a work station. However, in the general case, segmentations must be deduced by directly analysing the motion data itself, which often involves the use of sliding window heuristics or time series models. There are a number of ways to detect changes between adjacent time windows. Many are centered around determining a descriptive model of the data which permits a distance measure between a pair data sequences [36]. Barbic et al. proposed to use the Mahalanobis distance between a Gaussian distribution of a specific time frame and the subsequent sample to detect a significant change. Of the methods tested a distribution based on Probabilistic Principal Component Analysis (PPCA) features achieved the highest precision and recall for detection within a specified time bound. In [37] a dynamic hidden Markov model (HMM) was used to

describe a window of time, and segment points were detected from exceeding a cost factor in describing new data points as unique states. This approach has also been extended to segmentations of human motion capture data [38].

An alternative to finding segmentations locally is to model the entire time series and detect patterns within the predicted state space. Sampling models of hierarchical HMMs can organise the data into proposal classes and, as a consequence, segments [39], however significant training time is required for long high dimensional data sets. Despite this shortcoming the advantage of these hierarchical models is that it provides a probabilistic estimate of the structural parameters such as number and length of segments, thereby quantifying the uncertainty around the model structure [40]. Building the model incrementally may alleviate the burden of computational complexity whilst still clustering the dynamics [41]. Other recent work has investigated clustering the data independent of time and then analysing subsequences of states with dynamic programming. For instance, Zhou et al. [42] used a set of k-means clusters to model the data points and developed a dynamic programming algorithm to cluster related segments and fine-tune their respective lengths. This approach compared favourably in accuracy with the sampled HMM-based methods.

For direct robot demonstrations using kinaesthetic teaching the demonstrator is closely involved with manually positioning the robot and can employ segmentation within the teaching framework. In training a model of the demonstrations, a pre-processing segmentation based on a minimum velocity tolerance is commonly deployed for point-to-point position tracking [43, 44]. Other tele-operation segmentations have focused on zero-velocity crossings to segment the observations [45], however, this is only accurate for trajectories of the appropriate structure and is not accurate in the general case for a multivariate system since many feasible trajectories involved directional changes.

Although the problem of segmentation focuses on detecting change points, it is also important to categorise these segments in order to train generative time series models. Many of the aforementioned unsupervised time series modelling techniques address these objectives jointly.

2.3.2 Action Recognition

In the setting of PbD, if demonstrations are not explicitly referenced and linked to each other an observer will learn combined models of actions which are better applied when compartmentalised. For instance, consider two trajectories, a circular path and a trajectory which crosses over the first and intersecting the circle. If the trajectories are not relating to the same task the reconstructed motion according to the resultant model may follow the dynamics of either set of demonstrations with potentially conflicting objectives.

Human action recognition has had significant attention in the analysis of video streams [46], concentrating particularly on feature extraction. Understanding observations is an important aspect of autonomous systems and a significant amount of research in recent years has been devoted to identifying people and classifying their actions. Although feature extraction is less problematic for recognition using wearable sensors, in previous research the data is seldom modelled due to the high computation cost involved in unsupervised learning. The data is therefore commonly organized in a database structure in a reduced feature space leading to a database query system [47]. The methods pursuing this purpose usually exploit key frame techniques [48] and a nearest neighbor distance metric to return similar frames and predict a class membership.

Rather than rely upon large databases, modelling the data as a continuous process can lead to efficient and parsimonious descriptions of the data. Work by Ward et al.

[35] combined a pair of inertial sensors with microphones to classify actions in a staged workshop setting using HMMs with a manually crafted number of states. Their work also included extensive validation leaving out either repeated recordings or entire data sets from each participant for testing. Expanding upon the sensing capability, Altun et al. [49] used a set of 5 inertial sensors to discriminate between 19 activities. The features were extracted from 5 second windows of raw inertial sensor readings which were further reduced into principal components. A set of 7 discriminative algorithms were evaluated in recognizing the actions, of which, in the leave-one-out test, the Support Vector Machines (SVM) and k-Nearest-Neighbors (k-NN) achieved 88 and 87 percent detection accuracy respectively. Even larger public databases from a limited set of wearable inertial sensors recording daily activities [50] have been classified with a semi-supervised SVM to accuracies of approximately 76 percent [34].

Although these findings demonstrate the practical use of inertial sensors in the recognition, the methodologies and features used typically result in decision thresholds rather than an encoding or representation of the motion. Analysing the motion as composed of a sequence of constituent motion primitives provides an efficient representation with the potential for trajectory reconstruction. Sun et al. [51] recognized several hand gestures by training Gaussian Mixture Models (GMM) of varying sizes and dynamically aligning the sequences. Many studies in robotics focus on motion primitive analysis [43, 10]. A subset of the literature investigates segmentation where, for example, an expanding factorial HMM was used to incrementally organize optical motion capture [41], or for instance in mobile robots [52]. Recognition with motion primitives is, however, less prevalent in the literature [46, 53].

2.4 Reconstruction of expert control skills

Early research into the representation of control skills was motivated by applications such as improving the intuitive and efficient programming of industrial robots. Initially, demonstrated trajectories could be followed by extracting key points or way points for the end effector, allowing a demonstrator to show a particular path and the robot to follow by targeting and interpolating between the key points [54]. This resulted in a brittle control scheme where a robot could replicate motions but would fail in a different context without additional design work.

By observing the response of the system under human control one can train state-action rules in the form of a decision tree, which characterise the observed control inputs within regions of the system state space. These ideas have been applied to flight simulator control [6], crane operation [13] and pole balancing problems [55]. Although at the core this methodology a situation-action rule base also results in an inflexible controller research has targeted the learning of goal-directed rules, sub-goals and qualitative descriptions of system dynamics. Bain et al. [56] used a dual layer framework that learned consequences of control actions and a model of how goals directed the control. Isaac et al. [4] rather approximated flight simulation operator's reactions to differences between the actual and anticipated state of the system. This focused on learning goal settings and flight manoeuvres guided by decision trees and PID controllers. Shiraz et al. [57] attempted to supplement a model automatically using ripple-down rules (RDR). Pilots therefore were able to interactively build a rule base that could control subsequent flight paths.

The advantage of describing the learned control methodology in a decision tree structure is that it is amenable to interpretation and can express the intentions of the pilot as a function the control inputs in a symbolic language.

2.5 Robot programming by demonstration

In the last decade there has been an increased drive to research the ideas of learning by demonstration [58]. The early methods of encoding motion employed the idea of way-point extraction and interpolation. Identifying segmentation points form a key component in partitioning the control path into smaller manageable sub-tasks [59]. More recent work on this approach uses statistical methods such as discrete HMM, to cluster and align the key points from all demonstrations in order to reproduce a generalised trajectory [60]. However, the flexibility of the trajectories generated by these models is limited by the extraction of the key points making it difficult to adapt the trajectory in compensation for unforeseen circumstances in a real-time control setting.

Statistical approaches for encoding motion have a distinct advantage due to the capacity to quantify spatial and temporal uncertainty along the observed trajectories. Inamura et al. [10][61] represented demonstrations with continuous HMMs by assigning the observation distribution with a Gaussian density. [41] expanded upon this framework by incrementally updating the model and creating a hierarchy of HMM sequences using Factorial HMM (FHMM). While this model is well suited to the recognition of actions the reproduction approach entailed averaging over extensive stochastic samples from the model to attain a smooth trajectory. To remedy this deficiency of extensive sampling, Gaussian mixture regression (GMR) was proposed to form a regression of the state given an explicit time index, which efficiently reconstructs the motion, albeit without a HMM due to the direct inclusion of temporal information [43]. Related studies have employed a Kalman smoother and dynamic programming to temporally align the demonstrations to form a generalised trajectory and learn the local dynamics of the system with applications in helicopter aerobatics [7] and surgical robots [62]. Introducing an explicit time dependency, however,

increases the system sensitivity to disturbances.

Another approach entirely is to posit an underlying autonomous dynamical system of which the demonstrations are but manifestations starting from a particular initial condition. This idea was advocated in [9] and has lead to the technique termed dynamic movement primitives, which form a stable system per dimension by smoothly switching from nonlinear to a linear dynamical system (LDS) to terminate the motion. In [63] it was shown that the GMR formulation can be extended to describe a multidimensional nonlinear dynamical system as a weighted sum of a set of LDS. Several extensions to this initial model have been proposed in recent years. Zadeh et al. [44] proposed an optimization scheme to tune the system for global asymptotic stability for any point-to-point motion. Additional modification included constraining the system to reach the target at a particular velocity [64] and modulating the velocity in the immediate presence of obstacles [65]. Other authors have also proposed a Dirichlet process model to learn the number of states required by the system [66], whilst in [67] the Dirichlet process sampling was constrained to ensure global stability according to the conditions in [44]. To deal with trajectories which cross-over a HMM was proposed to calculate the probability of the current state [68], as opposed to the GMM posterior. In [68], a mass-spring damper control approach was introduced to stabilise the dynamic system which shares similarities with the method presented in this thesis.

Research has intended to advance from following trajectories to learning and generalisation of manipulation skills, to imitate humans in a flexible manner [5]. Part of the challenge in achieving this goal is to not only extend these techniques to higher degree of freedom cases but to account for structural and environmental challenges such as body stability for bipedal robots.

2.6 Robot constraint and correspondence mapping

In the translation of motion between kinematic structures there are usually discrepancies due to the available degrees of freedom, the distance and alignment between joints and the inertial properties of the body. There are also constraints due to the limits to the range of motion, constraints of the environmental obstacles or self-collision, and balance stability constraints on under-actuated robots.

In any imitation learning setting one requires a function which maps the set of coordinates describing the current pose of the demonstrators body to a corresponding set of coordinates on the target robot. There are a number of approaches to deriving this function dependent on the requirements of the application. For instance, for an anthropomorphic robot, that is, a robot with equivalent DOF and body size to a human, the set of joint angles may be found by a direct match of joint angles or joint position coordinates. Otherwise, matching the demonstrated motion to robot of different kinematic proportions would be resolved with an inverse kinematics function and may result in different postures to achieve the desired positions. For large differences in proportions the positions may be scaled to match the workspace of the robot. If the programming by demonstration can occur in a real-time teleoperation setting then the demonstrator can compensate for the correspondence by adapting their postures to assist the robot in achieving the goal. Of course, these demonstrations are then specific to the robot being driven and may not translate well to other robots.

Human motion is often captured in models with over 50 degrees of freedom (DOF) but, depending on the robot, the target frame has significantly less DOF. With the intention of unifying the correspondence between a human and robot, Azad et al. [69] proposed a Master Motor Map (MMM) framework to generalise this reference mapping problem. From this format a method was presented for converting to different

Euler angle conventions and reduced DOF models.

The synthesis of trajectories for robot control need to be constrained by limitations of the robot structure or the environment in which it operates. This is a necessary consideration for a reliable PbD system, particularly when the correspondence between the demonstrator and robot structure is not strong. Robots with redundant DOF synthesis can be constrained by exploiting the nullspace of constraint Jacobians which are designed to avoid joint limits and self-collision trajectories [70] or with common motion planners such as rapidly-exploring random trees (RRT) that can find feasible paths through the robot workspace [71]. With dynamic trajectory synthesis environmental obstacles can be posed as repellent regions to modulate the trajectory smoothly in real-time in response to the object [65].

Additional structural stability constraints need to be considered in a robot with unactuated 6-DOF at the base, such as a biped. Any underactuated robot needs to satisfy either static or dynamic stability conditions to avoid collapse. Static stability can be controlled by positioning the center of mass within the convex hull of the ground contact plane, as spanned by the contact forces. Whereas in locomotion, dynamic stability relies on positioning the zero moment point (ZMP) within the ground contact plane [72]. Some solutions to this have proposed optimising the mechanical design [73], referred to as passive dynamics, or by controlling the ZMP through the Jacobian of the center of mass [74]. This has been applied to real-time humanoid robot imitation with nullspace projection constraint satisfaction [75]. ZMP calculations have also been extended to account for disturbance and planning on irregular terrain [76] and iterative planners have optimised locomotion trajectories based on stability and minimal energy consumption criteria in simulation [77].

The remaining challenge is to capture the important characteristics of demonstrations in a model which is conducive to synthesising stable trajectories that balance

the similarity with the demonstrations with the satisfaction of external constraints.

2.6.1 Humanoid control and imitation

The control of humanoid robots has centered around solving the issues of satisfying constraints or environment, stability and correspondence. In [78] a humanoid robot imitated a dance routine, captured with passive optical markers, while selecting stable postures from a repository of motion primitives. Other research has focused on maintaining static stability while performing upper body actions. For instance, a humanoid robot was kinaesthetically demonstrated writing on a white-board and remained stable by incorporating a inverted pendulum model which was offset by the force applied to the board [79]. In [80] a robot maintained static stability in an attempt to hit a baseball according to demonstrations. In the use of dynamic motion primitives as a collection of LDS [14], a group of robots were synchronised to reconstruct demonstrations by mapping the state of a series of oscillators per joint to the demonstration traces using support vector regression [81]. Using the same procedure crowds of simulated characters could be dynamically synchronised from different initial conditions.

2.7 Manifold imitation learning

Extending the robot programming by demonstration approaches directly to high DOF systems typically encounters scalability issues such as singular covariance matrices. This is often addressed by dimensionality reduction techniques which seek to transform the data coordinates into a low dimensional space that still captures the data variance. Indeed, if the new space does not model the data well, there will be overlapping regions in the low dimensions due to the many-to-one mapping of dimensionality reduction. The simplest approach is to perform a linear transformation such as prin-

cipal component analysis (PCA) based on the eigenvectors of the data covariance. In [82], PCA was used to find a linear combination of the features as a preprocessing stage to reduce noise before learning a GMM. Linearity across the entire data set is, however, a restrictive assumption to be enforcing, instead many approaches exploit local linearity. It allows greater flexibility in the model but there are multiple ways to implement a model with this assumption. In the context of motion primitives and non-linear dynamical systems, if the GMM components are locally coordinated they produce a locally linear dimension reduction model [83]. Recent work in computer vision has demonstrated the effectiveness of these models in tracking human postures in video [25]. Similarly manifold mapping approaches have been applied succesfully to robot control problems that incorporate additional constraints [84].

Another dimensionality reduction scheme relies upon the traversal of a local neighbourhood graph that is reduced by multi-dimensional scaling. This algorithm, ISOMAP [85], was extended by Jenkins et al. to account for both spatial and temporal aspects of the data [86]. The temporal context of motion was considered in the construction of the neighbourhood graph and the resulting trajectories were replayed in an upper body humanoid.

In [87], a nonlinear dimensionality reduction was presented for robot correspondence mapping based on the Gaussian Process Latent Variable Model (GPLVM). A low dimensional correspondence space was formed between human motion capture and the posture of the robot. Samples from this latent space generate paired postures for the robot and human. This dimension reduction method has been further extended to modelling dynamics for human motion capture in Gaussian Process Dynamic Models (GPDM) [88].

2.8 Chapter Summary

In this chapter a review of the research problems related to robot programming by demonstration is presented with reference to past and current research. The first topic is on the study of understanding human motion and how biologically inspired algorithms can assist in controlling robots. The focus is then shifted to the sensor systems available for measuring motion and their advantages and disadvantages. The next section covered the segmentation and recognition of motion as recorded by demonstrators with a sensor system, while the following section described programming frameworks which utilise the segmented data of the demonstrator. The remainder sections discuss the practical challenges posed in deploying the PbD techniques in any robots and how modern approaches employ dimension reduction to exploit local structure in the data to overcome the degeneracies in model structures.

Chapter 3

Detection of detrimental posture adjustments from motion primitive analysis

The variability in demonstrations of motion skills is typically in response to environmental constraints, sensor error, or circumstantial goal changes. In the context of human demonstrations, changes in the execution of an action can further correspond to fatigue, rehabilitation, indicate shifting intentional goals or to skill refinement.

3.1 Introduction

This chapter introduces the foundations of motion primitive skill modules and provides an analysis of the sensitivity in detecting subtle changes in observed actions. From a robot demonstration learning perspective, one would typically assume for simplicity, that fatigue is not a factor in observing human demonstrations. However, detecting subtle posture adjustments is of central importance for interactive robot-assisted rehabilitation and supportive exoskeletal robots. If capable of detecting the

effects of fatigue the robot may be programmed to provide appropriate assistance without over-reacting to the natural variability in human movement [89]. Models capable of detecting and describing variation in exercises are also valuable for forming hypotheses for further targeted biomechanical studies and in designing sensors systems that provide clinical decision support.

An investigation into time-series segmentation and recognition on movement data should therefore consider the variance within action categories. Not only to gauge the sensitivity with which to classify a series of demonstrations but also to investigate other underlying factors influencing the variation in demonstrations. Kinematic indicators of fatigue or indeed strengthening, in a rehabilitation context, may play a vital role in longer-term human-robot interaction.

The concepts and equipment involved in the motion perception model are introduced before describing two experimental studies on the topics of tracking injury rehabilitation and posture evaluation in functional capacity assessment. In each case, an evaluation is required in terms of the quality of the observed motion against a standard. For injury rehabilitation, this standard is provided by the symmetry in motion between actions performed with and without recruiting the injured muscle or tendon. Improvement is expected over repetitions. For functional capacity assessments the standard is provided by historical observations of the exercise performed by both the patient and a cohort of control subjects and deviation from the standard is expected over repetitions due to the onset of fatigue.

3.2 Motion Primitive Models

The motion primitive models considered in this thesis are composed of a mixture of multivariate Gaussian distributions. Each component of the mixture model describes the variance in a particular region of the observational workspace and is often referred

to as a 'primitive' mode [14][10]. This is due to the fact that one component by itself can describe the simplest of motions as observed in a linear dynamical system, however, an arrangement or sequence of 'motion primitive' modes can describe an arbitrary trajectory.

3.2.1 Mixture Models

A Gaussian mixture model (GMM) is defined by a probabilistic combination of multivariate Gaussian distributions. A set of N data samples, $\mathbf{x} \in \mathbb{R}^{N \times d}$, of dimensionality d is assumed to be drawn independently and identically from one of a set of K multivariate Gaussian distributions indicated by the latent membership binary variable $\bar{s} \in \{0,1\}^{N \times K}$. The latent binary variable $\bar{s}_{nk}$ indicates whether the n^{th} data point belongs to the k^{th} Gaussian distribution and $s_n \in \{1, \ldots, K\}$ is the corresponding integer index. The joint probability $p(\mathbf{x}, s)$ is factored as

$$\begin{aligned} p(\mathbf{x}, s) &= p(s)p(\mathbf{x}|s) \\ p(\mathbf{x}|s) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s) \\ p(s) &= \pi_s \\ p(\mathbf{x}) &= \sum_{s=1}^{K} \pi_s \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s), \end{aligned} \tag{3.1}$$

where $\mathcal{N}(\mathrm{x}|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ is a multivariate Gaussian component with mean $\boldsymbol{\mu}_s \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_s \in \mathbb{R}^{d \times d}$, which is defined as

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \Big). \tag{3.2}$$

The component responsibility for a data point $\mathbf{x}_n$ is

$$p(s|\mathbf{x}_n) = \frac{p(s)p(\mathbf{x}_n|s)}{\sum_j^K p(j)p(\mathbf{x}_n|j)} = \frac{\pi_s \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{3.3}$$

and will also be denoted by $\gamma(\mathbf{x}_n)$.

Maximising the corresponding log-likelihood function,

$$\mathcal{L} = \sum_{n=1}^{N} \log p(\mathbf{x}_n), \tag{3.4}$$

increases the explanatory power of the model. Due to the form of the function as a logarithm of a mixture, it is maximised iteratively via the Expectation Maximisation (EM) algorithm [90]. A useful interpretation of this algorithm is that it places a lower bound over the log-likelihood function using Jensen's inequality to produce a tractable function to maximise. Introducing a distribution over the mixture, q, the bounded log-likelihood becomes

$$\begin{aligned} \log p(\mathbf{x}_n) &\geq \log p(\mathbf{x}_n) - \sum_{s=1}^{K} q(s|\mathbf{x}_n) \log \frac{q(s|\mathbf{x}_n)}{p(s|\mathbf{x}_n)} \\ &= \sum_{s=1}^{K} -q(s|\mathbf{x}_n) \log q(s|\mathbf{x}_n) + q(s|\mathbf{x}_n) \log(p(s|\mathbf{x}_n)p(\mathbf{x}_n)) \\ &= \mathcal{H}(q(s|\mathbf{x}_n)) + \sum_{s=1}^{K} q(s|\mathbf{x}_n) \log p(\mathbf{x}_n, s) \end{aligned} \tag{3.5}$$

where $\mathcal{H}(q(\mathbf{x})) = -\int q(\mathbf{x}) \log q(\mathbf{x}) d\mathbf{x}$ is the entropy of the distribution q.

To maximise the bounded log-likelihood one takes the derivative of the function with respect to each of the parameters $\boldsymbol{\mu}_s$, $\boldsymbol{\Sigma}_s$ to obtain the updates

$$\boldsymbol{\mu}_s = \sum_{n=1}^{N} \tilde{q}_{ns} \mathbf{x}_n, \qquad \boldsymbol{\Sigma}_s = \sum_{n=1}^{N} \tilde{q}_{ns} (\mathbf{x}_n - \boldsymbol{\mu}_s)(\mathbf{x}_n - \boldsymbol{\mu}_s)^T \tag{3.6}$$

where $\tilde{q}_{ns} = \frac{q_{ns}}{\sum_{n=1}^{N} \tilde{q}_{ns}}$ and $q_n = \gamma(\mathbf{x}_n)$.

The EM algorithm alternates between assigning responsibilities to the data points per component based on the current parameter estimate (E-step) and updating the parameters (M-step) with 3.6. This algorithm converges to a local maximum of the likelihood function.

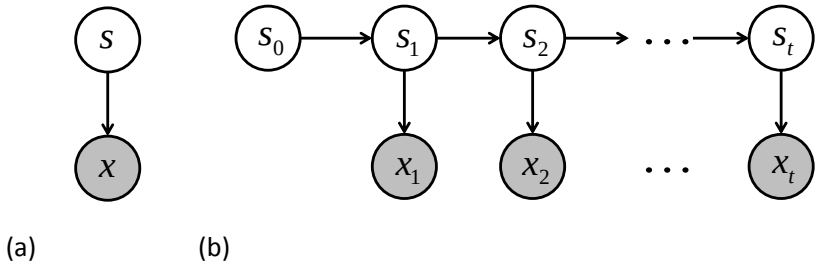

Figure 3.1: (a) Gaussian mixture model. (b) Hidden Markov model.

In contrast to the GMM, a natural extension to the mixture model in Figure 3.1a is to allow a time dependency between the latent variables, which gives rise to a hidden Markov model (HMM), as in 3.1b. The predicted state is then dependent on the sequence of previous state assignments up to time, t.

In a HMM, each hidden state, s, emits an observation drawn from a set of probability density functions, which is assumed here to be of multivariate Gaussian distribution form. The difference is that a transition probability matrix, $\mathcal{T}$, influences the probability of a occupying a particular state, where $\mathcal{T}_{ij} \equiv p(j_t|i_{(t-1)})$ is the probability of the state i at time $t-1$ transitioning to the state j at the next time instant, t. The forward-backward algorithm [91] computes the sufficient statistics of $\mathcal{T}$. The forward probability or filtering step, α computes the component probabilities for the current data point in the sequence as a function of the inferred state of all prior state variables,

$$\alpha_{s_t} = p(\mathbf{x}_t|s_t) \sum_{s_{(t-1)}} \alpha(s_{(t-1)}) p(s_t|s_{(t-1)}), \tag{3.7}$$

whereas the backward probability or smoothing step, β, accounts for inference over future observations.

The number of components in the mixture model is often referred to as the model complexity. In many applications, K is decided in advance, however, one can use model selection or averaging methods to estimate an appropriate model size. A simple method, derived from the Laplace approximation, is given by the Bayesian Information Criterion (BIC) [92],

$$\mathcal{L} \approx \hat{\mathcal{L}} - \frac{1}{2} K_p (\log N - \log 2\pi) \tag{3.8}$$

where K_p is the number of parameters in the model and $\hat{\mathcal{L}}$ is the likelihood with maximum a posteriori parameters. This approximation is more accurate for a large N. An alternative complexity penalty is given by the minimum message length (MML) criterion, which is based on the minimal coding length of combined model and data representations [93]. Apart from evaluating the measure for every K up to pre-defined maximum, which may prove time-consuming, iterative search algorithms have been proposed that employ various heuristics in order to split the clusters or re-merge based on the corresponding reduction in message length. The number of components, K, is therefore increased until there is negligible reduction in message length. Some examples of where this model selection algorithm has been used is in classifying rock samples [94], and power system events in harmonic data [95]. Another alternative model selection approach arises from an extension of the variational bound shown in Equation 3.5 to a model with explicit prior distributions [96]. Furthermore, a non-parametric prior can be placed over the number of mixture components in the form

of a Dirichlet process [97].

3.2.2 Dynamic Time Warping

The similarity between two data sequences can be efficiently evaluated with a local time scaling known as dynamic time warping (DTW). The algorithm finds the warping path or local time scaling which minimises the distance between the two signals according to a given distance measure. Given two sequences $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_x}]$ and $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N_y}]$ of length N_x and N_y, one computes a similarity matrix, $\mathcal{K}$, based on a choice of kernel function. Typical functions are Euclidean distance or a radial basis function. A known limitation of DTW is that it does not satisfy the triangular inequality, however, a variant of DTW known as dynamic time alignment kernel (DTAK) can satisfy this property [98]. A cumulative cost matrix, U, is generated by

$$u_{ij} = \min \begin{cases} u_{i-1,j} + \mathcal{K}_{ij} \\ u_{i-1,j} + 2\mathcal{K}_{ij}, \;\; u_{11} = 2\mathcal{K}_{11} \\ u_{i,j-1} + \mathcal{K}_{ij} \end{cases}$$

and the minimum cost of aligning the two sequences is given by

$$\tau(\mathbf{x}, \mathbf{y}) = \frac{u_{mn}}{N_x + N_y} \tag{3.9}$$

where the indices m and n are

$$(m, n) = \begin{cases} (N_x, \underset{m}{\operatorname{argmin}}\, u_{N_x m}), \text{if } \min(u_{:,N_y}) < \min(u_{N_x,:}) \\ (\underset{m}{\operatorname{argmin}}\, u_{m N_y}, N_y), \text{if } \min(u_{N_x,:}) \geq \min(u_{:,N_y}) \end{cases}$$

For instance, two sequences and the corresponding sample similarity are shown in Figure 3.2. By locally aligning the sequences it is clear that the two example sequences

share a high similarity, whereas a direct sample-by-sample Euclidean distance would result in a higher dissimilarity than is warranted. Trivial alignments with an extensive warping can be eliminated by bounding the local distance metric. A common bound is the Sakoe-Chiba band [99], which limits the extent of warping by a proportion of the sequence lengths, $b_{sc} \in [0, 1]$. It is applied by assigning $\mathcal{K}_{ij} = \infty$ for $j < i - \lceil b_{sc} N_x \rceil$ or $i > j + \lceil b_{sc} N_y \rceil$. [42]

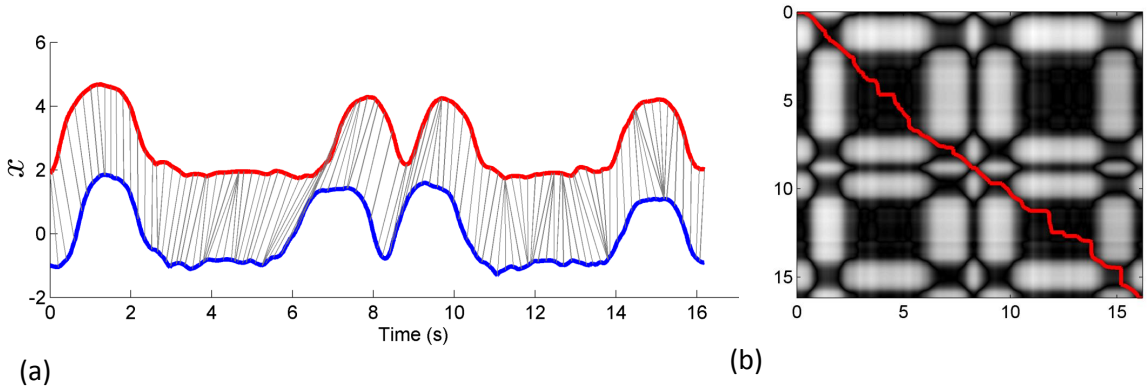

Figure 3.2: (a) The DTW alignment of two unidimensional signals, in red and blue, where the alignment of data points is shown in gray. (b) The sample kernel matrix $\mathcal{K}$ and the resulting alignment maximising the similarity between the two sequences.

3.2.3 Classification

The classification of motion according to a motion primitive model can be attained from the predictive machinery of mixture models. Given a training set of example classes of motions for which a model, $\mathcal{M}_i$, is trained, one computes the log-likelihood of the data conditioned on each particular model, $p(x|\mathcal{M}_i)$. The class with the maximum likelihood is assigned to the corresponding data point.

There are several forms of discriminative algorithms to compare with mixture models. Two prevalent algorithms are nearest neighbour classification and the support vector machine (SVM). The nearest neighbour method (k-NN) assigns a test

sample to the majority class of the nearest k data points (in Euclidean distance) in the training set. Although this method can often achieve a competitive accuracy it does not compress or summarise the data. The SVM is typically posed in an optimisation problem which seeks a maximal margin hyperplane, $\mathbf{w}$, whilst minimising erroneous classifications. A linear SVM may be described by,

$$\begin{aligned} &\text{minimise} && \|\mathbf{w}\|^2 + C\sum_{i}^{N}\mathcal{E}_i, \\ &\text{subject to} && y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 1 - \mathcal{E}_i, \forall i = 1 \ldots N, \end{aligned} \tag{3.10}$$

where $y_i \in \{-1, 1\}$ are the class labels, $\mathcal{E}_i$ are the classification errors, b is the bias term of the linear system, and C a regularisation parameter weighing the influence of misclassifications. If the classes a not linearly separable, within acceptable error weighted by C, the data may be mapped with a non-linear function into a high dimensional feature space in which the classes can be linearly separated.

The SVM was tested with the multiclass function of LIBSVM toolbox for MATLAB [100]. A radial basis function kernel was used which is parameterised by γ, in the form $\mathcal{K}(\mathrm{x}_m, \mathrm{x}_n) = \exp(-\gamma|\mathrm{x}_m - \mathrm{x}_n|^2)$. A logarithmic grid search was carried out in an attempt to find an optimal setting. The input attributes must be normalised to range between [-1, 1].

Model Validation

The extent to which a model represents the observed data is crucially balanced with predictive performance. Model validation is the assessment of the extent to which the model has overfitted the parameters to the training data and is commonly used to tune free parameters in the training algorithm. Bias in selecting the training data is reduced by organising an iterative cross-validation procedure. Many studies use n-

fold cross-validation where the data is randomly partitioned into n segments and one segment is held from the modelling step and tested against the model formed from $n-1$ segments. This process is repeated n times by rotating the data segments. The n segments may be chosen at random or based on the structure of the intended classifier application. The validation scheme generally used in this thesis is to partition the data based on distinct recording sessions or by the individual from whom the data was recorded, often referred to as a leave-one-out (L1O) validation. In this way, each of the validation iterations provide a realistic scenario of obtaining a new, unseen set of observations, assuming the order in which the data sets are recorded is of no significance.

Prediction Performance

The predictive performance of a classifier can be measured in a number of ways. In the simple case of a binary classification, there are only 4 possible outcomes as shown in Table 3.1. It is known as the confusion matrix or contingency table and will be denoted as $\mathcal{C}$. Each prediction belongs to one of four categories, True Positive (TP), False Positive (FP), False Negative (FN), or True Negative (TN). To calculate the accuracy of the classification the total number of correct predictions, on the diagonal of the matrix, is found as a proportion of the all predictions given by,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}. \tag{3.11}$$

Table 3.1: Contingency table.

		True class	
		c_1	c_2
Predicted class	c_1	TP	FP
	c_2	FN	TN

By itself the accuracy can portray a misleading assessment when there is an im-

balance between the number of positive and negative examples of the class. To assess the detection rate or **recall** of the classifier as,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.12}$$

provides measure of the sensitivity of the classifier to declaring a positive prediction. The **precision** measure rather, assesses how precisely positive predictions are made by considering the significance of false positives,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3.13}$$

In order to concisely report levels of precision and recall, a harmonic mean, or F-score, is often computed as the weighted average of the two measures,

$$\text{F}_1 = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.14}$$

Moreover, it is common to assess the sensitivity of a binary classifier to detection thresholds with a Receiver Operating Characteristic (ROC) curve, which is the rate of true positives versus rate of false positives when varying a detection confidence threshold.

Extending the prediction to a problem with c target classes, the confusion matrix expands to be a $c \times c$ matrix where the diagonal entries are the true positive classifications and off-diagonal entries are referred to as substitutions. The accuracy of classification may then be computed with the following expression[1],

$$\text{Accuracy} = \frac{\mathbf{tr}(\mathcal{C})}{\mathbf{1}^T \mathcal{C} \mathbf{1}}. \tag{3.15}$$

[1] where $\mathbf{1}$ refers to the column vector of ones, $\mathbf{1} = [1 \dots 1]^T$, and $\mathbf{tr}(\mathbf{A})$ refers to the trace of the matrix $\mathbf{A}$.

3.3 Motion Sensing Setup

In this section, we introduce the sensors used for the tracking demonstrations in the thesis and the kinematic model information, which involves the notation for describing the motion of the body and the associated mechanical properties.

3.3.1 Sensors

The sensors used in this thesis are inertial measurement units (IMU) consisting of tri-axial accelerometers (±5g), gyroscopes and magnetometers. Each sensing unit fuses the three tri-axial measurements to estimate the 3D orientation using an Extended Kalman Filter (EKF) [101]. To summarise the working principle, a gyroscope measures angular velocity around each axis. Integrating this signal retrieves the orientation around each axis, however, unavoidable sensor noise leads a cumulative error in orientation, known as integration drift. The EKF estimates the drift bias by comparing the orientation with estimates of the incline from accelerometers, sampling the gravitational direction, and heading from the magnetometers, sampling the deviation from the earth's magnetic field. The resulting estimation of the 3D orientation is reported to be accurate to within 2° providing an accurate measurement of the global orientation of a particular body frame. Figure 3.3a depicts the particular IMU used, the MTx available from Xsens technologies [1].

3.3.2 Motion capture system

Data was gathered with an inertial motion capture system MVN [1] comprising of 17 MTx sensors, which are secured to the body with a series of straps. These sensors are located along each body segment between pairs of joints and connected in a chain to 2 transmitters positioned on the back of the subject, as illustrated in Figure 3.3b.

Figure 3.3: (a) A single IMU, the MTx, from Xsens technologies [1] (b) Location of many of the 17 inertial sensors and 2 transmitters on body.

For each subject, the kinematic model is initialised with a set of standardised body measurements which allows the orientations measured for each limb to be resolved in a 23 segmental kinematic structure described by joint positions and angles. The output data provides representations in a global reference frame (defined by an initial calibration point) for joint positions, linear and angular velocities, joint angles and accelerations at a sampling frequency of 120Hz.

3.3.3 Kinematic Model

There are several sets of features or representations which may adequately describe the variation in motion between the actions, including 3D joint angles, θ, angular velocities, ω, relative joint positions, p, linear velocities, $\dot{p}$, or accelerations, $\ddot{p}$. In this thesis, the features predominantly used were the set of joint angles with the exception of the pelvis yaw angle, the linear velocity of each joint rotated relative to the pelvis sensor frame and the relative joint positions. The pelvis yaw angle is typically independent of any action since in practice it is merely the compass direction. In Figure 3.4 we display the joint positions, midpoint and center of mass (COM) positions of the 23-segment body. The position vector ${}_1x_{14}$ refers to the vector

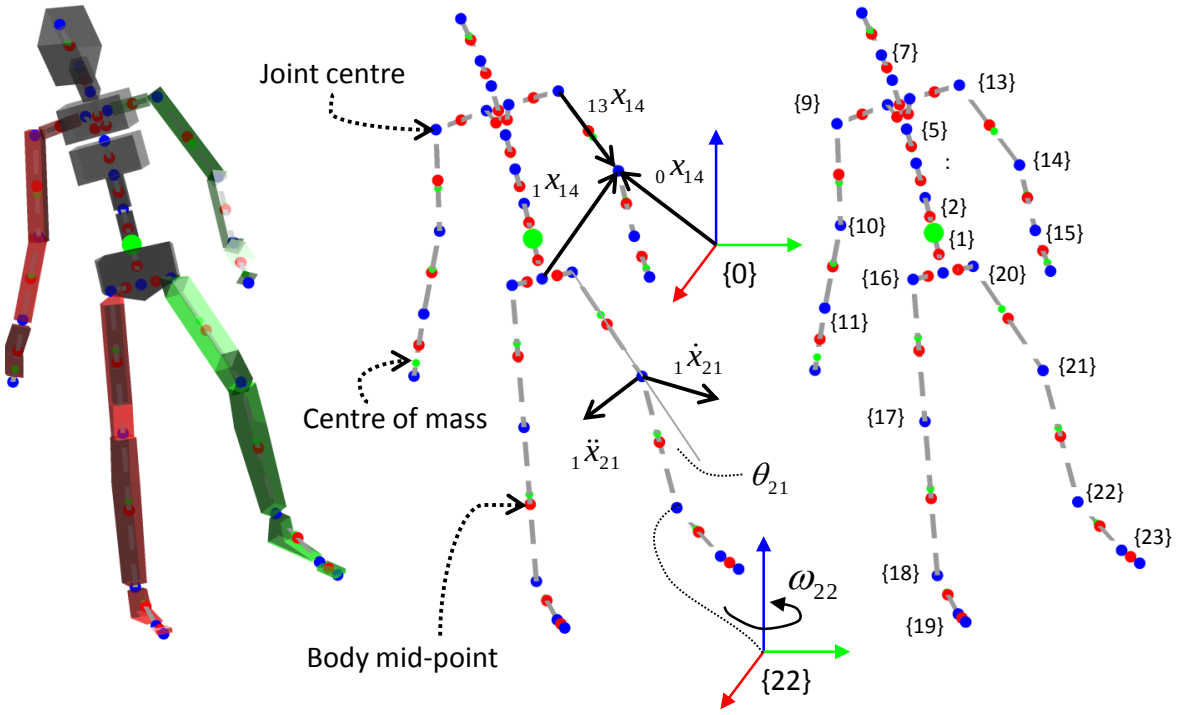

Figure 3.4: Joint angle and position data representations according to a 23 segment biomechanical model consistent with the MVN motion capture system [1].

pointing from the origin of frame {1} to the origin of frame {14} which happens the left elbow. The center of mass locations and proportions are informed by anthropometric literature [3] and are listed in Table 3.2.

The COM position of the whole body,

$$\mathbf{x}_{com} = \frac{\sum_i m_i \mathbf{x}_i}{M}, \tag{3.16}$$

where $\mathbf{x}_i$ and m_i are the position in the global frame and mass of the i^{th} body segment respectively and M is the total mass of the body ($M = \sum_i m_i$), in relation to ground contact is crucial to the stability of the body. A body is statically stable if the COM resides within the ground contact plane. This ground support polygon, bounded by the convex hull of the ground contact forces, as shown in Figure 3.5, is enclosed by a

Table 3.2: Center of mass proportions [3].

	Male		Female	
Body Segment	CoM position (%)	Mass(%)	CoM position (%)	Mass(%)
Head {7}	59.8	6.9	59.8	6.9
Thigh {16,20}	41.0	14.2	41.0	43
Shank {17,21}	44.6	4.3	44.2	4.8
Foot {18,22}	44.2	1.4	40.1	1.3
Upper Arm {9,13}	57.7	2.7	57.5	2.6
Forearm {10,14}	45.7	1.6	45.6	1.4
Hand {11,15}	79.0	0.6	74.7	0.6
Trunk {5}	44.9	43.5	41.5	42.6

set of normal vectors $\mathbf{a}_i$. If concatenated as,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_j \end{bmatrix}^T, \tag{3.17}$$

the region is expressed as the set of points, $\{\mathbf{x}|\mathbf{A}\mathbf{x} \preceq b\}$.

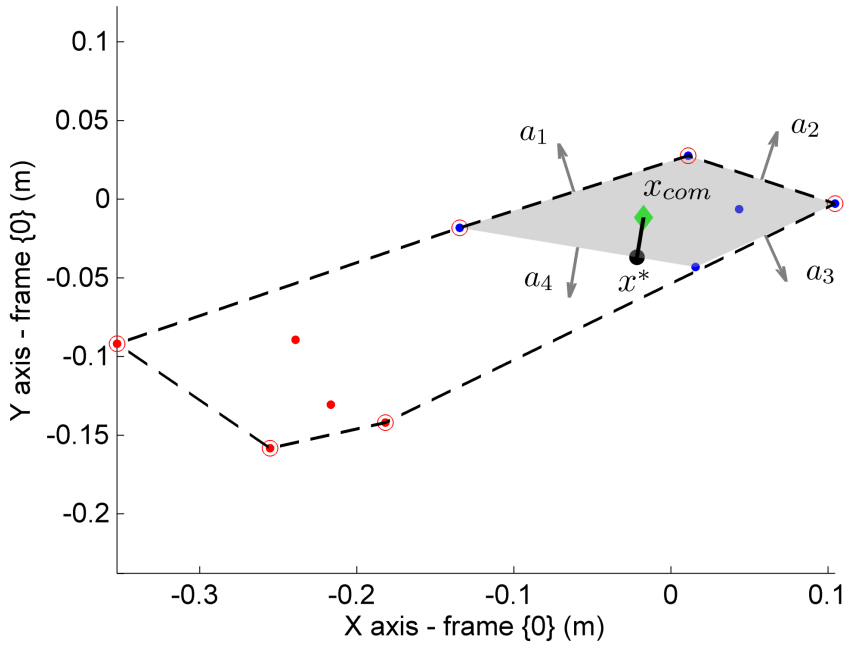

Figure 3.5: Illustration of the centre of mass tipping distance (black line) as the distance from the centre of mass, $\mathbf{x}_{com}$ (green), to the edge of the support polygon (black).

If the body is statically stable when the centre of mass, $\mathbf{x}_{com}$, is located within

this region, the minimum distance between $\mathbf{x}_{com}$ and an edge of the ground support, $\mathbf{x}^*$, is indicative of risk of instability. This can be expressed as a linear program,

$$\begin{aligned} \text{minimise} \quad & \|\mathbf{x}_{com} - \mathbf{x}\|, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \preceq b, \\ & \mathbf{a}_i\mathbf{x} = b_i, \end{aligned} \tag{3.18}$$

where the second constraint requires equality for at least one of j polygon boundaries, $i \in \{1 \dots j\}$, and b is a set of j linear bias terms. The resultant distance is denoted $d = \|\mathbf{x}_{com} - \mathbf{x}^*\|$ and when $\mathbf{x}_{com}$ resides outside the polygon the sign is reversed.

Since the position and orientation of the foot in relation to the body is measured, a set of points (fixed in the foot body frame) encompassing the surface area of the foot define the boundary of the ground support area. In the absence of pressure sensors to detect the ground contact, the kinematic model uses foot accelerations to detect foot falls. A threshold on the height of the foot points determines which subset of points are in contact with the ground in the event of a foot being raised.

The condition that d remains positive is insufficient for dynamic stability. It has been shown that dynamic stability can be ensured with the centre of pressure (COP) or the zero moment point (ZMP). The concept of zero moment point (ZMP), introduced by Vukobratovic et al. [72], is a projection point on the ground contact plane for which the total body moment is zero. It coincides with the center of pressure (COP) under certain conditions [102].

3.4 Injury Rehabilitation Monitoring: Case Study

An expanding application of motion sensors is to provide feedback on a rehabilitative exercise for sufferers of a stroke or a significant injury. As a first case study we

analyse the progressive rehabilitation of an injured athlete and compare approaches for highlighting improvements in condition using motion sensors. After an injury is incurred the performance of particular exercises can reveal the limitations imposed by the injury or the level of discomfort experienced by the subject [103][104]. Incremental changes are expected as the recovery period proceeds [105]. In this section, the sensitivity of the sensors and efficacy of quantifying the variance of human motion is investigated by recording rehabilitation exercises from a recovering athlete.

3.4.1 Experimental Setup

A group of 14 subjects (10 male and 4 female) participated in the study, four of whom were athletes. Two of the athletes were nursing an injury, one of which was monitored with a series of exercises every 4 weeks during a 12 week period of rehabilitation and training after surgery on the right ankle, and the other was observed once at the beginning of the rehabilitation. The central exercise recorded for all participants was a single leg balance test for 20 second periods alternating between each leg and between keeping their eyes open or closed. This exercise was repeated up to 3 times per recording session. Additionally, the recovering subject, recorded over multiple weeks, also recorded a session of walking for 40 metres with eyes closed. This motion was omitted from the schedule of the first session due to discomfort experienced by the subject.

In order to assess the significance of the change in motion across repeated rehabilitation sessions two of the uninjured subjects were recorded performing the balance test in three sessions spaced over a similar time period. This estimates the typical variation between recordings among control subjects.

3.4.2 Results

The data was examined from three perspectives. The symmetry of the data variance in successive trials, the rate of predicted state changes in GMM representing the data, and the center of mass trajectory while walking.

Probability Density - Variance

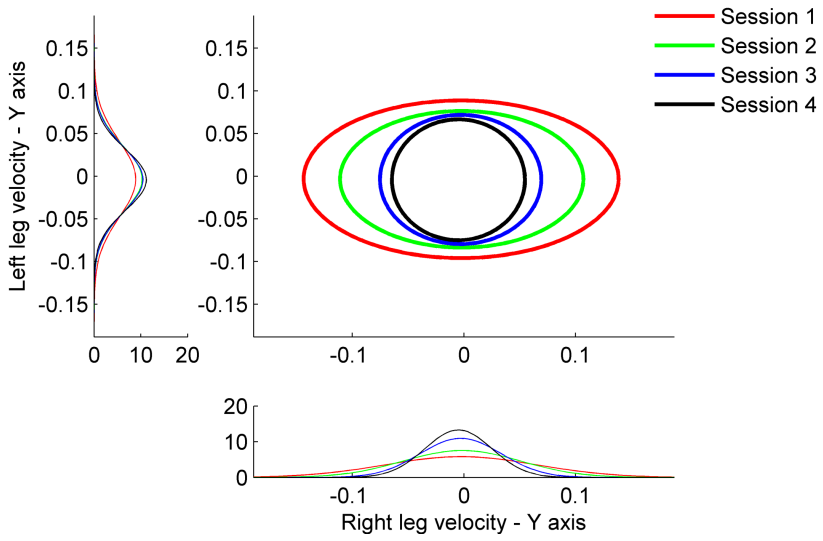

Figure 3.6: Comparison of the joint probability density in left and right foot balance for the Y-axis knee velocity over four sessions with the rehabilitating athlete.

During a balancing assessment the velocity of the supporting leg controls the placement of the center of mass and hence the stability of the body. As an example, the variability of knee velocity of the recovering subject is shown in Figure 3.6 over the 4 sessions. The variance when balancing on the right leg, which had undergone surgery, was greater than for the left leg and decreased over subsequent sessions until reaching equality with the left. Considering additional directions, the largest eigenvalue associated with the velocity of the hip, knee and ankle joints is shown on a symmetry graph in Figure 3.7 where the axes are the eigenvalues from balancing on

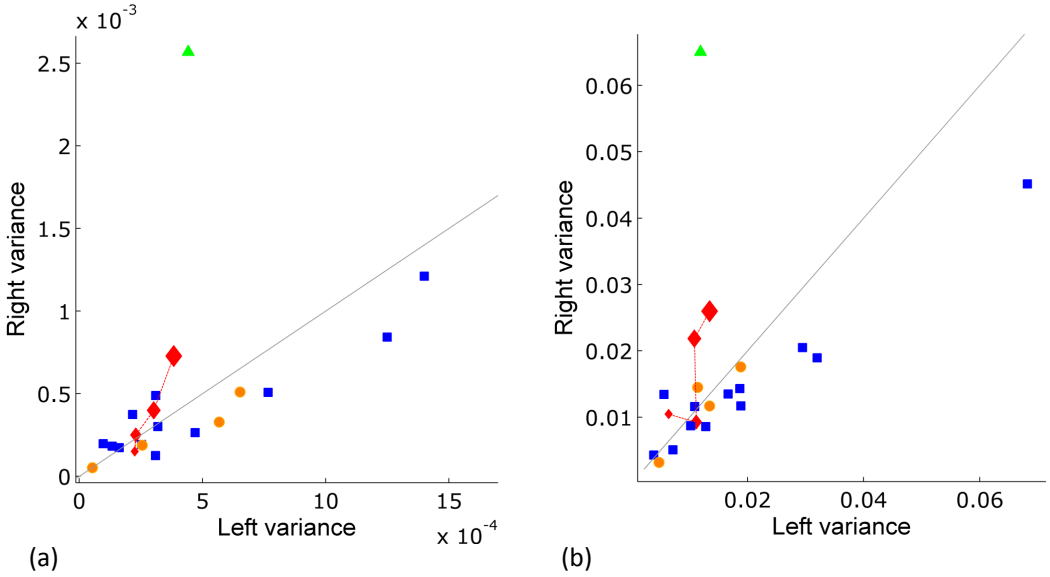

Figure 3.7: (a) Symmetry of static single-leg balance using the largest eigenvalue of x and y-axis velocity features of the supporting leg where each data point is the result of one trial. The recovering athlete is marked in red and the markers are scaled in size from earliest (largest) to latest (smallest). A second injured athlete is marked in green. The remaining points are the control group (blue - male, orange - female). (b) Equivalently labelled symmetry graph for angular velocity.

the left and right leg in the same session. The expected ratio for uninjured subjects would be unity (the diagonal line for reference) with a small bias due to a natural familiarity for one leg or the other. The injured athlete (shown in red) has a higher variance in the right leg balance for the first session and approaches unity on the subsequent trials. The second injured athlete (shown in green) displayed an even higher variance with the right leg balance test.

Cumulative GMM state changes

In extending this analysis to the GMM, the frequency of predicted state changes is also related to the variance observed in the motion. The input data for each GMM was the thigh, shank and foot velocity vectors for all recordings of a participant. The cumulative state changes, with sufficiently sensitive states, are increasing linearly over

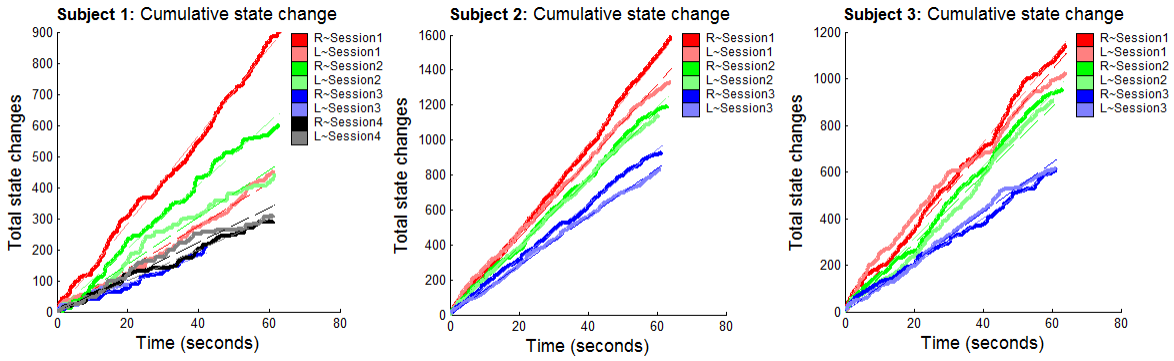

Figure 3.8: GMM cumulative state transition graph for the single leg balance using linear velocity data. Subject 1 (top) was injured and recovered over 4 sessions, while Subject 2 and 3 had suffered no injury.

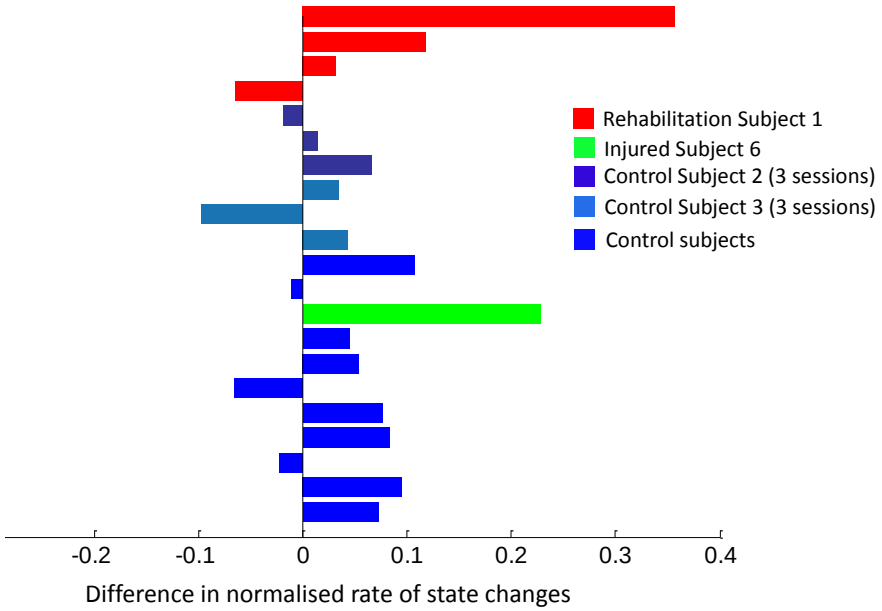

Figure 3.9: Difference in the rate of GMM state changes $(R - L)$, normalised by the maximum rate of change. Positive values indicate a higher rate of change for the right leg.

time as depicted in Figure 3.8. For the injured athlete (Subject 1) it is clear that the gradient of the cumulative graph is significantly different for each balancing leg in the first session. The ratio between these gradients approaches unity over the subsequent sessions, which differs from the relatively constant left-right ratio for the control subjects. In Figure 3.9 we visualise the difference between the gradients,

normalised by the highest rate of change. A large positive value indicates a higher gradient in state changes for the right leg compared to the left. For the two uninjured subjects, the relative difference of state changes between leg balance sessions is small compared to the initial sessions of the recovering athlete. This suggests the measure is detecting more significant changes in the injured subject than the normal variation between sessions.

Centre of Mass Tipping Distance

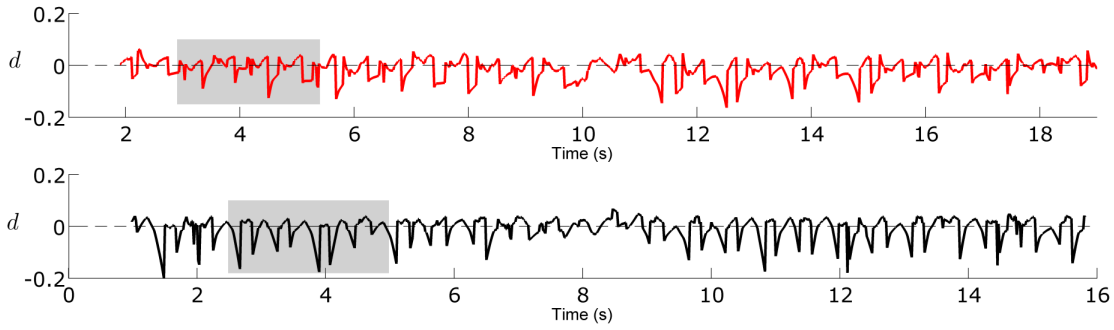

Figure 3.10: Center of mass distance to ground support polygon, d, during a walking motion for session 1 (red) and session 4 (black) with the recovering athlete. The grey region indicates period shown in Figure 3.11 and halfway through the recording the subject slows and turns around.

The center of mass tipping distance was examined for walking motion of the injured subject near the beginning of rehabilitation and at the end. Figure 3.10 displays the distance over time for each recording. In the first session (in red) the distance is seldom below zero compared to the last session (in black), which means $\mathbf{x}_{com}$ was outside the ground support polygon for shorter duration in each gait cycle. Figure 3.11 shows two walking cycles of each session in greater detail. Large decreases in d correspond to the foot lift phase while an increase indicates the time the swinging foot is planted on the ground. The later session is marked by confident strides because $\mathbf{x}_{com}$ consistently resides further from the ground support at the extremities of the

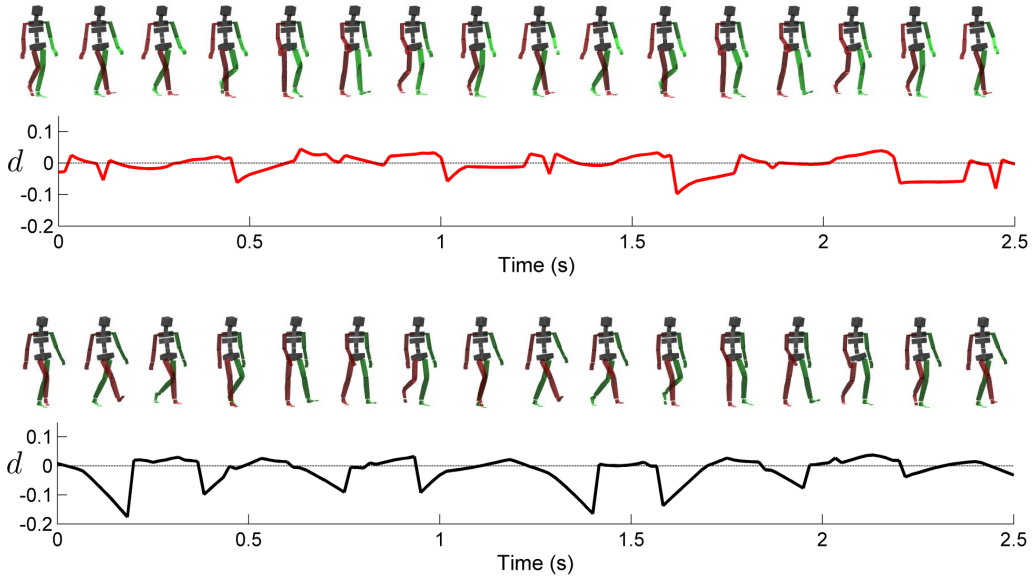

Figure 3.11: Center of mass distance to ground supporting polygon for two walking cycles in sessions 1 (red) and 4 (black) with the recovering subject. Illustrations above each graph show the phase of gait and highlight the significance of the pattern.

single leg phase whereas the first session suggests a more tentative gait.

3.4.3 Discussion

This case study shows that for simple postural stability tests, such as the single leg balance test, a GMM is sensitive to minor changes in posture. The test required the subject to minimise any postural change, so the accumulation of significant postural changes over the test may provide an objective measure of the ability to control stability. Further validation of these analyses in the field of rehabilitation would require additional subjects recorded more frequently during a recovery period. However, the experiment demonstrates the sensitivity of the sensors and the mixture model approach in detecting small posture adjustments when the experiment target was to minimise movement while balancing.

3.5 Functional Capacity Assessments: Case Study

The focus of this case study is the detection of detrimental postures during functional capacity assessments. Evaluating human performance is important for assessing the risk in approving an individual for work involving manual handling. Hospitals and medical centers provide this risk assessment expertise to employers and employees. Among the standard functional capacity assessments is the Progressive Iso-inertial Lifting Evaluation (PILE) [106], which concentrates on the assessment of lumbar lordosis (lower spinal curvature), postural consistency and fatigue. Due to the structured nature of this procedure and reliability PILE is used extensively in practice [107]. PILE is typically used in conjunction with other functional capacity assessments, including quantifying musculoskeletal range and assessing the ability to perform potential job specific tasks, to form a complete picture of the endurance, flexibility and manual handling technique of the candidate [108].

The PILE procedure is conducted as follows. Initially, a set physical measurements of the candidate, including their height, weight and resting heart rate dictate the limitations of the test such as maximum heart rate and maximum lifting weight. Each lift starts from a neutral upright stance, the candidate is instructed to pick up a container which holds a predefined weight, step forward, place the container onto a bench set to the elbow height of the candidate, then to retrieve the container and place it back on the ground. This action is then repeated for a second time and the subject is expected to complete both sequences of actions within 20 seconds. The weight in the box, starting at 5kg for male and 2.5kg for female candidates, is increased after each successful set of lifting actions until the mass reaches 55% of their total body mass, or the subject, in the judgement of an overseeing clinician, exceeds a set of constraints listed in Table 3.3 (A-D). This procedure is summarised in a block diagram in Figure 3.12.

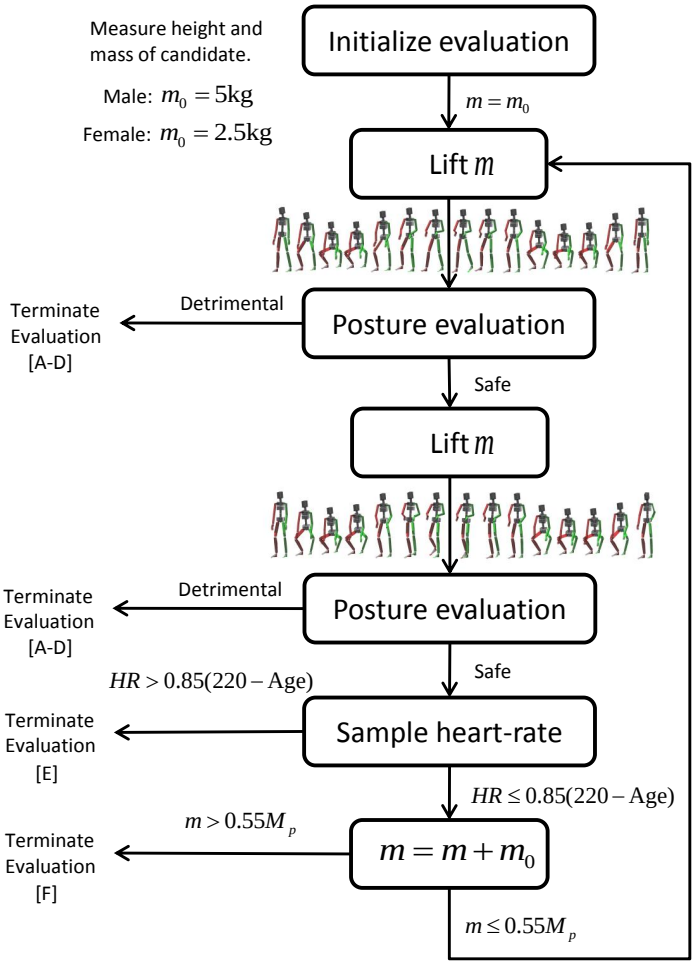

Figure 3.12: Block diagram of the PILE procedure where each cycle is scheduled to be performed in under 20 seconds.

The first four constraints are specific postural adaptations identified by clinicians as detrimental when lifting heavy objects and are reactions commonly observed in conducting the PILE. For instance, the loss of lordosis refers to the loss of the inward curvature of the lower spine due to the subject leaning over while lifting the weight. The counter balancing lifting motion is an attempt by the subject to partially support the weight using the hips.

The central motivation is to detect posture adaptations or biomechanical changes in the lifting action observed. Any repetitive action will naturally vary between

Table 3.3: Reasons for ceasing the evaluation as reported by clinicians.

Termination	Description
A	Accessory muscle use
B	Loss of lordosis
C	Protracted shoulders
D	Counter-balancing
E	Exceeded maximum heart rate
F	Exceeded weight limit

iterations, however, in this context we seek a quantitative description of the acceptable range of lifting postures as indicated by the observing clinicians. Feedback on the consistency of the recorded postures can support the decision taken by the clinician and identifying quantifiable boundaries on the motion leading to a standardisation of the evaluation across experts [109][110].

To investigate the standardisation of these assessments a group of test subjects were recorded while performing the PILE under the supervision of two clinicians. A detection algorithm is tested for identifying significant posture adaptations, which are subsequently analysed in detail.

This forms a precursor to segmenting different actions or classes since we have analysed the acceptable range of a particular action and rely on an observer to identify when this prescribed action fundamentally changes.

3.5.1 Experimental Setup

The experiment involved recording 12 subjects, 6 female and 6 male, (height 165 ± 15 cm, weight 74 ± 19 kg and age 34 ± 13 years) with the MVN motion capture system. A heart rate monitor was strapped around the chest underneath the participants clothing. Each participant formally consented to the recording and risks of the test as approved by an ethics committee. The weights, as the experiment progressed, were placed in a $30 \times 30 \times 30$cm box with handles near the top edge. In each experiment

Figure 3.13: Experimental PILE scenario with one subject lifting the assigned weight under the observation of two clinical physiotherapists. From left, the subject starts in a neutral standing posture, reaches for the target box whilst maintaining lordosis of spine, then deposits box onto a bench at a height such that the forearm is horizontal.

the box was placed a distance of one step away from the bench to which the weight is delivered. The height of the bench was adjusted to the height of the subjects elbow when holding the box, as shown in Figure 3.13. This is to avoid plausible compensatory postural adaptations for candidates of differing height, when depositing or retrieving the box.

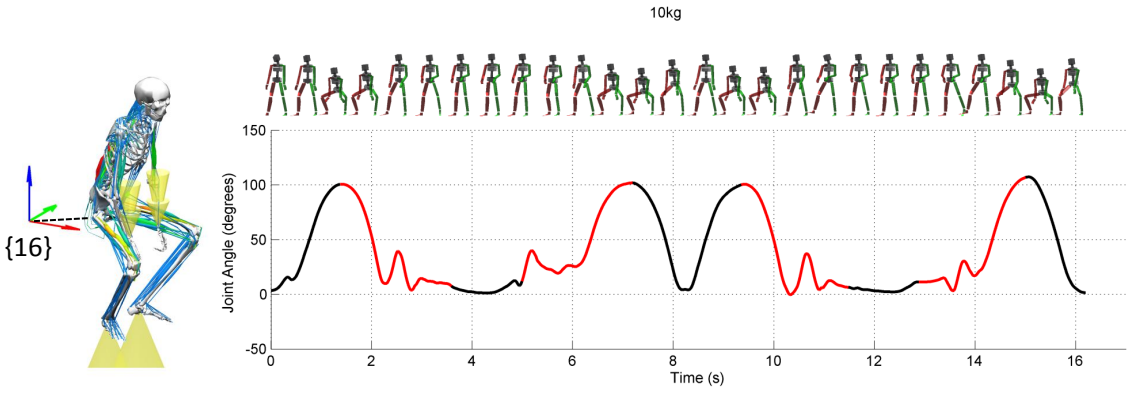

Figure 3.14: Segmentation of the motion into lifting phases using the Y-axis of the hip joint as identified on the left. The joint angle is shown in the graph where the red periods indicate the lifting and dropping motions as segmented by a threshold on the angle.

3.5.2 Data pre-processing

In applications involving the analysis of large sequences of data it is useful to partition the sequence into meaningful segments, especially in cases where the crucial data points are known to be within particular time frames. For the PILE motion the important periods are when the weight is being raised and lowered by the participant. Using this prior knowledge of the task the motion is segmented by monitoring the hip joint angle. The start of the lift action and end of the lowering action occur at the peaks of the hip angle as depicted in Figure 3.14. An alternative automatic segmentation process would be to attach a sensor on the box and partition the data based on the acceleration of the box.

The set of joint angles considered in the analysis were the pelvis, L5, T8, neck, right and left shoulders, upper-arms and forearms. These features were normalised per subject in order to identify a standardised threshold for biomechanical changes.

3.5.3 Results

The actions identified by the clinician as being detrimental can be examined by noting the differences in joint angles between the final motion and all preceding motions. In Figure 3.15, two joint angles are compared, for the first, penultimate and final phases of the test, to indicate the apparent change. The subject compensates for increased weight with a counter balancing posture, which is characterised by further leaning back while upright and holding the box closer to the body.

The analysis presented here seeks to detect these differences in a systematic manner. Firstly, an approach is proposed for the separation of detrimental lift postures using spectral analysis of a sequence distance measure derived from DTAK. The identified changes are examined in further detail for a particular case using biomechanical simulation [111], which validates the observation of a biomechanical change. Finally,

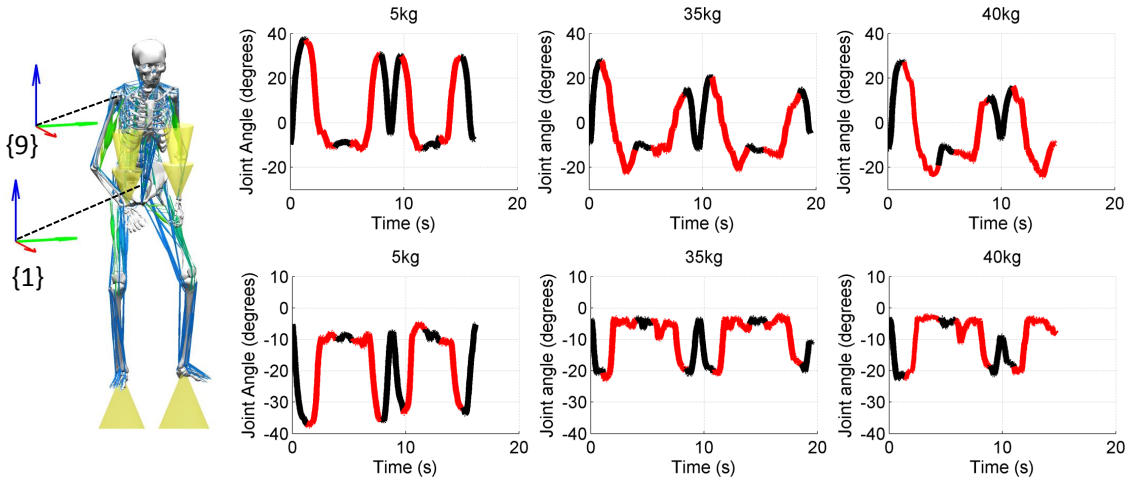

Figure 3.15: Features depicting the postural changes described as counter-balancing. In the upright stance the weight is supported on the hips of the subject. This posture is described by the pelvis angle around the sagittal axis (top) and the y-axis rotation of the upper arm (bottom). The subject leans back (pelvis angle) and holds weight closer to the torso (upper arm angle). The periods highlighted in red are when the subject is either lifting or depositing the mass.

the extension of this technique to data representations in terms of motion primitives is discussed.

Detection of significant posture changes

A distance matrix is constructed quantifying the similarity between all pairs of lift sequences, i and j, using the DTAK distance metric with a lower bound (Sakoe-Chiba band) of 50% of the sequence length. The relative separation between the sequences can be evaluated by examining the spectrum of the resultant similarity matrix, a process related to spectral clustering. A maximum margin separation between the latest lift sequence and all previous sequences quantifies the extent of any change. The margin was computed with a linear SVM with $\|\mathbf{w}\|$ relating to the separation distance.

An example of this process is shown in Figure 3.16, where the distance is calculated by performing DTAK on all of the features jointly. The final lift action is separable

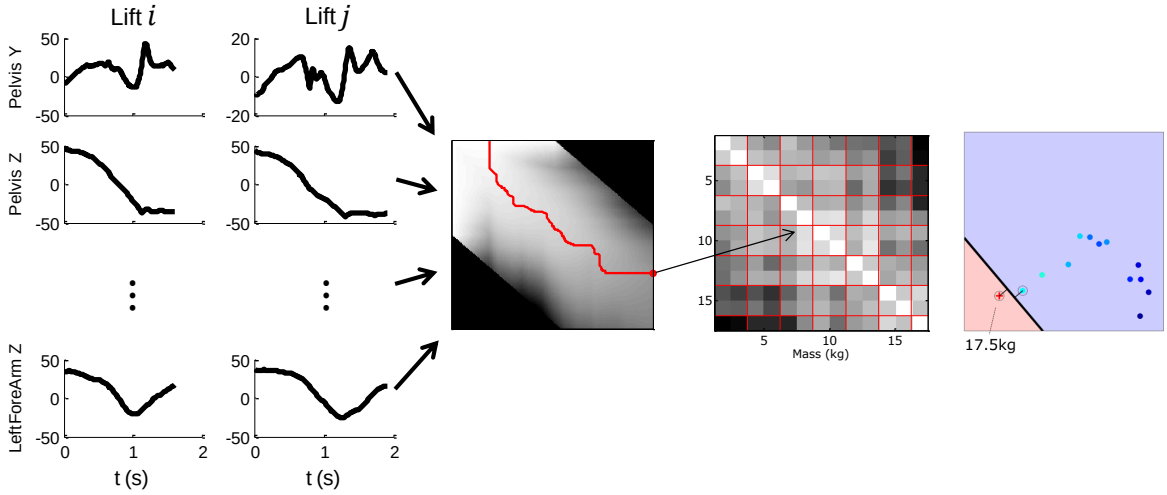

Figure 3.16: Spectral distance when computing DTAK of PILE features jointly. From left, two sequences i and j described by all the features are compared with DTAK resulting in a cost matrix and warping path (in red). Distances between all sequences are represented in a matrix where darker entries indicate high dissimilarity and red lines mark increases in mass. The spectral margin between the final sequence and previous sequences is displayed in 2D where lifts are coloured from dark blue as lowest mass to light blue as highest mass.

from the previous sequences by a small margin. In Figure 3.17, however, DTAK is calculated on each feature separately and as a consequence the margin detected for different joint angles can vary significantly highlighting the contribution of each feature to the change of lifting posture. Figure 3.18 summarises the distance margins detected across all subjects and features for the final lifting or lowering action where they were halted by the physiotherapist. When the sequence for feature i was not separable the margin was set to zero.

It is clear from this analysis that the posture adaptations vary between subjects. Despite significant variation in the terminating postures one can still test the efficacy of a threshold on the distance margin. After determining a threshold based on the set of distances that terminate the evaluation for a training set of participants we apply this threshold incrementally (on each lift in turn) for the remaining test candidates. The result is shown in Figure 3.19. Both subjects were stopped for a counter-balancing

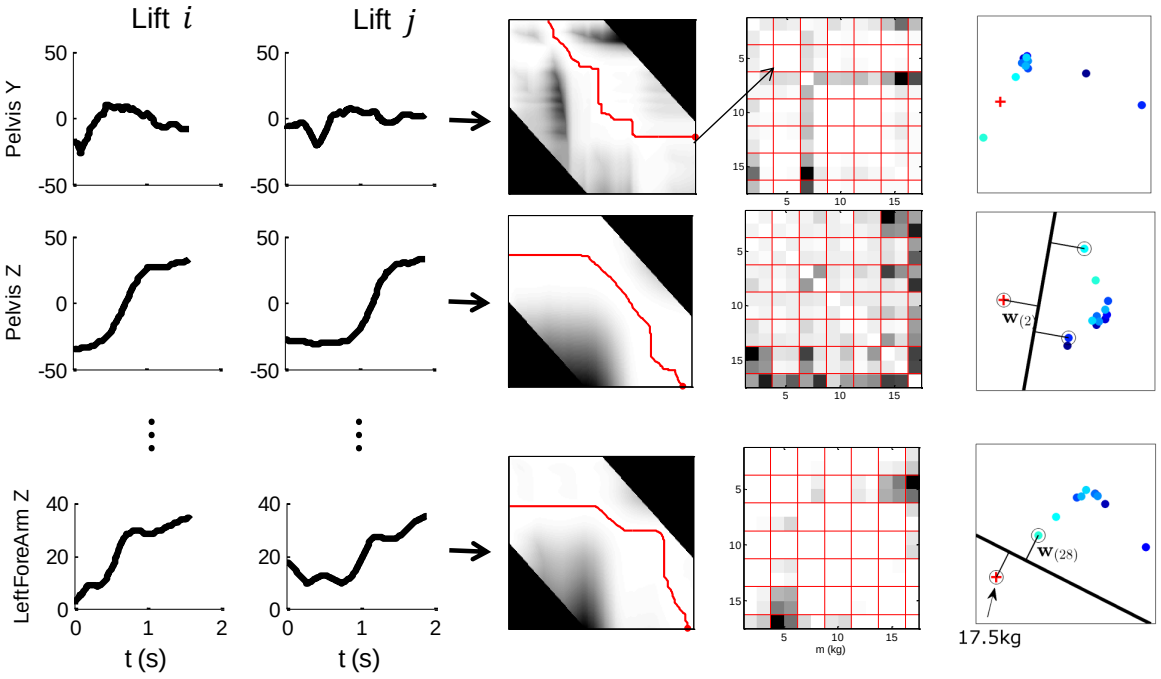

Figure 3.17: Spectral distance when computing DTAK of PILE features separately. From left, two sequences i and j are compared with DTAK resulting in a cost matrix and warping path (in red) for each feature. Distances between all sequences are represented in a matrix where darker entries indicate high dissimilarity and red lines indicate increases in weight. The spectral margin between the final sequence and previous sequences is displayed in 2D on a per feature basis where lifts are coloured from dark blue as lowest mass to light blue as highest mass.

lift action which is commonly observed in a upper arm angle along the Y-axis (normal to sagittal plane). Subject 8 adjusted their posture on the final lift, which is in agreement with the observation of the physiotherapist. However, for Subject 5 the change was detected on the penultimate lifting action, which was the first experience of 40kg, and is the reason why on Figure 3.18 the distance margin is not significant for this participant. This amounts to evidence that the evaluation should have been stopped on the penultimate lift and shows that further physiotherapist assessments are required to determine a ground truth to compare the detection algorithm against.

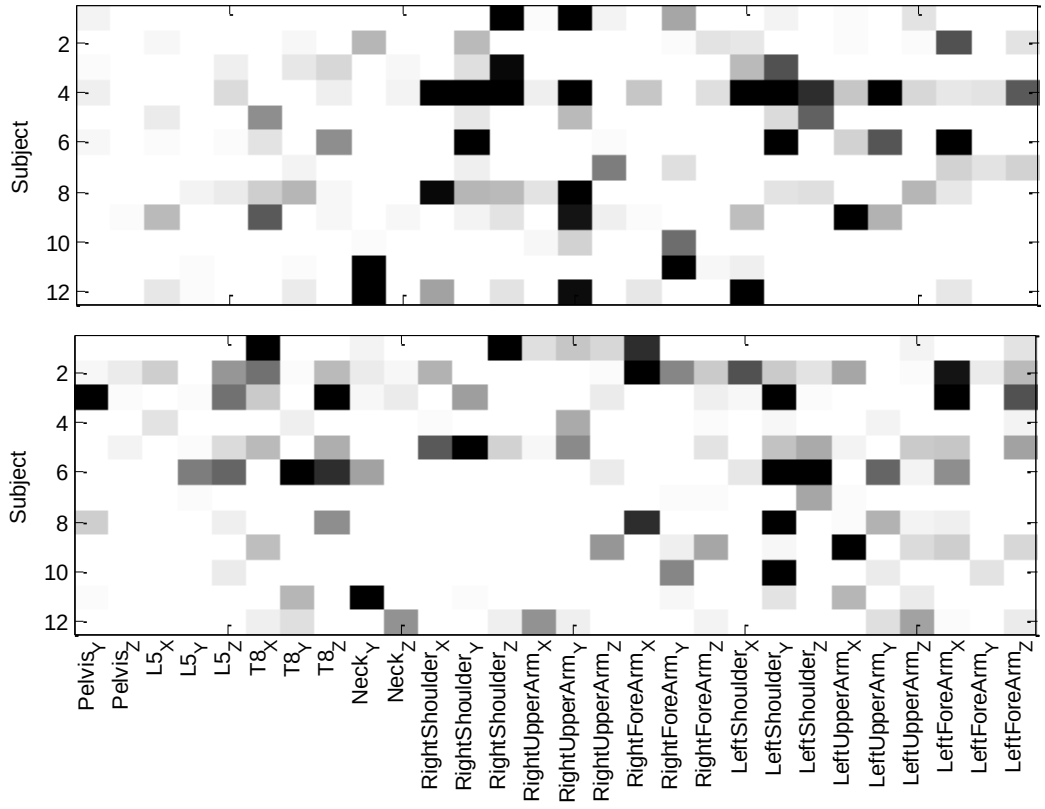

Figure 3.18: Distance margin calculated for the last two observed sequences, lifting (top) and dropping (bottom) of the mass, split into subjects and features. Darker entries indicate higher distances and therefore are features which have triggered a biomechancial change.

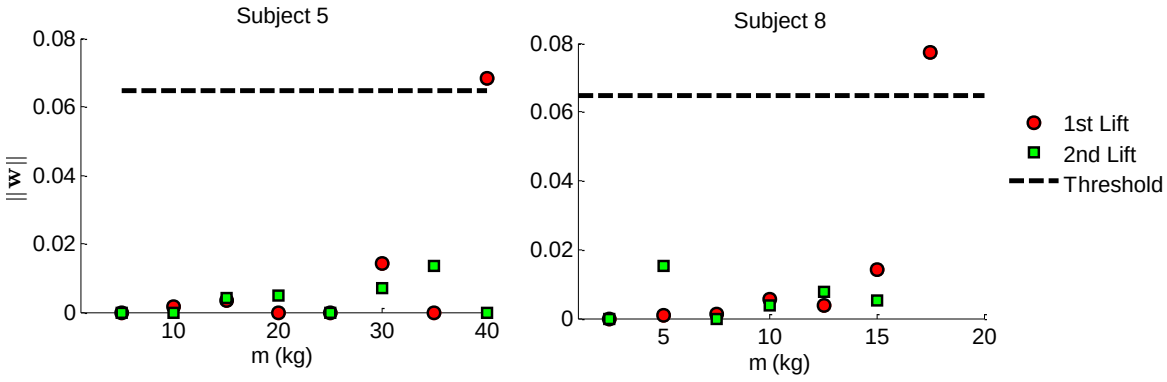

Figure 3.19: Application of a spectral distance threshold incrementally as the mass is increased in PILE for two subjects. The threshold is trained from other observations. It is exceeded in these two example cases on the penultimate and final lift actions. For Subject 5 the margin of the last lift is small and the postural change occurred on the penultimate lift.

Biomechanical validation

To visualise and validate the clinical decision we focus closely on an example through biomechanical simulation software [111]. Subject 4 adjusted their posture, by shifting the angle of both shoulders, in the final increment of weight in the experiment. This biomechanical adjustment was referred to as protracted shoulders. To simulate the likely muscle activations the appropriate weight was applied as vertical force evenly

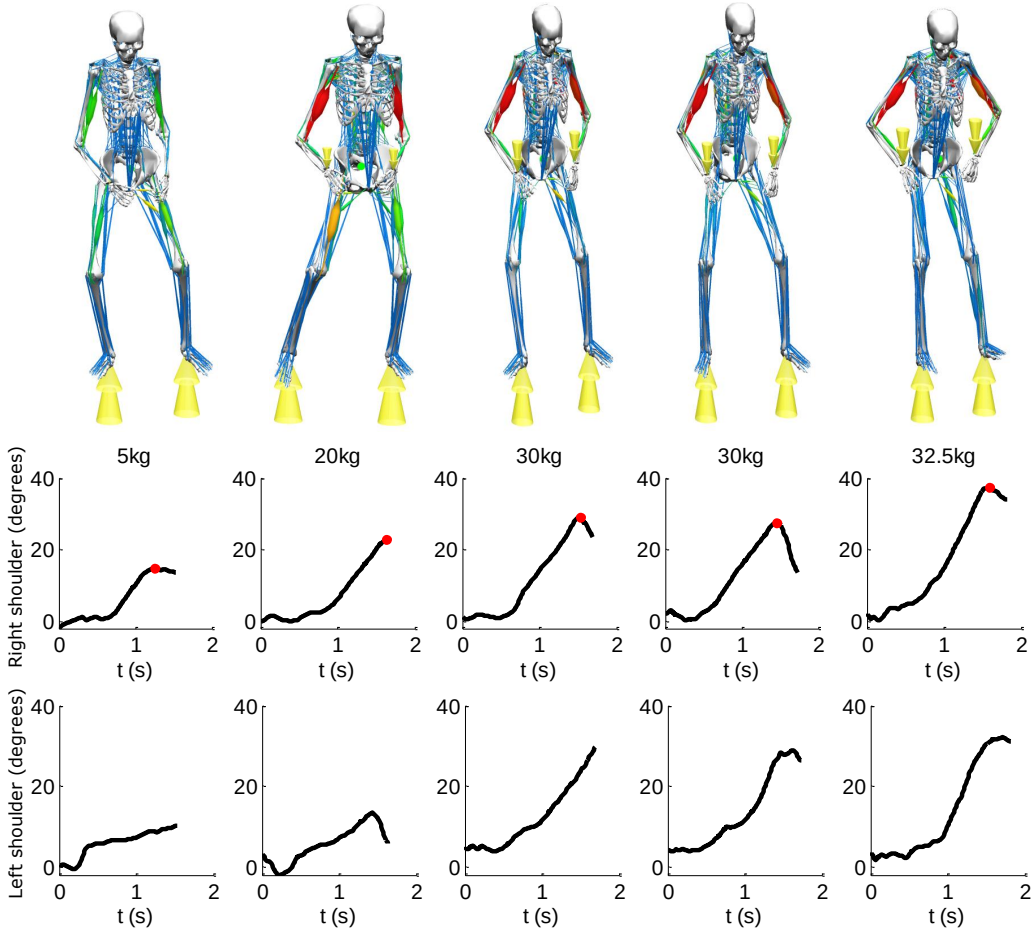

Figure 3.20: Visualisation of the progress in muscle forces and joint angle changes throughout the PILE with Subject 4. Arrows indicate the location and direction of simulated forces from body weight (on feet) and the box(on hands). Muscles highlighted in red indicate an increased level of activation, it is particularly pronounced in the final lift in the biceps and across the back. Joint angles are also displayed for each of these lifting actions and show that the maximum extent of the shoulder extension is clearly increasing.

between the hands and the weight of the participant applied as a force evenly between the feet. The relevant shoulder signals are displayed in Figure 3.20 along with the posture and muscle activation for the peak of the shoulder extension. It is clear from the posture that the elbows extend further in the final lift. This increases the torque required at the shoulder joints despite the smaller increment in weight of 2.5kg (to avoid the maximum of 33kg for this subject) as observed in Figure 3.21. If one examines the standard deviations of all muscle activations across all sequences, as in Figure 3.22, the final sequence exhibits high muscle activation for levator scapulae, serratus anterior and anconeus. They relate to the raising of the shoulders, tensing of the neck, and of the elbow respectively. This is a combination of accessory muscles that would expected to activate in this posture, validating the clinician decision.

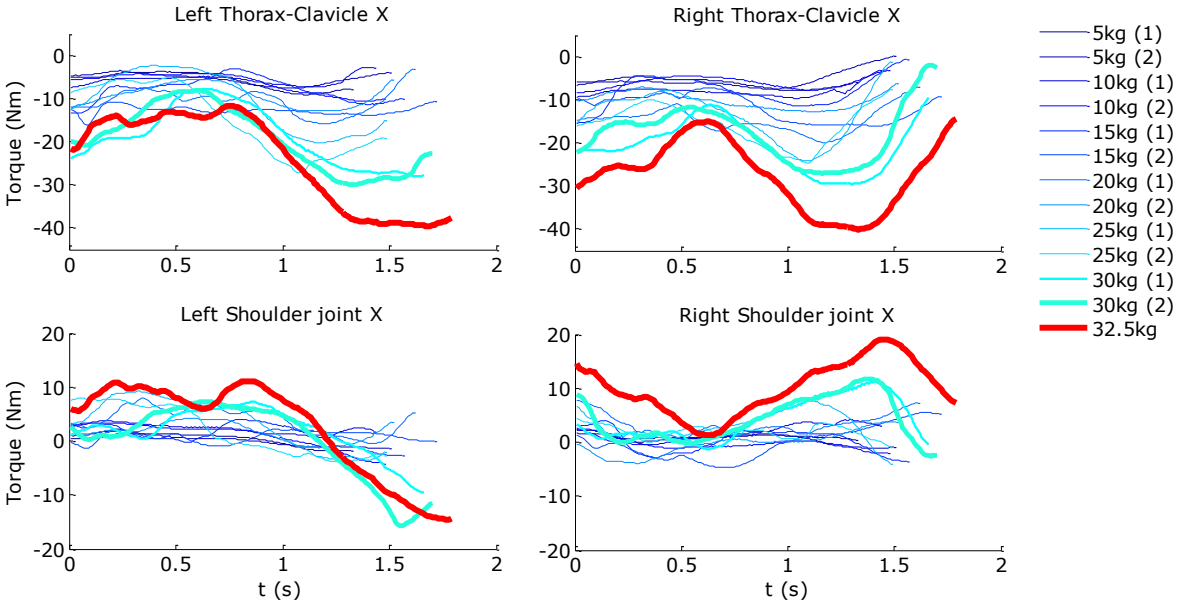

Figure 3.21: Joint torques for shoulder and upper arm on Subject 4. The signals are highlighted by the associated weight and the red trace belongs to the last observation.

As an additional case, Subject 3 was halted in their evaluation due to a loss of lordosis. In Figure 3.23, the posture is shown in sagittal plane for the first, penultimate and final lift. In the final sequence the subject leans marginally further forward

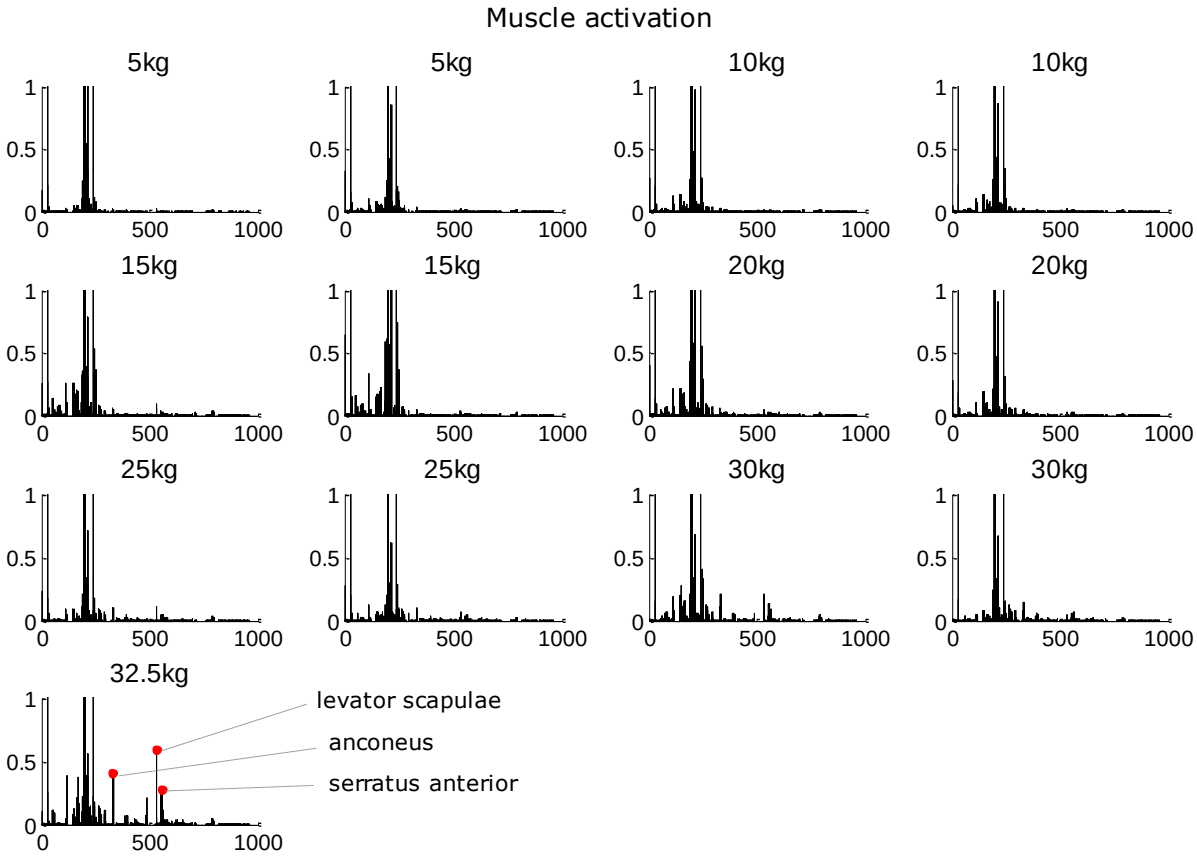

Figure 3.22: Standard deviation of the muscle activations on y-axis and muscle index of the x-axis for each lifting sequence. The final lift on 32.5kg is characterised by accessory muscle use in the shoulders, neck and elbows.

and higher stress is placed on the lower back muscles through the lift. The muscles are highlighted on a continuum from low activation (green) to high activation (red).

Detection of changes with motion primitives

The approach for the detection of postural change can be extended to the concept of motion primitives. After learning a mixture model on a training data set composed of the majority of the data, an unobserved subject is tested against the model by predicting the likely states at each time instance. An example unsegmented sequence with the predicted posture states from a GMM is shown in Figure 3.24, where the states

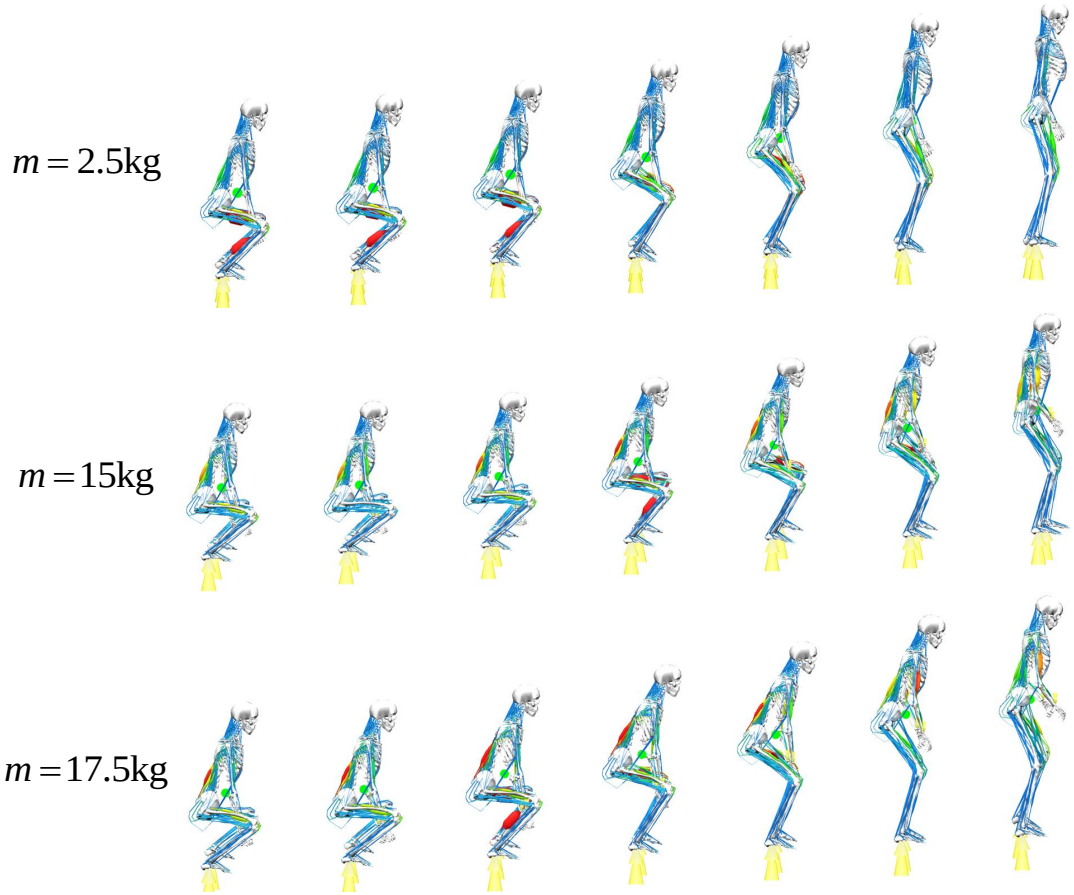

Figure 3.23: Visualisation of the posture during the first lift of mass 2.5kg, penultimate lift of mass 15kg and final lift of mass 17.5kg. The evaluation was terminated due to a loss of lordosis, which increases stress on lower back muscles. The muscle stress is indicated by highlighting the muscle with low stress in green and high stress in red.

are ordered by the magnitude of the hip angle. A change in the predicted sequence of states will then correspond to a biomechanical change. In a similar manner to the previous analyses we construct a kernel matrix based on the KL divergence between each state in the sequence. The DTAK algorithm is once again used to determine a sequence similarity measure. Sequence distance matrices and the associated spectral projection are displayed in Figure 3.25 for 3 subjects. The results bear a similarity to the previous section since Subject 4 exhibits a significant change in the final lift, while Subject 5 adjusts posture in the penultimate lift as also shown in Figure 3.19.

The advantages in learning a mixture model to cluster postural changes in a clini-

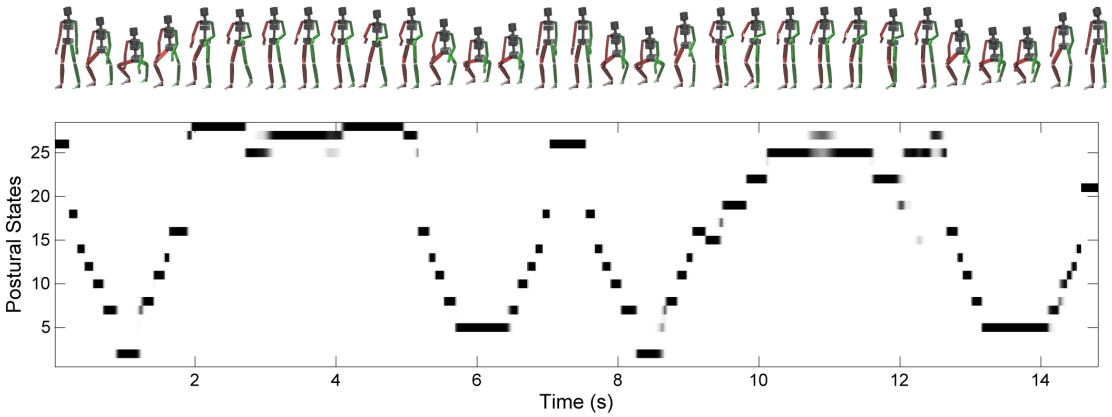

Figure 3.24: Graph of the posterior probability of the data given a GMM or set of motion primitives trained with the angles and velocities of the body, where darker regions indicate higher probability. The motion primitives are ordered by the magnitude of the knee angle. Only the motion from the first stage of the PILE (lifting 5kg) is shown. Above the probability graph is a visualisation of the pose throughout the motion.

cal test are that uncertainty with which change is observed is quantified. Additionally, the computational efficiency is increased if repeated sequences are commonly detected since the DTAK sequence distances have been precomputed. Therefore the similarity between sequences and the subsequent detection of significant posture changes could be reported in real-time.

3.5.4 Discussion

Clustering the lift actions highlights the change in the final stage of the PILE, however, the underlying causes of the changes when examined are difficult to relate across subjects. This is due to the small sample size. As additional detrimental postures are observed the causes may coalesce into categories relating to the Table 3.3. When clustering a larger cohort of subject data there could be an issue with the individual sensitivity of each subject. That is, if the scale of the changes are of a much lower magnitude for some patients that would lead to false negatives. A larger sample size would allow a more detailed analysis of these issues.

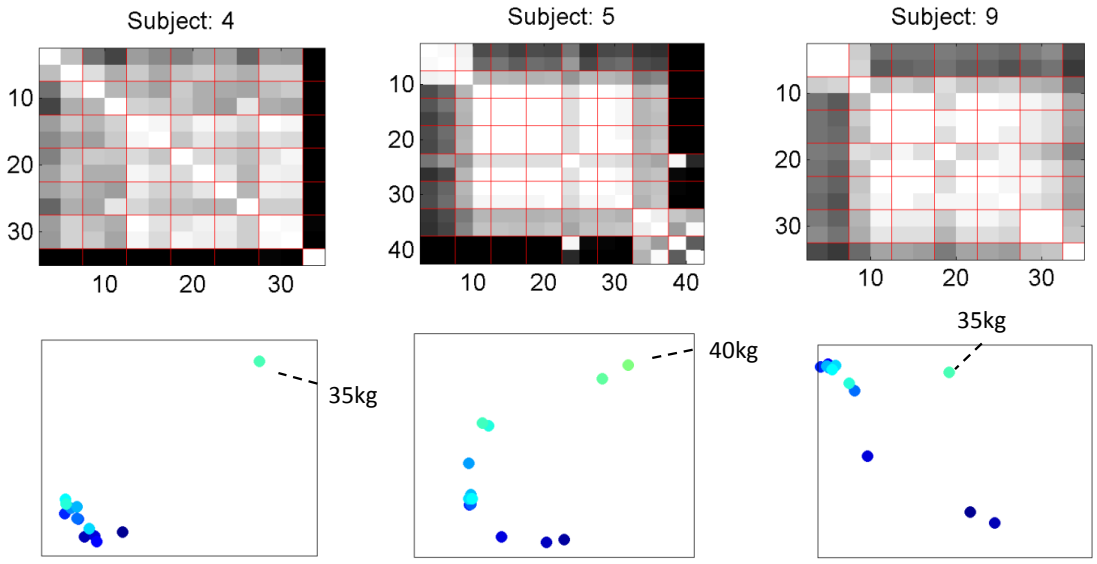

Figure 3.25: DTW distance between pairs of lifting actions in the PILE as described by motion primitives (top). Each element ϕ_{ij} is the distance between sequences i and j. Each sequence is projected onto a two dimensional subspace computed from the first two eigenvectors of the distance matrix (bottom). The points in the two dimensional eigenspace are coloured indicating a scale from low weight (dark blue) to high weight (light blue). In each case it is clear that the highest weights can be separated from the rest. For Subject 5 the last two are significantly different.

A plausible future direction is to introduce hidden variables, relating to either muscle or cardiovascular fatigue, which would modulate the quality of the observed action in the eyes of the clinician. In [112], a HMM of motion features was used to characterise fatigue in a squat exercise. This produced consistent results despite the constraint imposed on the number of fatigue states. A model of fatigue may also benefit from additional observations such as heart rate or selective surface EMG. Inferring a HMM structure from the data with nonparametric methods may reduce these model design biases [39].

The identification of significant or meaningful changes in the posture is a precursor to the segmentation and recognition of arbitrary actions for programming by demonstration which will be the topic of the following chapter.

3.6 Chapter Summary

In this chapter an analysis of incrementally changing motion profiles was presented. We discuss preliminary concepts of segmentation and recognition of repetitive motion. This type of motion analysis is important in the field of rehabilitation tracking. Two case studies were presented in the areas of injury rehabilitation and functional capacity assessment. For injury rehabilitation we tracked changes in movement variance over a recovery period of 12 weeks and the exercises and sensor system were able to detect a significant change. In the functional capacity assessment case, an approach for the detection of incremental postural changes was presented based on a dynamic alignment with previous observations. The results demonstrated the viability of a clinical decision support system which quantifies the relative divergence of a posture from previous postures. The significance of the postural change was revealed by accessory muscle activations in a biomechanical simulation. Importantly, we show how it is a simple extension to utilise GMM-based motion primitives to potentially improve the efficiency of the detection in real-time and also as an introduction to the time series segmentation and classification used in the remainder of the thesis.

Chapter 4

Temporal segmentation and recognition of human motion

4.1 Introduction

The observation of continuous demonstrator motion raises the problem of parcellating relevant subsequences suitable for learning. This process is typically addressed by either employing additional input for which the demonstrator can indicate the relevant sequences or by manual preprocessing of the data. Other alternatives have derived time series segments automatically from a known structure to the task or by deploying algorithms which seek to segment sequences based on their statistical differences [42][113]. The advantage of automatically organising sets of demonstrations is that it not only reduces the workload of a demonstrator, it also contributes towards systems of autonomous incremental learning, where there is always a possibility of observing a new category of action. The learned models can be further used to classify actions and identify styles by examining the constituent motion primitives. In this chapter the hypothesis is that demonstrator actions can be reliably classified from automatically segmented time series.

As a prelude to the approach presented in this chapter, Figure 4.1 indicates the intended outcome of a segmentation algorithm. A time series is observed corresponding to human motion devoid of any prior information on the likely categories of motion. One desires a segmentation to cluster the motion into separate meaningful categories based on a similarity measure over sequences of motion primitives. In Figure 4.1 there are six unique actions indicated by distinct colours.

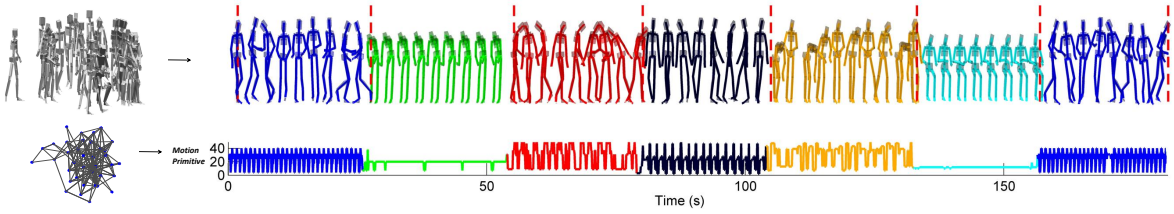

Figure 4.1: An example result from the algorithm presented in this chapter. Given a continuous data set of arbitrary motions (top left) the target is to discover transition events or segmentation points (dotted red lines) and organize the motion into distinct groups as indicated by the colour code in the figure. This is achieved by representing the motion with a Gaussian mixture model and analyzing the set of states (bottom left) as a sequence of motion primitives over time (bottom right), where each motion primitive corresponds to a component in the mixture model.

The chapter is organised as follows. Initially, a data collection is outlined in order to motivate the segmentation algorithm. The motion primitive mixture models are reviewed before we outline a segmentation procedure utilising the sequence of primitives. The algorithm is tested in two scenarios. First, the unsupervised clustering of segments is examined in the scenario where no *a priori* category information is available. With the inclusion of labels in a supervised learning context the recognition is also tested and compared against other common discriminative methods [114].

4.2 Experimental Setup

An experiment was designed specifically to test the continuous segmentation and recognition of human actions resulting from motion capture data. The motion capture

equipment deployed for the recording is the same as was introduced in Section 3.3 of Chapter 3.

4.2.1 Procedure

A group of 10 subjects (6 male, 4 female), with heights ranging between 150-187cm, performed a continuous series of 10 activities in a random order per recording. A group of 4 of the initial participants returned at a later date for a second recording of the same actions in a new sequence to assess the adaptability to subjects.

The set of 10 actions classified using the sensor units were: walking (A_1), jogging (A_2), sweeping the floor (A_3), bouncing a ball (A_4), sitting in a chair and reading (A_5), backwards walking (A_6), sitting and packing a bag (A_7), standing while operating computer (A_8), lifting a box onto a table (A_9) and balancing on one leg (A_{10}). Classes such as walking, sitting and standing are commonly considered in daily activity monitoring studies [115]. Lifting and sweeping were derived from functional capacity assessments based on typical work related motion [109] (Chapter 3), while jogging, balancing and ball bouncing are based on sport related actions that may reside in a common training program [103].

The selected set of actions were intended to be unconstrained and lead to significant signal variation between participants. Each action was repeated or sustained for a period of between 40-45 seconds before the subject was instructed to progress to the next, such that each recording lasted approximately $7\frac{1}{2}$ minutes. Each action was described to the subject once prior to the recording to limit the hesitation in transitions between actions. However, minimal instruction was provided on the way to perform the motion such that each subject was free to interpret the actions in their own style. A single observer assigned the class labels to all data points. Data was assigned to the null class during transitions between the actions when the pos-

ture did not resemble neighbouring classes. All motion was constrained to an area of approximately 4m^2 for the purposes of capturing video of the session.

This procedure leads to a data set with relatively clear transitions between abstract action categories, some of which are periodic at different time scales. The variations in the data arise from discrepant skeletal structure of each participant resulting in different set of joint angles for each action. Additionally, each participant interprets the actions slightly differently leading to large variations in actions such as sweeping and bouncing a ball. For example, in sweeping, participants held the broom at different locations, were leaning over depending on their height and sweeping to different extents and speeds. Ideally the classification of these activities should be invariant to all of these factors, assuming sufficient variation is included in the training data.

4.2.2 Features

There are several sets of features or representations which may adequately describe the variation in motion between the actions, including 3D joint angles, θ, angular velocities, ω, relative joint positions, p, linear velocities, $\dot{p}$, or accelerations, $\ddot{p}$. The features used were the set of joint angles with the exception of the pelvis yaw angle, since this angle was clearly independent of any action, and the linear velocity of each joint rotated relative to the pelvis sensor frame. Since the kinematic model is a 23 segmental body where $\theta \in \mathbb{R}^{N\times(23\times3-1)}$, $\dot{p} \in \mathbb{R}^{N\times(22\times3)}$ a data matrix is defined as $\mathbf{X} = [\theta \quad \dot{p}] \in \mathbb{R}^{N\times d}$, where $d = (23\times3-1)+(22\times3)$ and a set of manually assigned labels $\mathbf{Y} \in \{0,1\}^{N\times c}$, where N is the number of data points and c is the number of actions. Figure 4.2 shows joint angle data for three of the recordings, two of which are from the same participant, denoted S_{10} for Subject 10 and S_{5a} for the first recording of Subject 5. The class label is indicated below each graph by a coloured bar and

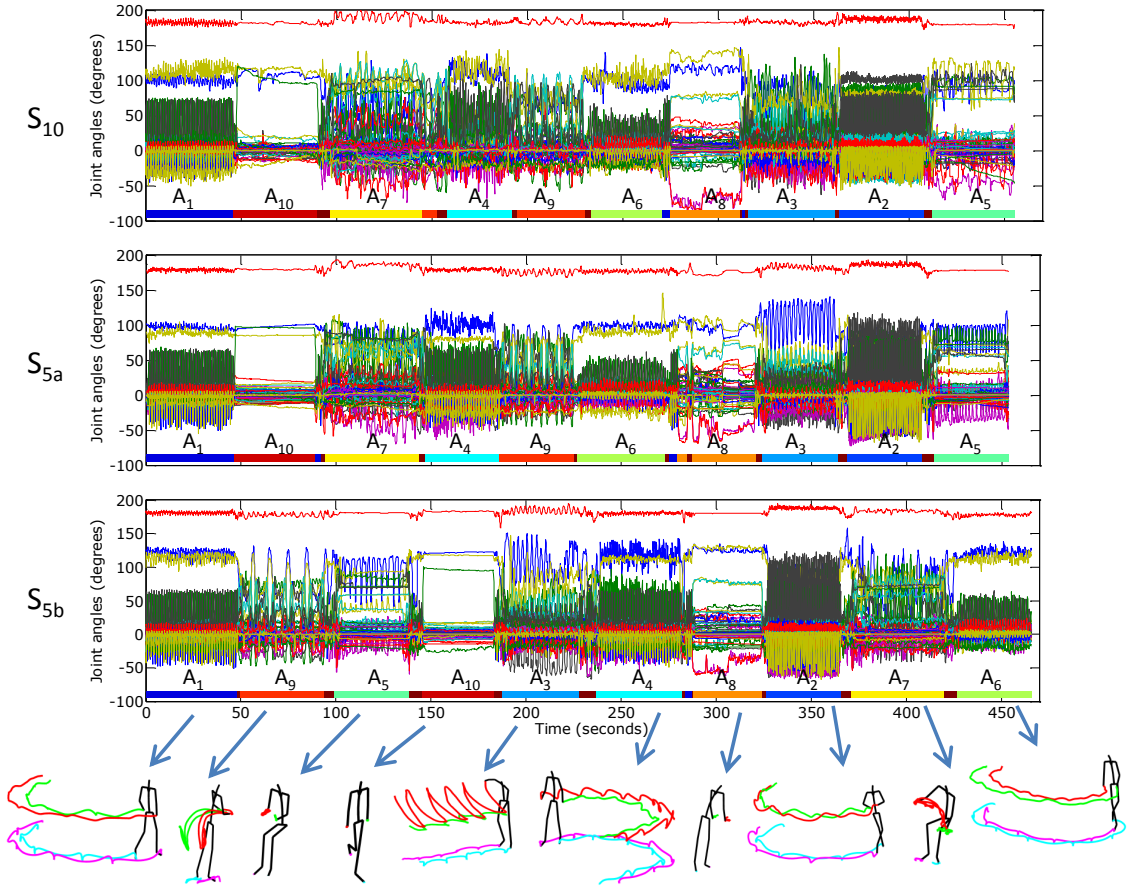

Figure 4.2: Joint angle data from 2 subjects are shown above where the third data set is the repetition for Subject 5. Below each data set is the ground truth labels indicated by a coloured bar and the label A_{1-10}. The sections in between the coloured bars are considered the null class where the motion does not belong to one of the known classes. Visualisations of the motion as a stick figure are shown at the bottom.

label A_{1-10}. Below the joint angle graph is a visualization of each action.

The data was down-sampled using a mean sliding window to 10Hz producing $N \approx 63\ 000$ samples $(10 \times 45 \times 10 \times 14)$ where each of the 10 activities lasted for approximately 45 seconds at 10Hz for 14 recordings.

Experimentation was performed with and without transforming the features using Principal Component Analysis (PCA). It was found that the algorithms presented here attain similar performance in both cases.

4.3 Segment Clustering Approach

Many meaningful actions comprise of a sequence of sub-movements which we refer to as motion primitives. This approach in essence employs mixture models to organize the actions into a set of unique components combined with a segmentation approach to identify transition points between significantly different sequences.

4.3.1 Clustering

There are a number of ways to represent the data using parametric distributions as a precursor to recognition. Many authors have demonstrated that multivariate Gaussian distributions provide a useful assumption for modeling human motion [116]. The Gaussian Mixture Model (GMM) has also been adopted here for clustering the motion data into a sequence of hidden components, each described by a mean and covariance.

Each of the N data samples is assumed to be drawn independently and identically from one of a set of K multivariate Gaussian distributions indicated by a latent membership binary variable $\mathbf{Z} \in \{0,1\}^{N\times K}$. If the data point $\mathbf{x}_n$ belongs to the cluster k, for the indicator variable $\mathbf{z}_n$, $\mathrm{z}_{nk} = 1$ and 0 otherwise. Each of the K distributions appear with a probability π_k where $p(\mathrm{z}_{nk} = 1) = \pi_k$, such that the probability distribution can be described as

$$p(\mathbf{x_n}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.1}$$

$$\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{\exp(-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k (\mathbf{x}_n - \boldsymbol{\mu}_k))}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}, \tag{4.2}$$

where $\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the multivariate Gaussian probability density function (pdf) with mean $\boldsymbol{\mu} \in \mathbb{R}^{d\times K}$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d\times K}$ for a model of K clusters evaluated

at data vector $\mathbf{x}_n$.

Maximum likelihood learning is generally handled by applying the Expectation Maximization (EM) algorithm [117] but the estimation of K requires model selection methods of which there are several proposed in the literature [118, 119]. The strategy applied in this work is to evaluate models using a minimum message length (MML) encoding technique, whereby the minimization of a message containing a model description and the data is equivalent to maximizing the log-likelihood [120]. There are various heuristics involved in the search algorithm including splitting, re-merging and swapping clusters. The parameter K is increased until there is negligible improvement in message length reduction as indicated in Figure 4.3.

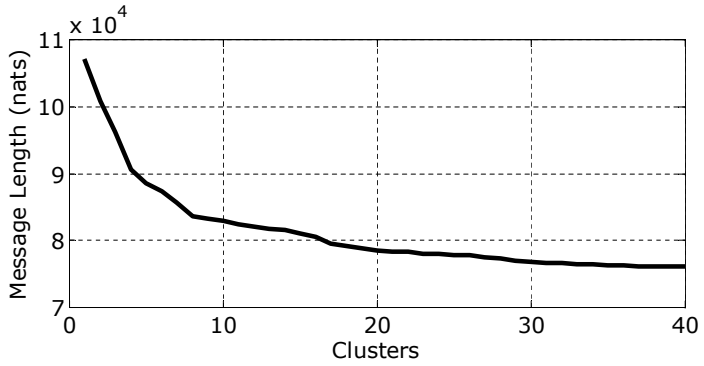

Figure 4.3: Selection of number of clusters using MML.

Aside from dealing with model selection, there are a number of ways to partition the data due to the structure of the experimental procedure. While devoid of any example labels, a GMM must be learned for the entire data set. The similarity of multivariate Gaussian distributions may then be assessed through the Kullback-Leibler divergence (KL) [121], where the divergence between a pair of distributions, $\mathcal{N}_1$ and $\mathcal{N}_2$ is given by

$$\begin{aligned} d^{KL}(\mathcal{N}_1||\mathcal{N}_2) = \frac{1}{2}\log|\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{-1}|+ \\ \frac{1}{2}\mathbf{tr}(\boldsymbol{\Sigma}_2^{-1}((\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T+\boldsymbol{\Sigma}_1-\boldsymbol{\Sigma}_2)). \end{aligned} \tag{4.3}$$

By comparing sequences of clusters on the basis of this divergence[1] the data can be organised into temporal clusters.

Once labels are available however, a GMM may be used to describe each class. The KL divergence between a GMM of action p_{A_i} and p_{A_j}, that is, $\int p_{A_i}(x) \log(p_{A_i}(x)/p_{A_j}(x))\, dx$ can be estimated through Monte Carlo sampling and visualised with multi-dimensional scaling [2] as shown in Figure 4.4. As is clear from the graph, the relative divergences between the actions such as walking and sitting must be greater than between the similar postures walking and jogging. This would suggest the plausibility of an incrementally expanding model whereby the divergence between incoming data and the current set of actions is examined for the detection of unique motion.

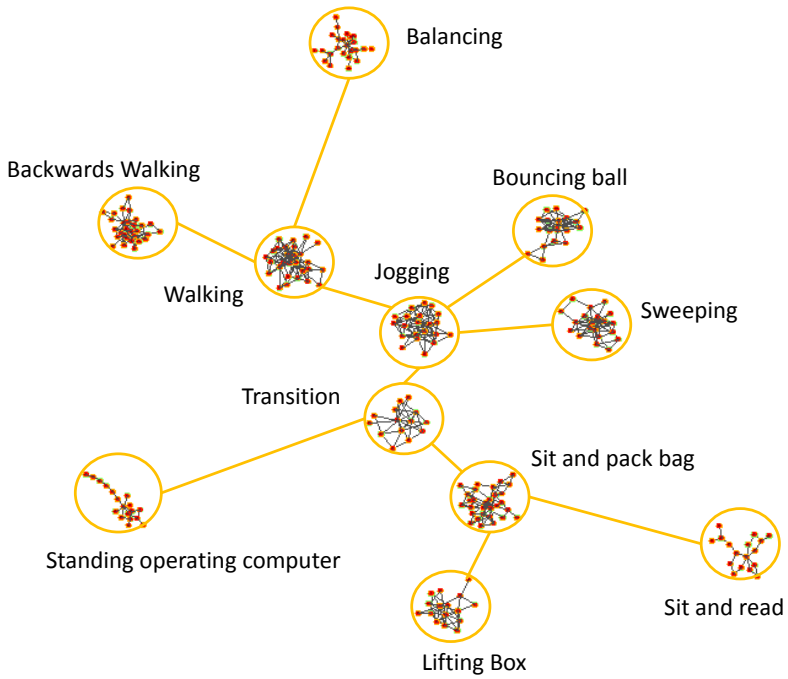

Figure 4.4: GMM_{MML} model showing Kullback-Leibler divergences between activities estimated via Monte Carlo sampling and shown using force-based graph layout [2].

Employing a complex model to describe each action is ultimately favourable for accurate recognition and also motion generation in the long term as definite event

[1] Other suitable distance measures include Euclidean distance or radial basis function.

labels become available. Although the actions are initially modelled in a single GMM, they become apparent as a pattern in the sequence of activated Gaussian distributions given by,

$$\tilde{\mathrm{z}}_n = \underset{k}{\operatorname{argmax}}\ p(\mathrm{z}_{nk}|\mathbf{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.4}$$

which indicates a particular action as demonstrated in Figure 4.5a where an excerpt of the data from two recordings of the 5^{th} subject are displayed. The following section provides an analysis of using temporal segmentations when dealing with unlabeled data, which results in clustering actions in an unsupervised learning approach.

It was assumed each cluster had a diagonal covariance matrix. The initial variance set to $\Sigma_0 = \begin{bmatrix} 3\sigma_\theta I & 0 \\ 0 & 3\sigma_p I \end{bmatrix}$ where σ is the mean of the feature standard deviations where the joint angle and velocity standard deviations were set separately.

4.3.2 Segmentations

Segmentation involves partitioning temporal data into subsequences by identifying modal changes. Human motion often can be characterized by sequential modes such as static poses, periodic motion or complex sequences utilizing several body parts in tandem. The data set, $\mathbf{X} = [\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_N]$, may be partitioned into M segments, $s = [s_1, s_2, \cdots, s_M]$ where s_m defines the starting frame of m^{th} segment and each subsequence is given by $\mathbf{X}_{(s_m, s_{m+1})} = [\mathrm{x}_{s_m}, \cdots, \mathrm{x}_{s_{m+1}-1}]$. The goal here is to closely align these switching points to those typically chosen by hand-crafted labelling.

Using the maximum *a posteriori* cluster sequence, $\tilde{z}$ (Figure 4.5a), a proposed distance measure, ϕ, is composed of a linear combination of KL divergence between two adjacent time windows of length w. The weights are chosen to increase the

influence of divergences closer in time, resulting in

$$\phi(t) = \sum_{j=t-w}^{t-1} \sum_{i=t}^{t+w-1} \psi_{z_{t-j}, z_{t+i}}(2w - i + j) \tag{4.5}$$

where $\psi = d^{KL}$ is a similarity matrix which has been chosen as the KL divergence Eq. (4.3). When expressed in matrix form, this equates to

$$\phi(t) = \mathbf{tr}((W + \lambda W^T)\Psi) = \mathbf{tr}(\tilde{W}\Psi) \tag{4.6}$$

where W is a $\mathbb{R}^{w \times w}$ matrix of weights defined by $W_{ij} = w - j + i$. λ influences the relative scaling of the divergences based on the time difference and is constrained to $-1 < \lambda < 1$, while $\tilde{W}$ is also constrained to be non-negative (element-wise) for $\lambda < 0$.

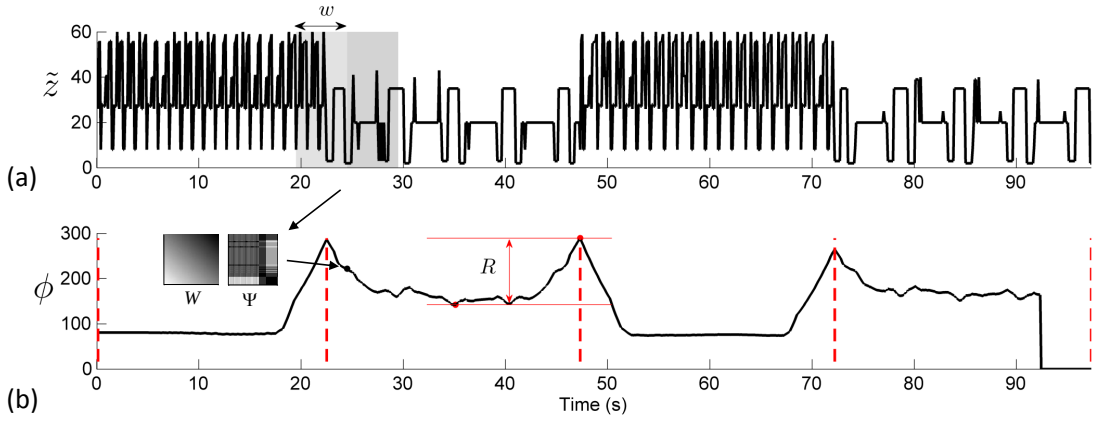

Figure 4.5: (a) Sequence of predicted GMM distributions for an excerpt of two consecutive recordings from subject 9 of the walking and lifting actions (where walking occurs first and lifting second). Two shaded regions indicate the adjacent time frames compared in Eq. (4.5) and Eq. (4.6). (b) Segmentation distance measure, ϕ, with peaks indicating change-points shown by red-lines.

Peaks in ϕ relate to a locally significant change in the set of clusters describing the data. Segment boundaries were chosen when the difference between the current segment minimum and the next peak is greater than a threshold R, which equates to $s_{m+1} = \{t \mid R < \phi_t - \min(\phi_{[s_m,t]}), \phi_{t-1} < \phi_t, \phi_{t+1} < \phi_t\}$. As depicted in Figure 4.5b,

the threshold can be varied to detect transitions between less pronounced behaviors, especially with a reduced setting of w. The segmentation approach utilizing the sequence of GMM clusters will be referred to here as S_{GMM}.

Similar segmentations can be achieved without relying on the GMM. Two other approaches investigated were the use of a body momentum signal and tracking zero-velocity-crossings. To detect changes of momentum, the features were linearly combined with a weighting proportional to the associated mass approximated from biomechanics literature [3], resulting in the feature $\hat{\mathrm{p}} = \sum_i m_i \|\dot{\mathrm{p}}_i\|^2$, for all body segments, which is an energy-like signal referred to here as S_E. Another alternative was to form an aggregate of the zero velocity crossings within a specified window w_{zc}, which is similar to the technique used in [45], and will be referenced as S_{ZC}.

4.3.3 Segment Similarity

Given a set of segments, s, each of which are represented by a sequence of clusters $[\tilde{\mathrm{z}}(s_m), \tilde{\mathrm{z}}(s_m+1), ..., \tilde{\mathrm{z}}(s_{m+1}-1)]$, the similarity measures between segments may be used to form a temporal clustering. The sequences often vary in length and speed affecting common similarity calculations based on Euclidean distance or dynamic programming [122], especially for periodic motion. In [122] the strategy was to minimize the distance calculations over a range of time scale factors of each sequence. Alternatively for this new approach, a kernel matrix is constructed based upon the KL divergence between each data point, $\mathcal{K}_{ij} = D^{KL}_{z_i,z_j}$, as shown in Figure 4.6a.[2] Within each segment a discrete Fourier transform is used on the columns (or rows) of this matrix to determine a significant frequency, if one exists, to partition the segment into subsequences. Figure 4.6b shows the segment frequency spectra of example instances of actions A_1 and A_9. To account for slight variations in frequency over the segment,

[2] D^{KL} was formed from d^{KL} by scaling all divergences down and taking the element-wise reciprocal to form a cluster similarity measure on the interval [0,1].

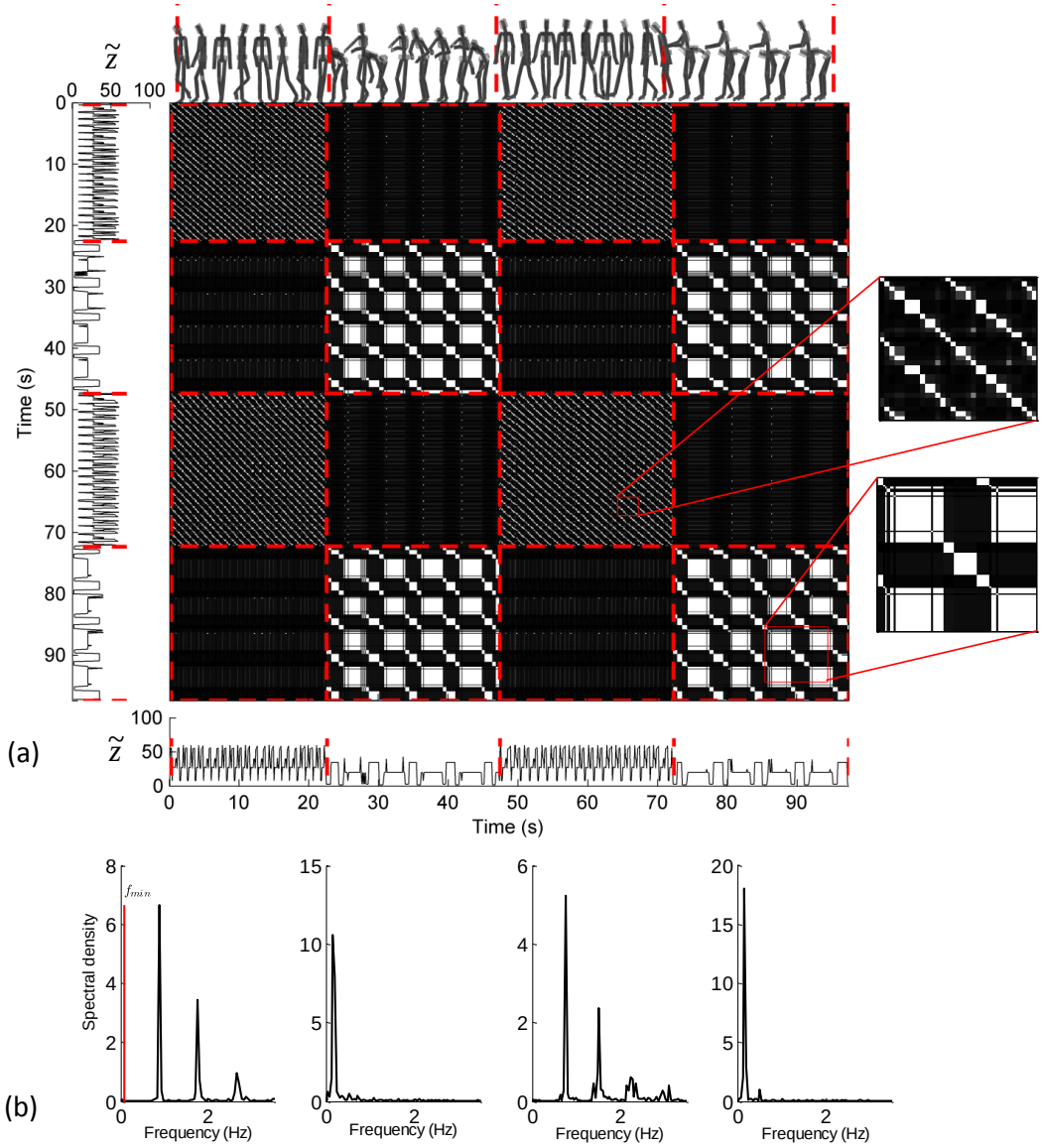

Figure 4.6: (a) Kernel matrix (scaled KL-divergence [0,1]) from repeated actions A_1 and A_9 from S_9 where segments are indicated by red dashed lines. The actions are visualised with sampled postures along the top of the Figure. (b) Spectral analysis for each segment for partitioning into subsequences.

one can partition the segment based on the activation of a particular cluster with the highest spectral density and when the resulting sequence length was within a small bound on the period. If there was no significant frequency component the segments were split into blocks of 50 samples (equivalent to 5 seconds). This process thereby

permits a sequence indicator matrix $T \in \{0,1\}^{N \times \tilde{M}}$ which consists of block diagonal components $T_m \in \{0,1\}^{l \times \tilde{m}}$, where $l = s_{m+1} - s_m$ and $\tilde{m}$ is the number of sequences constituting the m^{th} segment.

The subsequent partitions for the data in Figure 4.6 are displayed in Figure 4.7a. Having split the data into sequences the DTW distance is computed between each pair of sequences where Φ_{ij} is the distance between sequence i and sequence j. As an example in Figure 4.7b the distance is computed by the minimum cost path through this block of the kernel matrix, in this case from two examples of a walking cycle. The cost is calculated with the DTAK function detailed in Section Section 3.2.2. When the similarity is expanded out on per sample basis, Figure 4.7c, the degree of similarity between segment i and segment j can be evaluated by the proportion of similar sequences through the relation $\tilde{\Phi}_{ij} = \mathbf{tr}(\Phi_{ij}\mathbf{1}_{l_j \times l_i})$. Applying a threshold to this segment kernel ($\tilde{\Phi}_{ij} > \beta$) controls the confidence with which one assigns the same label to segments i and j. Figure 4.7d compares the result of setting $\beta = 0.5$ against the ideal similarity matrix based on the hand-crafted labels where incorrect (dis)similarities are depicted in red.

In order to reduce computational overheads the distance for each sequence, z is instead compared to an expanding set of unique sequences q stored in Q. The result is a label, y, assigned to each data point in each segment. For the relatively simple example involving A_1 and A_9 the label corresponds clearly to the identity of the action, Figure 4.7e. This algorithm is summarized in Algorithm 1.

Supervised Mixture Models

Temporal segment clustering provides a useful starting point for interpreting incoming motion data. But as labelled examples of a class become available it is important to incorporate this information to refine the recognition process. This can be achieved

Algorithm 1: Temporal segment clustering

Input: X, w, R, λ, β, α

Output: s, $\mathbf{y}$

Initialize: $Q = \emptyset$, $\Phi = \emptyset$, $T = \emptyset$, $M = 1$

$\{\mu, \Sigma, \pi, \tilde{D}\} \leftarrow \mathrm{GMM}_{\mathrm{MML}}(\mathrm{X}, \alpha)$

$\tilde{\mathrm{z}}_t = \arg\max_k \; p(\mathrm{z}_{tk}|\mathrm{x}_t, \mu_k, \Sigma_k); \quad \forall t$

$\mathcal{K}_{ij} = \tilde{D}_{z_i, z_j}; \quad \forall i, j \in 1 \ldots N$

$\phi_t = \mathbf{tr}((W + \lambda W^T)\Psi_t); \quad \forall t \in \{w < t < N - w\}$

for $t = 1 \ldots N$ **do**

  if $R < (\phi_t - \min(\phi_{[s_M, t]}))$ & $(\phi_{t-1} < \phi_t)$ & $(\phi_{t+1} < \phi_t)$ **then**

    $s_{M+1} = t$

    $T_M \leftarrow DFT(\mathcal{K}_{[s_M, s_{M+1}][s_M, s_{M+1}]})$ // $T_M \in \{0,1\}^{l \times \tilde{M}}$, $(l = s_{M+1} - s_M)$

    $T = \begin{bmatrix} T & 0 \\ 0 & T_M \end{bmatrix}$ //block diagonal segment indicator.

    $M = M + 1$ // increment number of segments.

  end

end

foreach *column* m *of* T **do**

  foreach *sequence* q *in set* Q **do**

    $\tilde{\mathcal{K}}_{ij} = \tilde{D}_{q_i, z_j}; \forall i \in 1 \ldots l_q, \forall j \in \{t | T_m > 0\}$

    $\hat{\phi}_q = DTW(\tilde{\mathcal{K}})$

  end

  if $\hat{\phi} \prec \mathbf{1}$ **then**

    $Q \leftarrow z_j$ // add sequence to Q.

    $\hat{q} = \hat{q} + 1$ // increment size of Q.

    $\gamma_m = \hat{q}$ // index of latest sequence.

  else

    $\gamma_m = \arg\max_q \; \hat{\phi}_q$ // index of nearest sequence in Q.

  end

  $\Phi = \begin{bmatrix} \Phi & T_{[1,m]} \hat{\phi}_\gamma \end{bmatrix}$ // use columns $[1, m]$ to express Φ per sample.

end

$\tilde{\Phi}_{ij} = \mathbf{tr}(\Phi_{ij} \mathbf{1}_{l_j \times l_i}) \qquad \forall i, j = 1 \ldots M$

$\mathbf{y} \leftarrow$ Combine segments i and j for $\tilde{\Phi}_{ij} > \beta$

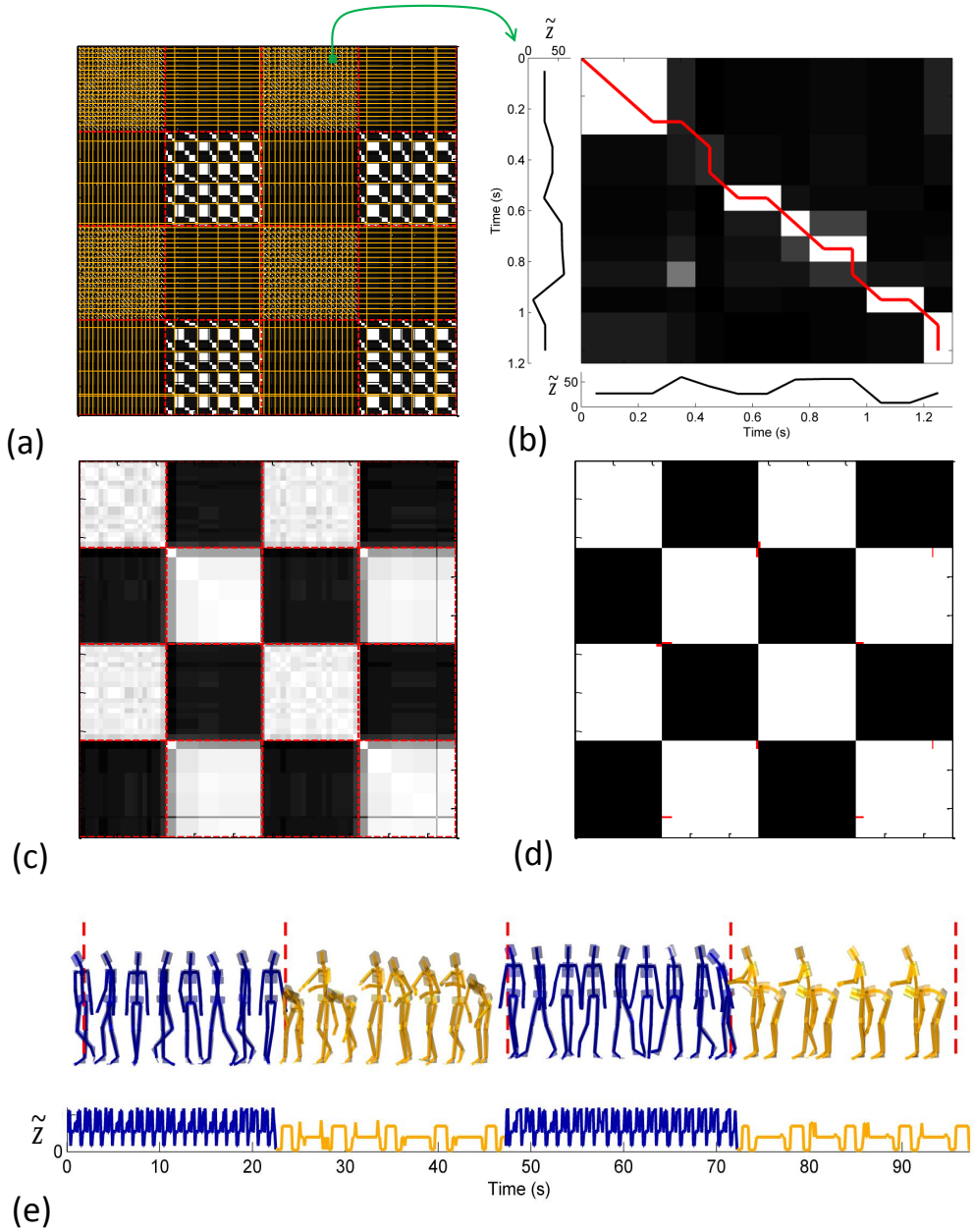

Figure 4.7: (a) Kernel matrix displaying the frequency partitioning of the segments (orange) from the same data in Figure 4.6. (b) Example of the computed DTW path for a pair of sequences (indicated in green in (a)). (c) Segment similarity matrix from computing all DTW distances where the gray scale indicates the level of similarity. (d) Comparing the true segment equality (based on labels) with the result of an applied threshold β to Φ, errors are indicated by red squares. (e) Motion capture visualisation and GMM sequence where the labels assigned by the algorithm to the segments are indicated in colour. The two classes are walking (A_1) and lifting a box (A_9)

by forming a class specific model, Θ_c, composed of the set of Gaussian distributions of maximum posterior probability for data points belonging to class c, that is, the set $\{k|p(z_{nk}|x_n) > p(z_{nj}|x_n)\}$, where $k \neq j$ and $\{n|Y_{nc} = 1\}$. The prior probabilities, π_k, must then be calculated for each GMM.

The likelihood of a new data point belonging to a particular GMM $p(x|\Theta_c) = \sum_k p(x|\mu_k, \Sigma_k, \pi_k)$ can thereby indicate an appropriate classification. However, classifying points in isolation would fail to recognize some modes involving sequences of postures. Classifying sequences of postures or segments can be performed by accumulating a log-likelihood term over the the data points in a proposed segment. Alternatively, if a set of unique sequences, Q, as assembled in Algorithm 1, is assigned class membership, new sequences may be recognized by computing the minimum DTW distance to the sequences in set Q. From initial testing using the DTW distance attained similar results to the likelihood approach but was significantly less efficient. Due to the large number of validation tests conducted the supervised classifications were obtained with likelihoods.

4.4 Experimental Results

In this section the results obtained from a classification of the data using a set of evaluation constraints is reported. Initially, the evaluation measures are described before reporting results from unsupervised clustering and supervised recognition cases.

4.4.1 Unsupervised Segment Recognition

In this section temporal segment clustering is evaluated over the entire data set of 14 recordings and key parameters are explained using a subset of the data. The evaluation of the segmentation result was divided into a number of metrics.

For this problem the number of temporal clusters is not defined in advance and

is bounded by the number of detected segments. The evaluation of the accuracy of the label assigned to a segment is achieved by maximizing the correspondence to the ground truth labels. Constructing a confusion matrix, $\mathcal{C}$, where $\mathcal{C}_{ij}$ is the number of data points in class i that belong to temporal cluster j permits the use of the Hungarian algorithm [123] to compute a permutation of the algorithm labels that minimizes the total error or equivalently the off-diagonal entries of $\mathcal{C}$. One can accordingly compute the accuracy with the expression $\frac{\sum \mathbf{diag}(\mathcal{C})}{\mathbf{tr}(\mathcal{C}\mathbf{1}_{c\times\tilde{c}})}$.

Segmentation points

Firstly the segment change point times were compared against the changes which occur in the ground truth labels, Y. The mean minimum time difference between these two sets of segment points was calculated as,

$$\Delta t_{s_{min}} = \frac{\sum_m^M |\min(s_m - \hat{s})| + \sum_m^{\hat{M}} |\min(\hat{s}_m - s)|}{M + \hat{M}}, \tag{4.7}$$

where $\hat{s}$ and $\hat{M}$ describe the segment times and total number of segments derived from the labels respectively.

The correspondence between the segmentations provided by the proposed methodology and the change points derived from the labels was calculated by Eqn. (4.7). Figure 4.8 shows this measure for 3 approaches suggested in Section 4.3.2 when compared to the PPCA method [36].

The energy based momentum segments, S_E, and mixture model segments, S_{GMM}, were most closely aligned with the labels. The main difference between these sets of segments was due to a pair of transitions between actions resulting in a single segment point. Increasing the sensitivity of the segmentation to include the transitions, however, often resulted in additional segment points within more complicated actions.

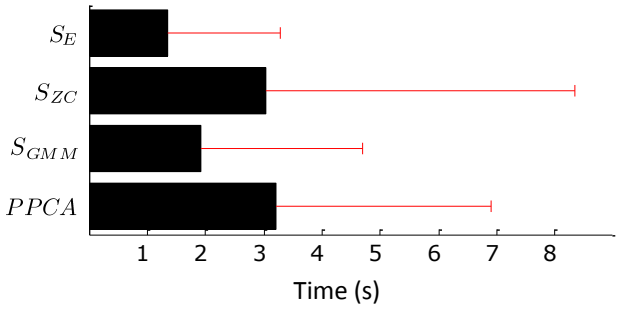

Figure 4.8: Bar graph of the average time deviation between segment locations and the hand-crafted boundaries. Error bar indicates a single standard deviation.

Recognition Parameters

Since the goal here is to address the challenging problem of clustering equivalent actions across different motion capture recordings, two data sets are considered at a time by concatenating the sequences of clusters, $\tilde{z}$, and segments s associated with each data set. Initially, in order to test parameters, a subset of the data was assembled comprising of actions $A_{2,8,1,4,9,5}$ from 6 randomly selected recordings. This allowed an examination of the effect of varying the number of GMM clusters in the model, K, controlled in the MML search algorithm by the variance parameter vector α, and the decision threshold, β.

Figure 4.9a summarizes the error rate of the temporal segment clustering over a number of selections of K. The error from models with K ranging from 5 to 100 indicates a superior performance is obtained for model sizes in this work of approximately 20 clusters. Clearly this analysis is not possible in a practical setting therefore one requires a method to estimate appropriate model size for different data sets. Employing an MML search to select the model size is predicated on a variance parameter per feature which was selected to be proportional to the maximum independent standard deviation of the features, σ_{max}, scaled by a factor α. Varying this factor influences

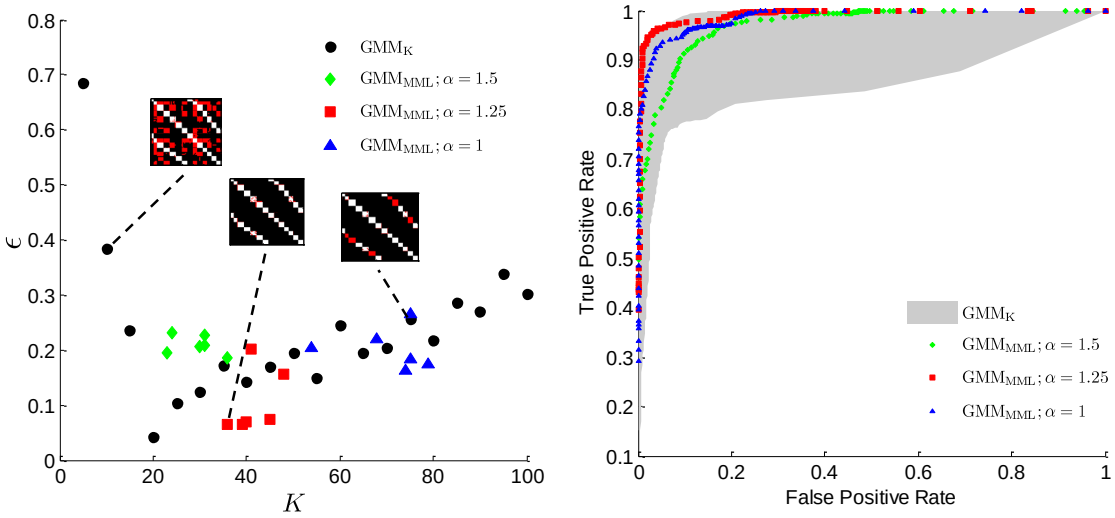

Figure 4.9: (a) Classification error rate (ϵ) over a range of target K for example data set $A_{2,8,4,1,9,5}$ and target α values for MML search. (b) The rate of correctly merged sequences versus rate of incorrectly merged sequence as β is varied in interval [0,1]. The gray region is the extent with which the rates vary with a specific target model size.

the resulting K as also depicted in Figure 4.9a where several models were trained for values $\alpha = \{1, 1.25, 1.5\}$. The choice of α controls the granularity of the model where K will increase in the advent of new observations. Setting α to 1.25 typically resulted in a lower error which suggests that these models are able to generalize the observed data and was used to train models further in this paper.

Additionally, the decision threshold β controls the sensitivity to the similarity of sequences. Figure 4.9b shows the accuracy as β is varied in the interval [0,1]. The true/false positive rates are expressed as proportion of correctly/incorrectly merged sequences. This decision threshold is applied after the computationally intensive similarity calculations, and could be tuned to optimize the performance.

Results

For the reduced problem involving a subset of classes and subjects the accuracy reached 93.6% and Figure 4.1 displays an example section of the outcome from this

algorithm.

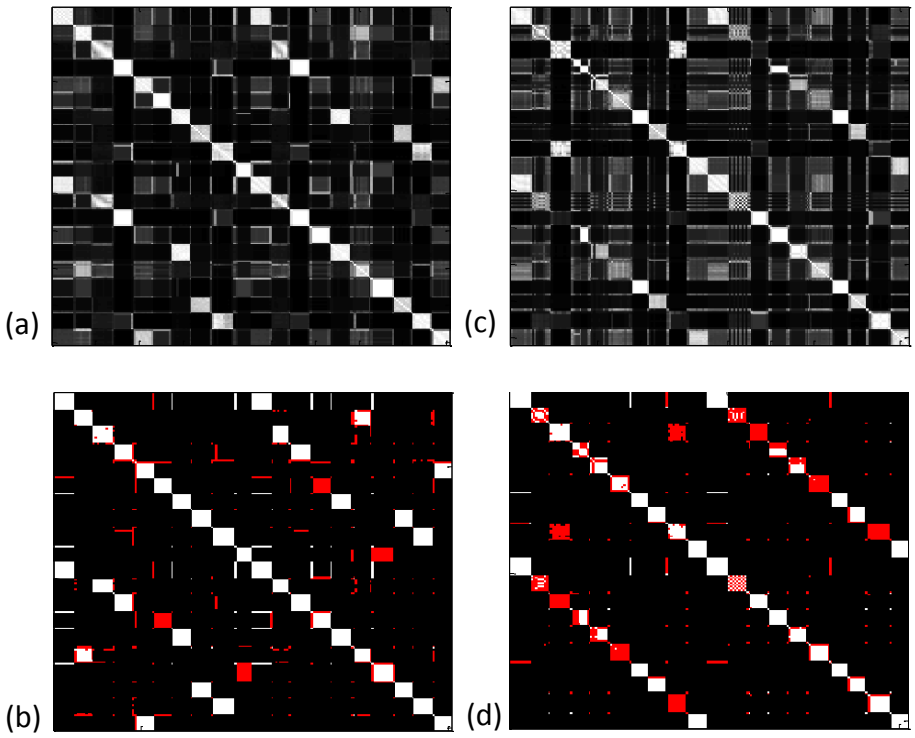

Figure 4.10: (a) Segment similarity matrix comparing repeated actions from S_5. (b) Result from applying threshold β to similarity for merging segments, errors are indicated by red squares. (c) Segment similarity matrix comparing S_5 and S_{10}. (d) Result from applying β for merging segments.

As the amount of data considered is extended it becomes increasingly difficult to cluster actions in this way. Segmenting two complete recordings (ten actions each) on average results in 67% accuracy for a fixed decision threshold β or 71.6% if β is varied. It was 76% when comparing repeated recordings from the same individual. To appreciate how the errors arise Figure 4.10 displays the result from two example cases. The self-similarity kernel is shown in Figure 4.10a for two concatenated recordings of Subject 9 and Figure 4.10b the relationship between the sequences where erroneous (dis)similarities are depicted in red. Some actions are missed because a different execution of an action results in a unique set of clusters describing each iteration. Similar errors occur for Subject 2b and 5b shown in Figures 4.10c and Figures 4.10d. In fact, in this case half a segment is not merged in Figure 4.10(d) because in the

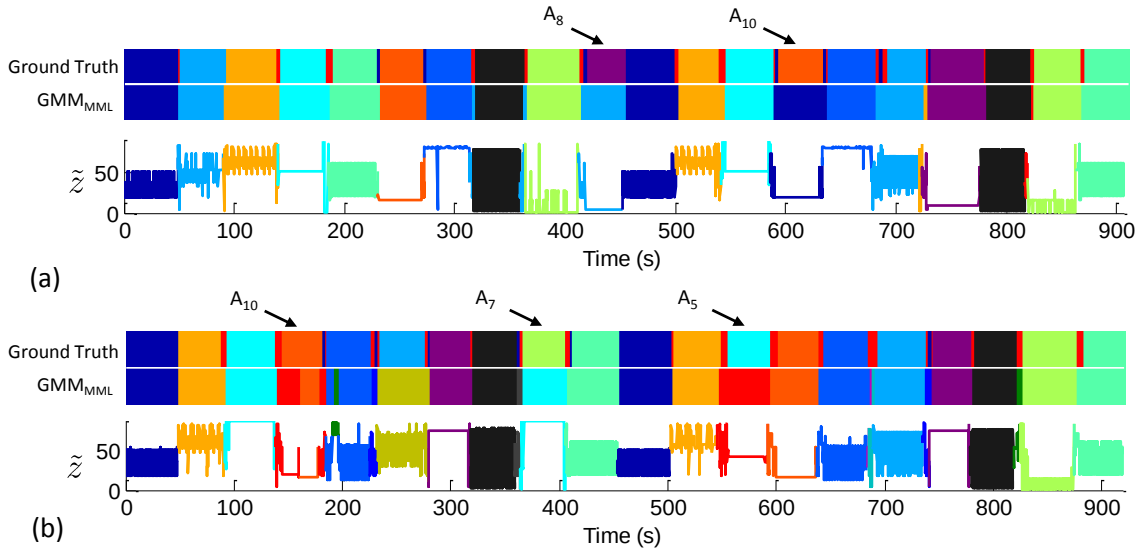

Figure 4.11: (a) The result from combining segments based upon threshold β for the recordings by Subject 9 where erroneous labels are (b) The result from applying β when segmenting data from Subject 2 and Subject 5.

balancing task, A_{10}, it was subsequently discovered that Subject 2 was switching the supporting foot half way through session.

Indeed, this suggests the ground truth labels are relating significantly different postures to describe an action because actions which are nominally equivalent are frequently executed differently due to the unconstrained nature of the experiment. These variations can be accommodated when one has multiple training examples to form a classifier. The classifier may then recognize a collection of postures as belonging to the same class. Errors arising from different postures are designated as new classes, however, false positive errors are harder to recover from and problematic if the organized data were to be subsequently used to form a supervised classifier.

4.4.2 Supervised Classification

With the inclusion of example labels the improvements in recognition performance can be assessed and compared to other standard algorithms.

Evaluation measures

When investigating a supervised recognition problem, performing a systematic cross-validation ensures reliability of the recognition, independent of which subject is tested. The validations adopted here are based on composing subject adapted (SA) and leave-one-out (L1O) models [35]. A subject adapted model validation involves training a recognition algorithm with data from all subjects leaving out one repetition recorded by one subject. This simulates recognition performance for a returning user interacting in a new session. Each repetition for a participant is left out at a time, while training a model from the rest of the data. The leave-one-out validation involves training the algorithm with data from all but one subject and predicting the class of data from the subject left out of the modeling process. All accuracy percentages are averaged over all permutations of leaving one data set aside.

In many applications the classes of interest appear for only a small fraction of the time. This is clearly the case for daily activity monitoring where there may be periods of waiting or rest in which there are movements that have less meaning. To simulate these circumstances five of the classes were selected as classes of interest while leaving the rest to be re-labeled as the null class.

In order to show the value of clustering, the results from supervised recognition are compared with discriminative models which are regularly used in this context, k-NN and SVM. For the implementation of k-NN a range of values for k were tested as shown in Figure 4.12a. When leaving one participant out, the optimal k was found to be 10, which was the setting used in the remainder of the tests. The SVM was tested with the multi-class function of the LIBSVM toolbox for MATLAB [100]. The radial basis function kernel, $\mathcal{K}(\mathrm{x}_m, \mathrm{x}_n) = \exp(-\gamma|\mathrm{x}_m - \mathrm{x}_n|^2)$, was used and is parameterized by γ. A logarithmic grid search was carried out in an attempt to find optimal settings for γ and the misclassification slack variable, C, where the result is shown in Figure

4.12b. The parameters were finally set as C=0.095 and γ=0.25. A further comparison was conducted with a GMM composed of a fixed number of clusters of $K = 10$ per class, which will be referred to as GMM_{10}.

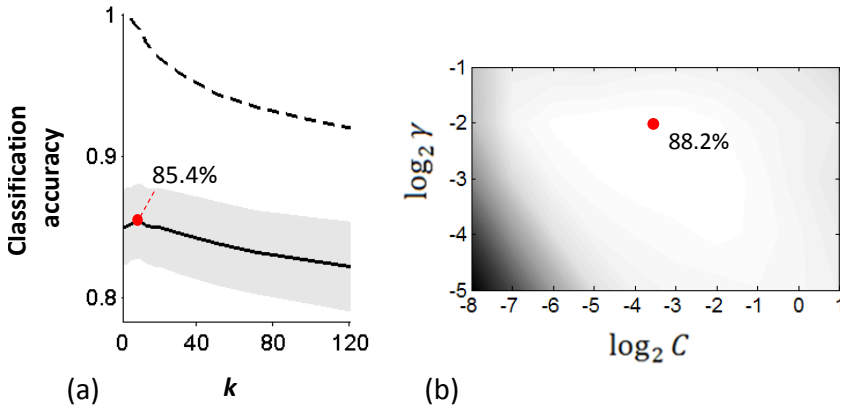

Figure 4.12: (a) Selection of k in k-NN. Solid line is testing accuracy, dashed line is training accuracy and gray region is 1σ of test accuracy across 10 subjects. (b) Selection of SVM parameters C and γ. Gradient indicates test accuracy.

Finally, there are different schemes for annotating blocks of time series data. A majority vote could be computed for the most common label in one block of data where each block may consist of a single frame, a pre-determined window of fixed length, a set of segments derived from the approaches in Section 4.3.2, or bounded by the ground truth labels. Realistically only the first three cases can be used in practice and the most successful scheme is used to compare the classifiers in the results section.

Results

The test accuracy for each algorithm with each cross-validation method is given in Table 4.1. $\text{L1O}_{0.5}$ refers to the test where half of the classes are re-labelled as null and consisted of the actions $A_{(2,4,6,7,10)}$. k-NN and SVM were tested on a frame-by-frame basis but the GMM was tested on each segment detected with S_E and assignment of the maximum log-likelihood of the time-series segment per model.

This result shows that these recognition methods are relatively robust to cases where a significant amount of irrelevant data is gathered since each accuracy is typically above 90%. In addition, leaving-out recordings from returning participants resulted in a slightly higher accuracy (in the order of 2%). The SVM was most effective by a small margin over GMM_{MML} for the reduced set of classes but GMM_{MML} gave the highest accuracy with the entire set of classes with 89% in the L1O test. Figure 4.13a displays a fraction of the classification result with half the classes relabelled. Another excerpt of the accuracy including all classes is shown in Figure 4.13b. The ground truth label (top) is compared to the recognition for each method. In Figure 4.13a GMM_{10} has several false negatives whereas for 4.13b most errors in each method occur as substitution errors. For approximately 30 minutes of data as shown in 4.13b only 2 substituted events occur for GMM_{MML}.

Table 4.1: Classification accuracy for validations in percent.

Algorithm	$SA_{0.5}$ (%)	$L1O_{0.5}$ (%)	SA (%)	L1O (%)
k-NN	92.3	90.9	88.4	85.4
SVM	**93.5**	**92.2**	90.2	88.2
GMM_{MML}	93.2	91.5	**91.5**	**89.4**
GMM_{10}	91.4	87.6	89.5	86.1

For each algorithm, the user adapted validation had a higher accuracy than the L1O test. On average, the difference was approximately 1-4% which indicated that previous observation of an individual did not significantly increase the accuracy of recognition for this data set. This may be due to the few repetitions that were gathered or due to differences that are not accounted for in the motion between recording sessions. Each action was not constrained to be performed in specific postures. A minor difference in posture in a new recording is likely since they were repeated weeks apart.

To analyse the recognition results in greater depth, the accuracy in recognizing

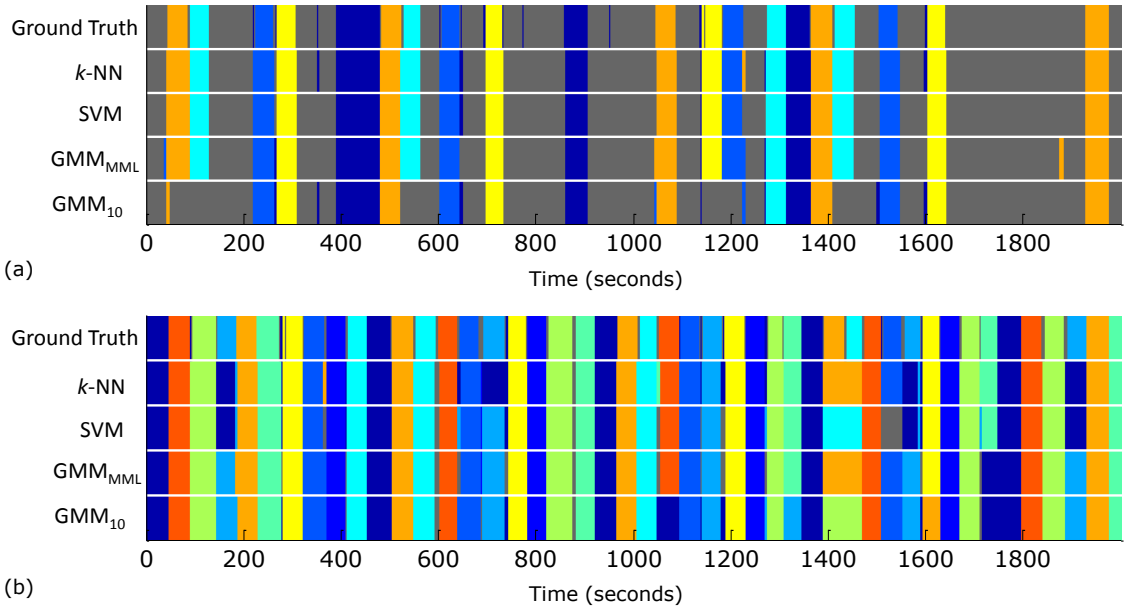

Figure 4.13: (a) Excerpt from the leave-one-out classification output as compared with the manually assigned labels (top) with half the classes re-labeled as null. Particular colors refer to the class while the gray areas indicate the null class. (b) Excerpt from leave-one-out classification with all target classes included.

particular actions is shown in detail in Figure 4.14. This displays the source of errors divided into actions and participants. On the left, the test accuracy is decomposed into recordings per action where darker pixels indicate higher error. On the right is a confusion matrix of the actions whereby darker entries indicate to which class a high proportion of data was assigned. The data labelled as transitions, A_{11}, were clearly the most difficult to recognize especially when they were short events not detected by the segmentations. The error is high even when classifying frame-by-frame with an SVM which is to be expected due to wide variation of postures in this class. Common misclassifications among the classifiers were from A_5 (sitting action) A_6 (backwards walking) for S_{7a} and A_8 for S_{9b}.

A comparison of mixture model recognition accuracy is presented in Table 4.2 between using fixed-width segments, the segments generated using S_E and the segments indicated by the original labels. This result shows that in many cases the segments

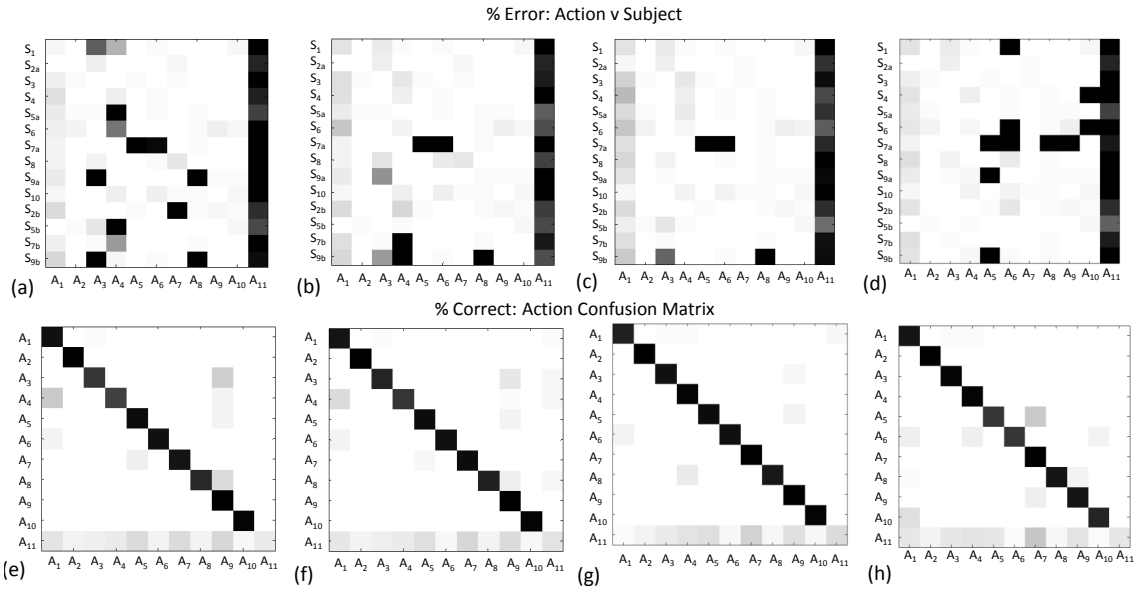

Figure 4.14: (a)-(d) Error percentage of subjects (S_{1-10}) and actions (A_{1-11}) (defined in Table 1) for k-NN, SVM, GMM-MML and GMM-10 respectively where the darker pixels indicate higher error. (e)-(h) Confusion matrix between actions where darker pixels indicate where a high proportion of data was classified. For instance a value in row 3, column 9 says data which should have been classified as sweeping (A_3) was classified as lifting (A_9).

generated increased the accuracy of detection over simply separating the data into equal width segments. It is therefore reasonable to assume that as the change points become more accurate the recognition accuracy approaches that of having known the ground truth segment locations.

4.5 Discussion

The recognition of actions as presented in this work bears relevance for several applications.

Table 4.2: Comparison between segmentations and fixed window recognition.

	Algorithm	Frame (%)	Window (%)	Segments (%)	Labels (%)
SA	k-NN	88.4	88.6	75.7	92.8
	SVM	**90.2**	**90.6**	77.0	95.8
	GMM_{MML}	87.0	83.2	**91.5**	**99.2**
	GMM_{10}	89.0	81.7	89.5	99.1
L1O	k-NN	85.4	86.0	85.3	92.9
	SVM	**88.2**	**88.8**	86.9	94.3
	GMM_{MML}	84.8	75.7	**89.4**	**96.6**
	GMM_{10}	85.3	75.0	86.1	95.6

4.5.1 Sensor Placement

For applications in tracking daily activities a comprehensive suite of sensors is cumbersome and currently impractical to wear for long periods of time. Extensive research has been performed in recognition from either a single accelerometer or a smaller set of strategically placed sensors [115]. Since, in this work, motion was recorded across the entire body the sensitivity of the recognition algorithms to the combination of sensors can be tested. Table 4.3 shows the recognition accuracy when using data from selected sensor configurations. This is indicated by the set of reference frames, labelled in Figure 3.3b, that are included in the data. These frames correspond to the sensors located at the pelvis ({1}), sternum ({5}), head ({7}), upper arms ({9,13}) and thighs ({16,20}).

Table 4.3: Classification accuracy for a range of sensor configuration under the L1O validation.

Algorithm	{1} (%)	{1,5} (%)	{1,5,7} (%)	{1,5,7,9,13} {16,20}(%)	{1-23} (%)
k-NN	66.5	69.6	77.3	82.7	85.3
SVM	55.5	65.8	73.6	83.1	86.9
GMM_{MML}	68.7	**76.3**	**78.2**	**84.6**	**89.4**
GMM_{10}	**69.2**	71.2	75.5	75.9	86.1

4.5.2 Relevance of results for applications

The unsupervised event detection method achieves a 72% correspondence with the labels. The majority of the differences are due to classifying the events as unique rather than a false matching of postures. This is useful for partitioning the data and matching clearly equivalent events but requires feedback for matching different styles of the same class. For more constrained problems, however, the unsupervised temporal clustering performs to a higher standard. With the example pair of classes $A_{1,9}$ the events were often matched perfectly and for $A_{2,8,1,9,4,5}$ the average accuracy was 94%. In a constrained action monitoring context $A_{1,9}$ (walking and lifting) may, in some businesses, constitute a significant proportion of a working day and the ability to reliably detect these actions across multiple workers would help to manage the work load and health of the staff. Temporal segmentations are beneficial in reducing the cost of labelling in this domain. Aside from off-line segmentation, an on-line 'experience sampling' system [34] would also be less disruptive by sampling at meaningful change points discovered through segmentation.

With an existing database the algorithm consistently achieved an average recognition rate of over 89% for an unobserved user despite the lack of constraints over the execution of the actions. In the case of long-term activity monitoring the data sets may span days and important actions are often sparsely located within these time frames. The method considered here reaches over 91% accuracy for detecting sparse actions albeit in a significantly reduced data set.

In a robot programming by demonstration (PbD) application the clustering methods described are particularly useful. The models can recognise gestures and, with appropriate generation procedures, reproduce similar motion trajectories for deployment in path planning. Since no state transition information is stored within a GMM the velocity of the motion may be used to model the dynamics [43, 44]. A potential

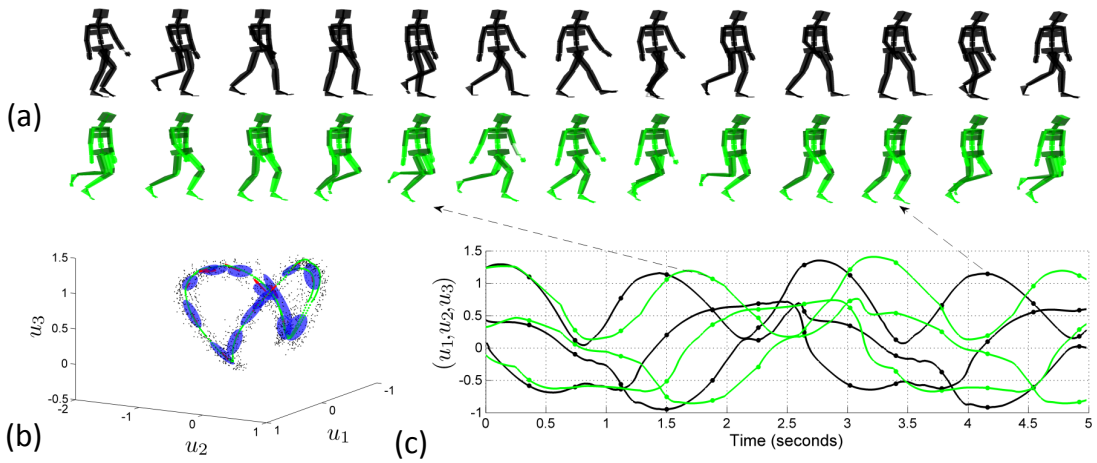

Figure 4.15: Motion generation from a Gaussian mixture model of two actions, A_1. (a) Sequence of posture visualizations from the simulation (left) and the data (right). (b) GMM structure along the three largest eigenvectors (u_1,u_2,u_3) where each distribution is an ellipsoid, the data is indicated in black and the simulation path is in green. (c) Graph of the data projected onto three eigenvectors (black) against the simulation result in the same eigenspace (green).

application of PbD motion sampling is for programming robotic surgical assistants to perform subtasks [124] which could be achieved here by tracking the positions and velocities for relevant end effectors throughout a demonstration of the task. One example of motion reproduction is shown in Figure 4.15, where a walking cycle is generated from the principal eigenvectors.

Another consideration for these applications is the ability of a model to incrementally learn to account for new actions. With each action modelled as a set of multivariate Gaussian distributions, the relative KL divergences were displayed in Figure 4.4. This shows which actions may be misclassified and bears some correlation to the confusion matrices in Figure 4.14. If a new data set is observed the KL divergence may be used to quantify the similarity with previously observed actions. Given particular thresholds, a new action may have been observed and thus added to the model [41].

4.5.3 Limitations and future improvements

For an incremental model or an unsupervised segmentation (Section 4.3.2), there would, in practice, be a point at which the recognition imperative overrides the necessity to learn new actions. The set of clusters employed to describe a particular action would form an action model as used in Section 4.4.

It was assumed here that the data is adequately described by a uni-dimensional label, however, multiple simultaneous events may be observed on subsets of the features. In this work, the dominant actions maintained precedence over other concurrent actions although in special cases it may be important to make this distinction. It is also unclear as to the consistency in attaching labels to actions. Different observers may apply the labels in a different manner so for a more accurate ground truth for comparison one would report the variation in multiple observers.

The mixture model methods can be improved by proposing more complex hierarchical models with less structural limitations. For example, instead of the assumption of one latent variable controlling the distribution at any one time it may be a combination of latent variables. Learning the structure of the hidden interactions is a challenging problem which may be handled by non-parametric methods [39]. But the disadvantage is the complexity of these models and consequently the amount of computation and data required to estimate the parameters.

4.6 Chapter Summary

In this chapter an algorithm and experimental validation was presented for an autonomous segmentation and recognition of human motion. A series of actions were recorded for multiple subjects and then a mixture model trained with these actions informed a segmentation procedure subject to the variation in component assignment

locally in time. An experimental evaluation is conducted for an unsupervised case where the actions are automatically segmented and categorised, and for a supervised case where we test with a leave-one-out cross validation. Moreover, the recognition performance was tested with previously observed subject motion and with a reduced set of actions for the case where significant amounts of irrelevant data are captured. This allowed for an evaluation of the system performance under the conditions likely to arise in practice.

Chapter 5

Robot programming by demonstration through dynamic motion primitive models

5.1 Introduction

The ability to learn motion skills is of increasing importance in the field of robot control. As the intended role of a robotic assistant expands, learning capabilities will be required for the robot to achieve a variety of unforeseen tasks. Programming by demonstration or imitation learning provides a simple mechanism to endow a robot with a new capability. The simplicity lies in the fact that a teacher or demonstrator is only required to show the robot the desired task as opposed to developing control software for each specific task. This can be achieved by manually guiding the robot through the motion or by the robot observing the motion of a demonstrator and translating the motion to the robot context.

Research into programming by demonstration models has concentrated on representing the set of demonstrations as derived from a collection of discrete units,

referred to as motion primitives. Partitioning the motion in this way enables the deployment of incrementally expanding models allowing adaptability to new observations as they arrive. This approach also enables flexibility for the robot to employ subsets of motion primitives to achieve a trajectory depending on the current task requirements. Furthermore, sequences of these discrete components are conducive to symbolic logic analyses for higher cognitive tasks such as planning and intention.

The approach undertaken here is to assume that the demonstrated paths are trajectories from a latent non-linear dynamical system parameterised by the set of motion primitive components. This should be distinguished from referring to a set of trajectories as a motion primitive, which is derived from another approach known as Dynamic Motion Primitives (DMP) [125]. As was established in the previous chapter, mixtures of Gaussian distributions can form a representational model of human motion which achieves a relatively high accuracy in gesture recognition. Each Gaussian distribution in the GMM is then referred to as a motion primitive. Given a desired sequence of motion primitives each activated over a specified time interval, it is possible to generate highly similar trajectories to the observed motion. The difficulty associated in the application of motion primitives in robots is specifically how this motion is reconstructed in a stable and reliable manner and how to incorporate contextual information to modify the desired motion.

It should be emphasized that the purpose of the motion primitive model is to represent intended trajectories. It is assumed that a control algorithm is already available to execute any arbitrary target trajectory. The primitives therefore provide a mechanism for summarising the set of desired paths. As shown in Figure 5.1 the motion primitive model generates the sequence of commands as learned from a set of demonstrations. However, one must still rely upon kinematic functions defining the robot structure and a control scheme for the robot dynamics in order to achieve the

desired trajectory.

The approach adopted here is based on the dynamical system representation of the trajectory as introduced in [44] and [68], where a mixture of Gaussian distributions forms a parametric representation of a non-linear dynamical system. Some of the challenges in the practical usage of these models include ensuring that the resultant dynamic system is stable, handling separate trajectories that cross over the same location in space and modifying the model structure based on environmental constraints.

This chapter is structured as follows. In the next section we describe the background theory on motion primitive models with non-linear dynamical systems. A stability analysis of the resultant model is presented, showing it is robust to arbitrary starting locations and accounts for oscillatory motion. An experiment is shown to demonstrate these concepts.

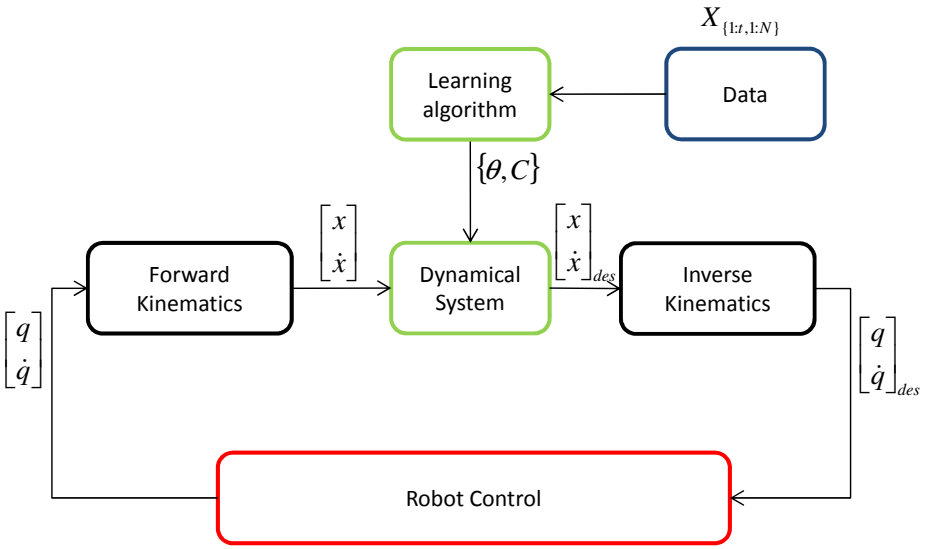

Figure 5.1: Block diagram of outer loop control system which is reliant on commands derived from a learned dynamical system. θ consists of the relevant parameters describing the dynamics or the trajectory whereas C denotes the relevant constraint parameters.

5.2 Non-Linear Dynamical System Motion Primitive Model

In this section we review the approach of learning non-linear dynamical systems with mixture of Gaussian distributions.

The trajectory of a demonstrator is defined as the evolution of a system state of dimension D denoted by $\mathbf{x} \in \mathbb{R}^D$ over a time T. The system state may be the joint angles or the Cartesian coordinates of links or end effectors, and is selected based on the context of the problem and capabilities of the robot. The set of N demonstrations, $\{\mathbf{X}_n = [\mathbf{x}_{(1,n)}, \mathbf{x}_{(2,n)}, \dots \mathbf{x}_{(T_n,n)}]\}_{n=1}^{N}$, with the nth demonstration of length T_n, that is observed by the robot constitutes the available information on the intended task. If one assumes that the system dynamics are derived by a deterministic function of the current state, which is corrupted by additive noise,

$$\dot{\mathbf{x}} = f(\mathbf{x}) + \epsilon, \tag{5.1}$$

a representation of the dynamical system may be constructed by learning a parametric model of the system position and velocity profile, $\hat{\mathbf{x}} = [\mathbf{x}^T, \dot{\mathbf{x}}^T]^T$, throughout the set of demonstrations. In this case, a mixture of Gaussian distributions is used to represent $\hat{\mathbf{x}}$. The marginal probability of a Gaussian mixture model, as per the previous chapters, is written as,

$$p(\hat{\mathbf{x}}) = \sum_{i=1}^{K} \pi_i \mathcal{N}(\hat{\mathbf{x}} | \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \tag{5.2}$$

where $\mathcal{N}(\hat{\mathbf{x}} | \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$ denotes a Gaussian density on $\hat{\mathbf{x}}$ with mean $\boldsymbol{\mu}_i$, covariance $\boldsymbol{\Sigma}_i$, and prior probability π_i for component i. Expectation-Maximisation (EM) [90] is the standard approach to learning model parameters that maximise the likelihood function. This introduces a set of latent variables, $s \in \{1, 2, \dots, K\}$, which denote

an index identifying the model component responsible for each data point, where $p(s) = \pi_s$.

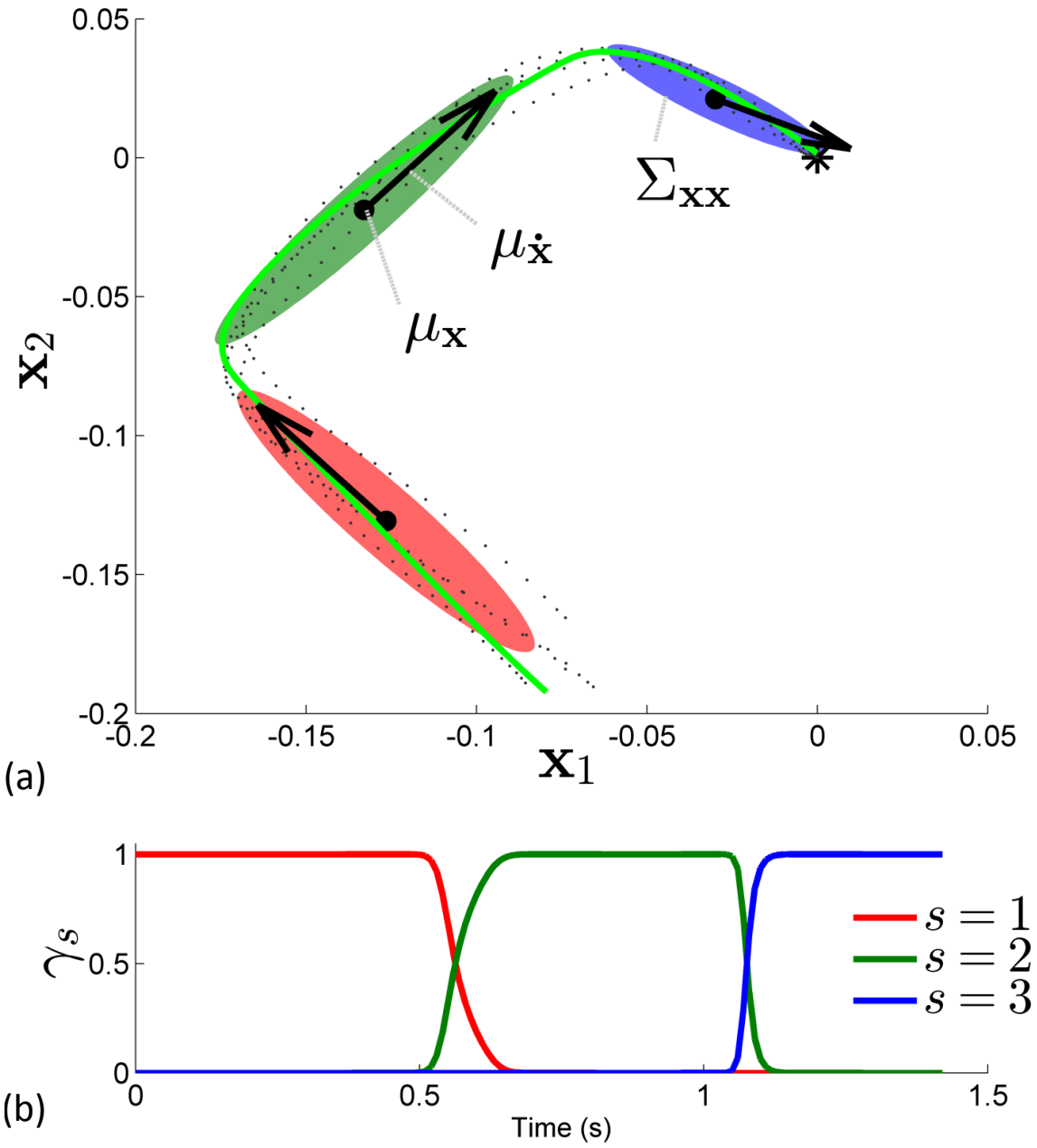

Figure 5.2: (a) A set of N demonstrations (dotted black) are represented by model consisting of a set of 3 Gaussian distributions and a simulated trajectory (green line). The arrow at the mean position of each distribution indicates the mean velocity for that particular distribution The star indicates the target position. (b) The responsibility, γ_i, of each Gaussian distribution at each point throughout a simulation of the system.

When modelling the position and velocity, $\hat{\mathbf{x}}$, in a GMM the mean vector and covariance matrix are partitioned as

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \boldsymbol{\mu}_{\dot{\mathbf{x}}} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{x}\dot{\mathbf{x}}} \\ \boldsymbol{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}} & \boldsymbol{\Sigma}_{\dot{\mathbf{x}}\dot{\mathbf{x}}} \end{bmatrix}. \tag{5.3}$$

As an example, Figure 5.2 illustrates a set of trajectories in two dimensional Cartesian coordinate space. The motion is thus partitioned into K ellipsoidal regions spanned by the position covariance, $\boldsymbol{\Sigma}_{\mathbf{xx}}$, each centered at the corresponding $\boldsymbol{\mu}_{\mathbf{x}}$, and within which the expected velocity is $\boldsymbol{\mu}_{\dot{\mathbf{x}}}$. The responsibility of a component, $p(s|\mathbf{x})$, for a particular data point is given by the following expression,

$$p(s|\mathbf{x}) = \frac{p(s)p(\mathbf{x}|s)}{\sum_{i=1}^{K} p(i)p(\mathbf{x}|i)}, \tag{5.4}$$

and is illustrated in Figure 5.2b, where we have denoted $\boldsymbol{\gamma}_s = p(s|\mathbf{x})$. Therefore the expected state velocity, $\dot{\mathbf{x}}$, at current state location, $\mathbf{x}$, is given by the conditional Gaussian relation,

$$\dot{\mathbf{x}} = \sum_{s=1}^{K} \gamma_s (\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}} + \boldsymbol{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}_{(s)}} (\boldsymbol{\Sigma}_{\mathbf{xx}_{(s)}})^{-1} (\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}_{(s)}})). \tag{5.5}$$

The system evolution over time thereby becomes a simple integration step state update, $\mathbf{x} = \mathbf{x} + \dot{\mathbf{x}}\Delta t$.

A disadvantage of directly applying the EM algorithm to learn the model parameters is that the resultant system parameters do not necessarily correspond to a stable non-linear dynamical system. Global stability of the autonomous dynamical system can be achieved by optimizing the model parameters subject to a set of stability

constraints. In [44], this system was arranged in the following manner,

$$\begin{aligned} \mathbf{A}_s &= \boldsymbol{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}_{(s)}}(\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}_{(s)}})^{-1} \\ \mathbf{b}_s &= \boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}} - \mathbf{A}_s(\boldsymbol{\mu}_{\mathbf{x}_{(s)}}) \\ \dot{\mathbf{x}} &= \sum_{s=1}^{K} \gamma_s(\mathbf{A}_s\mathbf{x} + \mathbf{b}_s), \end{aligned} \tag{5.6}$$

and a set of stability constraints were presented in terms of $\mathbf{A}$ and $\mathbf{b}$ to ensure globally stable point-to-point motion. A second optimisation program maximised the likelihood of the model given the new constraints. The constraints are designed to ensure a stable equilibrium point at the origin and was subsequently extended to achieving the equilibrium point with a given velocity. This was demonstrated with a robotic golf swing [64]. Aside from stable equilibria it may also be desirable to achieve a stable oscillation or as known in the context of non-linear dynamics, a limit cycle. Approaching an equilibrium point at a target velocity could be a solution to this problem, however, it would require a hierarchical switching mechanism between multiple models in order to produce an oscillation. An alternative is to introduce a number of adaptations of the model structure to permit either an oscillation or a stable equilibrium.

Another difficulty in reproducing arbitrary demonstrated motions are trajectories that cross-over. That is, positions where neighbouring data points, if considered independent, have conflicting velocity vectors as a result of an overlapping trajectory path. This problem can be alleviated by introducing loosening the independence assumption to a HMM thereby considering the dependency of data points neighbouring in time. By replacing γ with α the model can predict more conclusively the current state for locations in which trajectories cross-over, where for the same position there at least two states with different velocities.

The response of the system in the immediate presence of obstacles in the environment is also an important consideration. An obstacle present in the workspace can be posited as a disturbance which modulates the trajectory [65].

5.2.1 Demonstrator Representations

In many applications the desired motion for robot is represented in the task space, defined by the position and orientation of an end-effector. The dynamics of the observed motion are considered here by defining the task space as the position and linear velocity of the end-effector.

This position is with reference to the current task. It may be relative to the reference frame of a demonstrator or with respect to a location or object in the environment. The state vector may also be expanded to include additional positions if it is rather the pose or gesture of the demonstrator that is to be imitated. In the general case, there is a disparity between the kinematic structure of the demonstrator and target robot, which precludes the use of a direct joint space representation.

In this chapter we focus on the two dimensional position and velocity of an end-effector in order to simplify and clarify the modelling approach. However, the methodology can be expanded to include additional targets to handle redundancy or for imitation of more complex demonstrator postures.

5.3 Proposed Motion Primitive Model

The limitations involved with this motion primitive model are that system trajectories are stable at a point, the origin, the probability of each state in the system is independent of the previous data points which precludes the modelling of sequential motion primitives. Modelling the acceleration, which can be subjected to greater noise, is then required to differentiate between local states with different velocities.

Also an obstacle avoidance module can modulate the system when approaching an obstacle but lacks a mechanism for forward planning over a greater horizon.

To address the issue of periodic system behaviour we describe a set of control laws which can be tuned to attain limit cycle behaviour. Although one could develop a target based on the negative log-likelihood as shown in Figure 5.3, there are local minima centred on each of the components and posterior probability of a component far from the demonstrations can produce divergent system behaviour. Model modifications are shown to eliminate these potential divergent trajectories. To account for point-to-point motion we introduce an additional component to the model if at the end of demonstration the velocity drops below a threshold. The parameters of the component are designed to be an attractor and the state transition probability is set to be a terminal state. The transitional probability of predicted states at the end of the demonstrations are also tuned to only transition to the terminal component so that the system converges to the target. Allowing for implicit time dependency with the HMM results in the capability of predicting the correct velocity around co-located paths.

If the model parameters are arranged in the following way,

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \mathbf{I} & \alpha_2 \mathbf{I} & \cdots & \alpha_K \mathbf{I} \end{bmatrix} \tag{5.7}$$

$$\dot{\mathbf{x}} = \boldsymbol{\alpha} \begin{bmatrix} \mathbf{A}_1^T & \mathbf{A}_2^T & \cdots & \mathbf{A}_K^T \\ \mathbf{b}_1^T & \mathbf{b}_2^T & \cdots & \mathbf{b}_K^T \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \tag{5.8}$$

and control inputs, $\mathbf{u}$, are added to augment the system trajectory, the system equation becomes[1],

$$\dot{\mathbf{x}} = \boldsymbol{\alpha} \tilde{\mathbf{A}} \mathbf{x} + \mathbf{u}, \tag{5.9}$$

[1]The notation $(\cdot)^T$ refers to the transpose of a matrix.

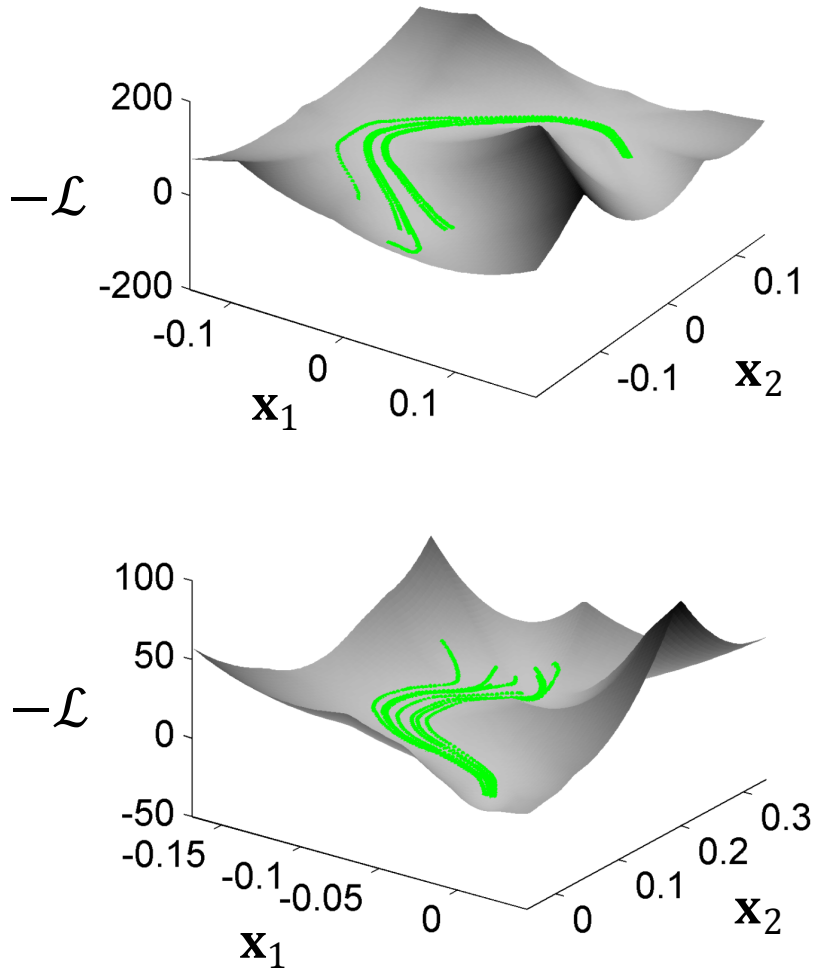

Figure 5.3: Negative log-likelihood in the workspace of the demonstrator for two example paths. Simulations initialised in different locations are shown in green and converge to a common path.

For now, we will assume $\mathbf{u} = 0$.

5.3.1 Stability Analysis

The EM algorithm seeks to maximise the likelihood of the model and the resultant parameters are not guaranteed to produce stable non-linear dynamics. Here we consider the stability of a system in a low dimensional space. Optimising the stability of a non-linear dynamical system derived from mixture models was addressed in [44] for point-to-point motion. A set of stability conditions were shown to ensure global

asymptotic stability at a target. However, the conditions are too conservative to allow stable oscillations such as those observed in a gait. We propose a control law to ensure the system will converge to one of two behaviours; a stable oscillation or approach an equilibrium point.

Each component, s, in the proposed model is governed by linear system dynamics, $\mathbf{A}_s = \mathbf{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}_{(s)}}(\mathbf{\Sigma}_{\mathbf{x}\mathbf{x}_{(s)}})^{-1}$, as in Equation (6.10). According to contraction analysis [126], if the symmetric part of the system Jacobian is uniformly negative definite across the state space, the system is contracting. That is, a positive constant ζ exists such that $\frac{1}{2}(\frac{d\dot{\mathbf{x}}}{d\mathbf{x}} + \frac{d\dot{\mathbf{x}}}{d\mathbf{x}}^T) \leq -\zeta\mathbf{I} < 0$ for all $\mathbf{x}$. This means that any two neighbouring trajectories ultimately converge. Since for each component $\frac{d\dot{\mathbf{x}}}{d\mathbf{x}} = \mathbf{A}_s$, setting $\mathbf{A}_s = \mathbf{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}_{(s)}}(\mathbf{\Sigma}_{\mathbf{x}\mathbf{x}_{(s)}})^{-1} - (\sigma_{max} + \zeta)\mathbf{I}$, where σ_{max} is the maximum positive eigenvalue (otherwise zero) and ζ is a small positive constant, ensures that each subsystem is stable. Additionally, any convex combination of a set of contracting systems, as subsumed by the posterior probability, is also a contracting system and thus stable.

However, for a stable oscillatory motion such as a limit cycle any two points that reside along the cycle are out of phase and do not converge. If the Jacobian is negative semi-definite ($\zeta = 0$) this corresponds to a semi-contracting system, in that it only contracts along directions associated with negative eigenvalues. Define

$$\mathcal{V}_s = \frac{\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}}\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}}^T}{\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}}^T\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}}}, \tag{5.10}$$

as the projection matrix for the expected velocity of component s. Applying a nullspace projection operator, $(\mathbf{I} - \mathcal{V}_s)$, to the error between the current position and the component mean projects onto a transverse coordinate space, perpendicular to the expected trajectory. Minimising this quantity shifts the trajectory to coincide with the expected trajectory by a factor of β. Therefore, with $\mathbf{A}_s = \mathbf{\Sigma}_{\dot{\mathbf{x}}\mathbf{x}_{(s)}}(\mathbf{\Sigma}_{\mathbf{x}\mathbf{x}_{(s)}})^{-1} - \sigma_{max}\mathbf{I} - \beta(\mathbf{I} - \mathcal{V}_s)$, the system should be contracting in all directions

apart from the expected velocity.

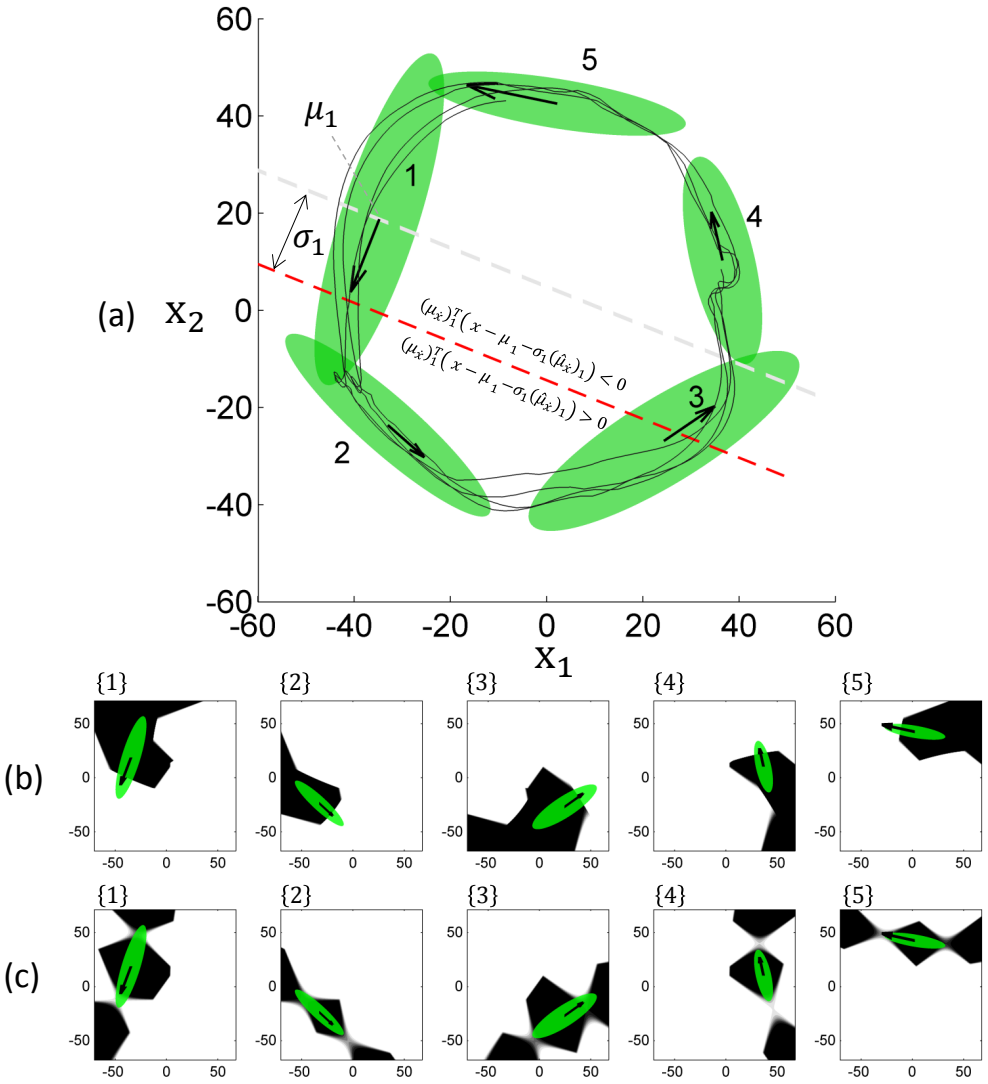

Figure 5.4: Depiction of the transformation applied to the probability density of each component. (a) A model of 5 components is used to describe example 2D data. Points that reside in the halfspace that opposes the velocity are suppressed. (b) The posterior probability after skewing each density along the direction of the velocity. (c) The posterior probability before skewing the density functions.

Furthermore, let the region $\mathcal{D}$ be the set of points in the latent space such that the likelihood, given the model, is above the minimum likelihood of any one of the demonstrated points, $\mathcal{D} = \{\mathbf{x} \in \mathbb{R}^d \mid p(\mathbf{x}) \geq \mathcal{L}_{min}\}$, where $\mathcal{L}_{min} = \min(p(\mathbf{x}_t)), \forall t$. The region $\mathcal{D}$ encompasses all of the demonstrations, and the boundary of this region defines a contour of minimum likelihood. It is assumed that the model is not to

be extrapolated far beyond this region. To ensure that this is indeed the case, we define a damping function according to $\sum_{s=1}^{K} \alpha_s(\mathbf{x} - \boldsymbol{\mu}_s)$, which directs the system toward a probabilistic combination of the component means. For points within $\mathcal{D}$ the model parameters control the trajectory, otherwise outside $\mathcal{D}$ the damping function is deployed to approach $\mathcal{D}$. A sigmoid function provides a smooth transition between these regions along the boundary,

$$\mathcal{P} = (1 - (\frac{1}{\phi} - 1)e^{-\varphi(\mathcal{L}_{min} - \mathcal{L}(\mathbf{x}))})^{-1}, \tag{5.11}$$

where φ controls the smoothness of the transition at the boundary and the center of sigmoid is shifted to set the level of damping to $\phi \in (0, 1)$ along the $\mathcal{L}_{min}$ contour. The parameter ϕ is interpreted as the extent to which the model dynamics are trusted along the boundary of demonstrations. The system behaviour design in regions of low likelihood may be further tuned based on workspace, environment and safety constraints as required.

The probability of each component determines the local system behaviour, yet, as shown in Figure 5.4c, there are regions of high posterior probability where $\boldsymbol{\mu}^T_{\dot{\mathbf{x}}_{(s)}}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}_{(s)}}) > 0$. This implies that unintended equilibria may exist near the boundary of $\mathcal{D}$. Therefore, the probability of each component is modulated along the axis aligned with the velocity, $\boldsymbol{\mu}^T_{\dot{\mathbf{x}}_{(s)}}(\mathbf{x} - \boldsymbol{\mu}_s)$ with a gating function, $\lambda_s = (1 + e^{-\sigma_1^{-1}\hat{\boldsymbol{\mu}}^T_{\dot{x}_{(s)}}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}_{(s)}} - \sigma_1 \hat{\boldsymbol{\mu}}_{\dot{x}_{(s)}})})^{-1}$, where σ_1 is the largest eigenvalue of $\boldsymbol{\Sigma}_{\mathbf{xx}}$ and $\hat{\boldsymbol{\mu}}_{\dot{\mathbf{x}}_{(s)}}$ is the normalised velocity direction. Computing

$$p(s|\mathbf{x}) = \frac{p(s|\mathbf{x})\lambda_s}{\sum_i p(i|\mathbf{x})\lambda_i}, \tag{5.12}$$

introduces a skew to each component in the direction of the expected velocity, effectively eliminating components that have been recently activated and avoiding cases of

divergence when the system is initialised beyond the demonstrated workspace. The modified posterior probability is shown in Figure 5.4b. Further optimisation of this modulation can be achieved by varying the offset and slope of λ with equivalent parameters to ϕ and φ in $\mathcal{P}$.

Finally, the system equation is

$$\dot{\mathbf{x}} = \mathcal{P} \sum_{s=1}^{K} \alpha_s (\boldsymbol{\mu}_{\dot{\mathbf{x}}_{(s)}} + (\mathbf{A}_s - (\frac{1}{\mathcal{P}} - 1)\mathbf{I})(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}_{(s)}})), \tag{5.13}$$

and the parameters were chosen as $\{\beta, \phi, \varphi\} = \{\sigma_{max}, 0.9, 1\}$.

Validation

The Lyapunov function candidate presented in [44] is related to stability at a point, the origin. It is strictly positive definite for all $\mathbf{x}$ except at the origin and its derivative is strictly negative definite. For an oscillation or periodic orbit, any point that resides on the orbit has zero energy, or has converged to the target behaviour. This is a case where the target behaviour resides in an invariant set and can be analysed with invariant set principle [127]. Given a Lyapunov function $V(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$, the system is stable if

$$\begin{aligned} V(\mathbf{x}) &> 0, \quad \forall\, \mathbf{x} \notin \Theta \\ \dot{V}(\mathbf{x}) &\leq 0, \quad \forall\, \mathbf{x} \\ V(\mathbf{x}) &\to \infty, \text{if } \mathbf{x} \to \infty \end{aligned} \tag{5.14}$$

where the set Θ contains either the points on the target limit cycle or an equilibrium point. Based on the modifications in the previous section, let the proposed Lyapunov function candidate be given by,

$$V(\mathbf{x}) = \mathcal{P}(\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathbf{x}})^T (\mathbf{I} - \hat{\mathcal{V}})(\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathbf{x}}) + (1 - \mathcal{P})\|\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathbf{x}}\|^2 \tag{5.15}$$

where we have defined,

$$\hat{\boldsymbol{\mu}} = \sum_{k}^{K} \boldsymbol{\gamma}_k(\mathbf{x})\boldsymbol{\mu}_{\mathbf{x}_k}, \quad \hat{\mathcal{V}} = \sum_{k}^{K} \boldsymbol{\gamma}_k(\mathbf{x})\mathcal{V}_k. \tag{5.16}$$

$V(\mathbf{x})$ is a positive definite function which tends to zero as it approaches a limit cycle. Once again, $\mathcal{P}$ provides a transition between a global stability region and local stability region. Therefore $\dot{V}(\mathbf{x})$ becomes,

$$\begin{aligned}
\dot{V}(\mathbf{x}) &= \Big[\frac{dV(\mathbf{x})}{d\mathbf{x}}\Big]^T \frac{d\mathbf{x}}{dt} \\
\frac{dV(\mathbf{x})}{d\mathbf{x}} &= \frac{d\mathcal{P}}{d\mathbf{x}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T(\mathbf{I}-\hat{\mathcal{V}})(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) + 2\mathcal{P}\Big[\frac{d}{d\mathbf{x}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T\Big](\mathbf{I}-\hat{\mathcal{V}})(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) \\
&\quad + \mathcal{P}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T\Big[\frac{d}{d\mathbf{x}}(\mathbf{I}-\hat{\mathcal{V}})\Big](\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) + 2(1-\mathcal{P})\Big[\frac{d}{d\mathbf{x}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T\Big](\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) \\
&\quad - \frac{d\mathcal{P}}{d\mathbf{x}}\|\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}\|^2
\end{aligned} \tag{5.17}$$

Adopting an approximation, that $\frac{d\gamma(\mathbf{x})}{d\mathbf{x}} \approx 0$, due the smoothness and boundedness of $\gamma(\mathbf{x})$, since it is defined by a sum of exponential functions and as a probability is bounded by $\gamma(\mathbf{x}) \in [0,1]$, simplifies the analysis of $\dot{V}(\mathbf{x})$. Now,

$$\dot{V}(\mathbf{x}) \approx -\Big[\frac{d\mathcal{P}}{d\mathbf{x}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T\hat{\mathcal{V}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) + 2(\mathbf{I}-\mathcal{P}\hat{\mathcal{V}})(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})\Big]^T \dot{\mathbf{x}}. \tag{5.18}$$

To analyse the first term of the equation, the quadratic, $(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T\hat{\mathcal{V}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})$, is positive whereas $\Big[-\frac{d\mathcal{P}}{d\mathbf{x}}\Big]^T\dot{\mathbf{x}} < 0$. This is because the gradient of $\mathcal{P}$ is always in the direction of the minimum likelihood contour and under the assumption that the skew introduced by λ ensures that $\dot{\mathbf{x}}$ does not diverge in this region. For the second term, the global stability compensation, where $\mathcal{P} \to 0$, is $-(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})$. As $\mathcal{P} \to 1$, $(\mathbf{I}-\hat{\mathcal{V}})\hat{\boldsymbol{\mu}}_{\dot{\mathbf{x}}} = 0$, and the term $(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}})^T(\mathbf{I}-\hat{\mathcal{V}})\hat{\mathbf{A}}(\mathbf{x}-\hat{\boldsymbol{\mu}}_{\mathbf{x}}) \leq 0$, since we have set each motion primitive matrix $\mathbf{A}_s$ to be negative semi-definite in Section 5.3.1 and $\mathbf{I}-\hat{\mathcal{V}}$ is an idempotent projection matrix. Therefore, the criterion $\dot{V}(\mathbf{x}) \leq 0$ holds for this

approximation. Regions of increasing $V(\mathbf{x})$ occur when $\frac{d\gamma(\mathbf{x})}{d\mathbf{x}}$ is significant. It must be noted that to satisfy this criterion strictly one can increase β, however, the resultant trajectory would resemble a linear interpolation between the intersection points of neighbouring motion primitives velocity vectors.

5.4 Experimental Validation

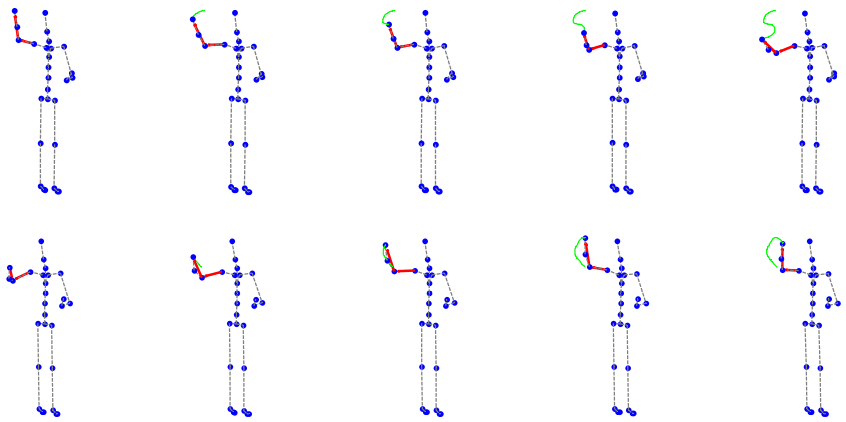

Figure 5.5: Visualisation of the posture of the motion capture demonstrations. Two of the task space paths are shown in green at various stages of the sequence, and the positions of wrist and elbow for tele-operation are shown in red.

An experiment was devised to demonstrate a set of 2 dimensional trajectories to a robot. A set of arm motions restricted to a 2 dimensional plane were gathered using an inertial motion capture suit MVN [1], as was described in Chapter 3. The posture of the body is shown in Figure 5.5, where the position of the hand was the demonstrated target trajectory for the robot. Eight handwritten characters were recorded as test trajectories each with at least 5 demonstrations. The paths included regions of overlap and some were formed as cycles. This allowed for the evaluation of robustness of the system to these previously avoided cases.

A 6-DOF robot arm was constructed from a set of servo-motors [128]. Four MX-28 motors were used for the equivalent shoulder and elbow joints with each having

a stall torque of 24kg.cm. The wrist joints were controlled with AX-18 motors of 18kg.cm. The kinematic structure of the robot was designed for anthropomorphism for ease of kinaesthetic teaching and correspondence mapping. It is described with Denavit-Hartenberg (DH) parameters in Appendix A.

5.4.1 Results

Applying the learning algorithm to the recorded demonstrations results in the models as shown in Figure 5.6. Example simulations according to the model are also displayed over the state space (green trace). In each case a similar motion to the observations is reconstructed. The character 'B' and the loop trajectory in the top right of the figure present the challenge of trajectories that crossover each other. A GMM system would not be able to distinguish between the two feasible directions at the same position. However, employing a HMM transitional probability enables the model to resolve the most likely direction based on the states observed previously in the sequence. Another situation which presents a challenge is a change in the opposite direction as in the character 'A'. In this case the simulation does not explore an overlapping section of the state space. It is a region where, depending on the location of the components, the probability according to the position surpasses that of the transitional probability. Another feature of the simulated trajectory is that the deviation from the mean velocity is minimised, which can lead to a path resembling a piecewise linear interpolation ignoring the local covariance.

In Figure 5.7, the energy-like function $V(\mathbf{x})$ is displayed over two simulations. It is clear that the function is decreasing as the system approaches a nominal trajectory, however, small increases are observed during some motion primitive transitions when $\frac{d\alpha(\mathbf{x})}{d\mathbf{x}}$ is significant. These increases are observed in 5.7(a) and 5.7(c) but are not a source of instability. In 5.7(b) and 5.7(d) the value of $V(\mathbf{x})$ across the workspace is

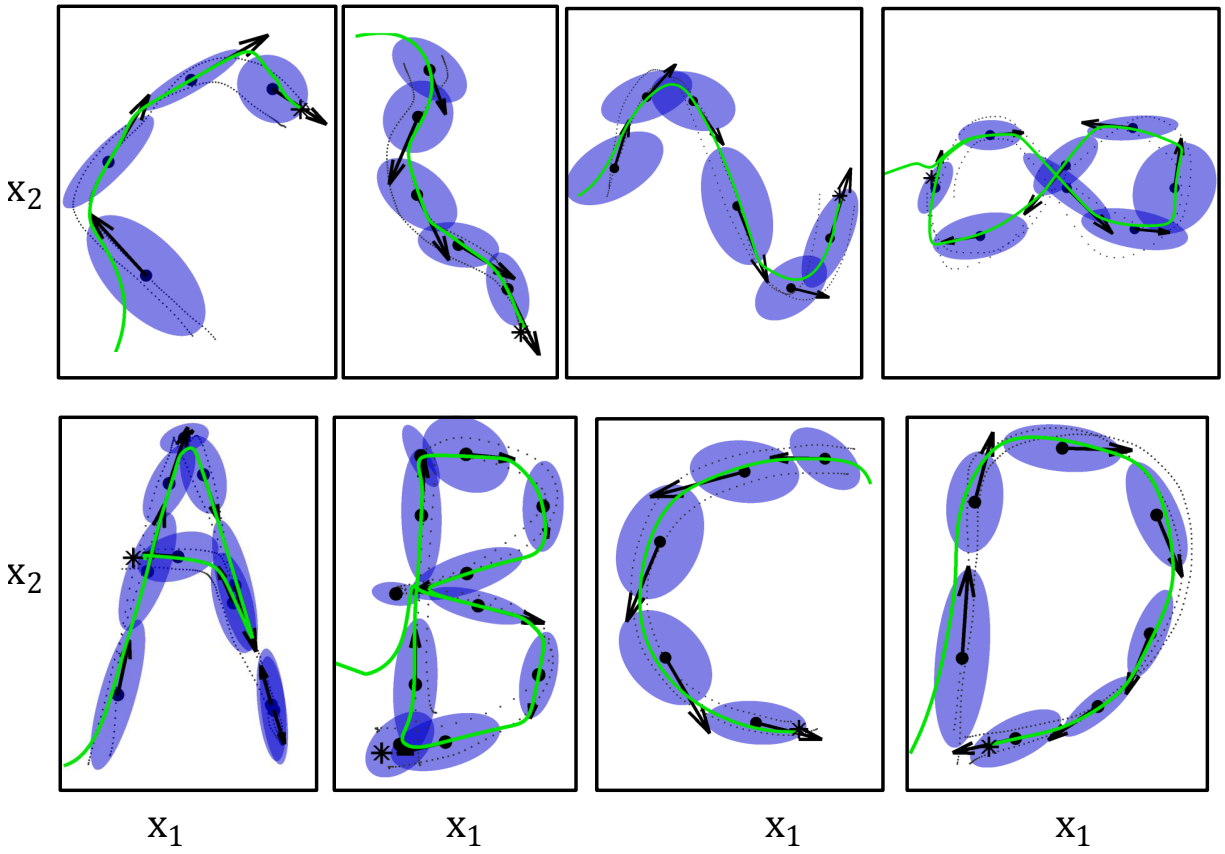

Figure 5.6: Simulation of the non-linear dynamical systems learned from demonstration. Each motion primitive (blue ellipsoid) is associated with a velocity (black vector). In each case the system reconstructed a generalised trajectory, in green, which is similar to the observations.

displayed with the simulated trajectory. Importantly, the target trajectory resides in the region of the state space where $V(\mathbf{x})$ approaches zero and in regions of low likelihood $V(\mathbf{x})$ is decreasing.

Finally, in Figure 5.8 the simulated trajectories are replayed by the 6DOF robot. An inverse kinematics routine resolves a joint space trajectory based on the target system trajectory along a plane that is aligned with a whiteboard. In 5.8(a) and (b), a simulation of the robot kinematics and in the real robot during the course of a trajectory are shown respectively. The pictures in 5.8(c-e) depict example results from the whiteboard. Two of the trajectories that contain crossover regions are

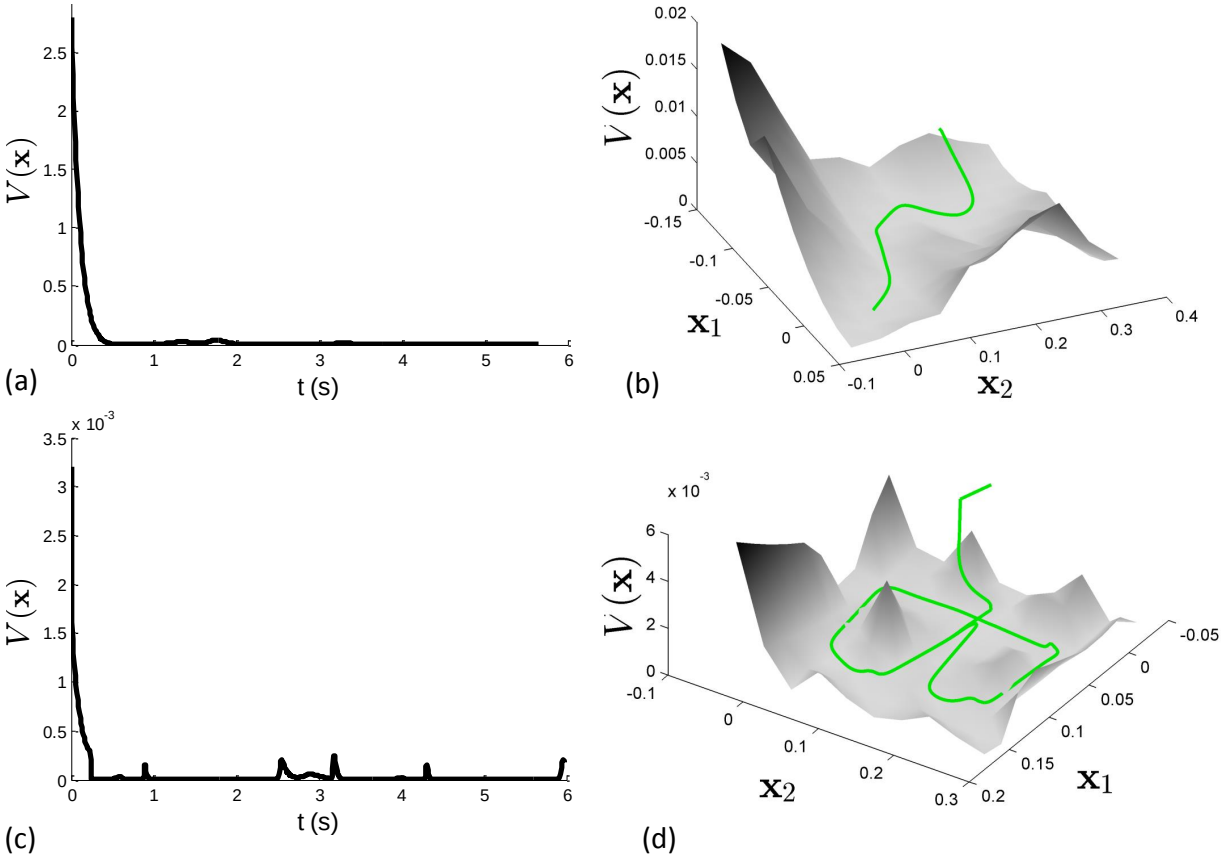

Figure 5.7: (a) The non-negative function $V(\mathbf{x})$ throughout a simulation over time. It only increases in some transitions between motion primitives. (b) A visualisation of variance in $V(\mathbf{x})$ across the state space. (c) A second example of the function $V(\mathbf{x})$ during a simulation reconstructing the character 'B'. (d) Example of the variance of the function $V(\mathbf{x})$ across the state space.

shown as well as the outcome from initialising the dynamical system from a range of starting locations. The robot trajectory was in agreement with the simulation and despite starting the system from a range initial positions, system contracted to a single trajectory.

5.5 Discussion

Modelling a trajectory which oscillates along a line is problematic for a system represented by the construction presented here. This situation occurred in the demon-

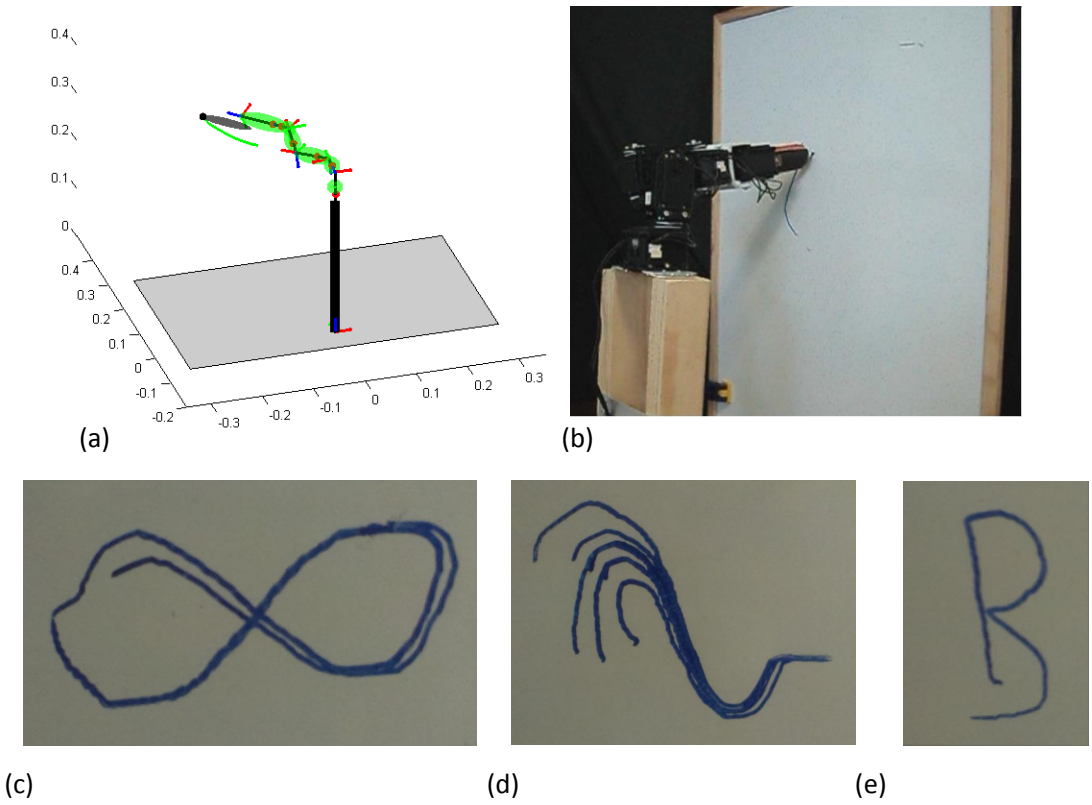

Figure 5.8: (a) Simulation of 5-DOF robot where each robot link is a green ellipsoid and the pen position is represented by the grey ellipsoid. (b) Actual robot experiment of writing a planar trajectory on whiteboard. (c) Example experimental result on the whiteboard from a cyclic trajectory with a crossover point. (d) Example trajectories from several starting locations converging to a single trajectory. (e) Example experiment of writing character B, where the system was able to resolve the correct path in the central region of potential uncertainty.

stration of the handwritten character 'A' in Figure 5.6. A possible solution to this scenario is to map this motion into another set of coordinates that separate both directions or include additional features in the system state that are independent of this direction. Alternatively, one could partition an oscillation into two simpler connected paths activated in sequence. This complicates the reconstruction process into potentially multiple submodels which, individually, do not intuitively relate to the intended demonstration.

The approximation of the function $\dot{V}(\mathbf{x})$ which neglects the derivative of the pos-

terior probability does not strictly prove that the system is stable despite successful convergence in simulation. Devising a strict Lyapunov function and control law is a difficult task since even detecting the existence of limit cycles is, in general, a hard problem. Moreover, in this case the non-linear dynamical system returned from learning may not immediately exhibit a limit cycle or orbit but this behaviour may be the most appropriate manner in which to describe the data.

A plausible direction for detecting and stabilising periodic orbits is to employ the concept transverse contraction. This method introduces a phase variable for the cycle, determining the current likely phase and applying the appropriate compensation to converge to the cycle [129]. However, in some cases it may impose prior knowledge on the system by assuming a limit cycle exists. Either additional domain information or a search for stable oscillations is required to resolve this.

5.6 Chapter Summary

In this chapter we analysed an approach for reproducing demonstrated actions from motion primitive models. It is based on positing a non-linear dynamical system parameterised by a Gaussian mixture model. Maximum likelihood learning does not take into account the dynamic stability of the final system therefore control laws are implemented to compensate. Using Lyapunov stability analysis it was shown that if the likelihood function varies smoothly within the state space the system is asymptotically stable. The modifications tend to stabilise the systems as demonstrated by a robot arm experiment which showed successfully reproduced a number of actions from the motion primitive model.

Chapter 6

Humanoid robot imitation on a manifold

6.1 Introduction

The challenge presented by the extension of non-linear dynamical system control of [130] to high degree of freedom, nonholonomic robots is that the resultant systems typically suffer from degenerate modes, instability issues and quadratic increases in model complexity. Although a humanoid end effector can track a 3 dimensional position using inverse kinematics, there are a significant number of actions that one may wish to demonstrate requiring the positioning of multiple joints simultaneously. A common example would be that of a walking gait. Furthermore, one often seeks a correspondence model between the demonstrator and robot, if it is tele-operated through the task rather than guided manually.

This chapter explores the exploitation of the local covariance structure of the data by assuming that the demonstrations reside along a continuous manifold, a topological space that is locally Euclidean, that constrains the motion to a feasible subspace. Although this assumption is an approximation of the data density which

can introduce a level of error into the reconstruction, the structure is less subject to degenerate and unstable modes as the data increases in dimensionality.

There are many approaches to learning the structure of a manifold. Some methods are dependent on identifying a neighbourhood graph structure of the data points based on a local geodesic distance metric such as ISOMAP [85] or local linear embedding (LLE) [131] while other approaches focus on aligning local eigenvector coordinates such as manifold charting [132] or coordinated factor analysis [83]. Recent applications of dimensionality reduction for controlling robots include deploying Gaussian Process Latent Variable Models (GPLVM) [133] whereby a shared manifold was learned for real-time correspondence mapping [87]. Despite the high fidelity of these models, the complexity of GPLVM remains an issue due to the estimation of covariance matrices that expand with sample size. Another is the local manifold smooth learning [134] that is based on local radial basis functions was applied to robot arms and humanoid robots [84]. Although this method is structurally similar to the coordinated factor analysis (CFA) model, it focuses on low level planning for motion generation on the manifold, whereas the model proposed here is able to dynamically react in a local region without having to necessarily plan the remainder of the path again.

This chapter initially describes the process for translating motion between a human and robot for constructing a correspondence map, as well as the consideration of constraints, collisions and structural stability which are important to represent in the motion generation phase. The proposed modeling approach based on CFA and HMM is then presented including an algorithm for iteratively estimating the number of components. Finally, experiments which involve learning models of walking cycles and reaching tasks are shown to generate feasible motion for a robot. The results are reported in a simulation and with the target 19DOF humanoid robot [135].

6.2 Globally Coordinated Mixture of Factor Analysers

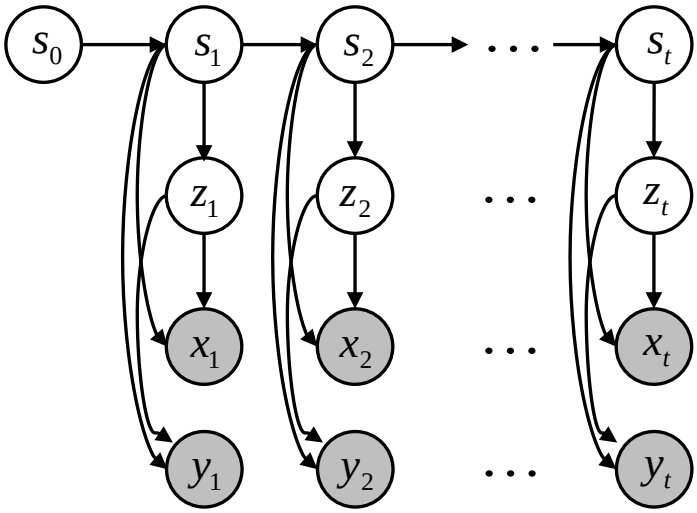

Figure 6.1: Dynamic coordinated mixture of factor analyzers model of corresponding data sets, where $\mathbf{x}$ refers to the human motion and $\mathbf{y}$ refers to the robot trajectory.

In the context of imitation, assume that there are two data sets supplied, the demonstrator data, $\mathbf{x}$, and the corresponding robot posture, $\mathbf{y}$.

Within the same framework of mixture models, we can assume that each component is composed of a factor analyzer [136]. A factor analyzer seeks to relate D-dimensional data vectors to d-dimensional latent factor variables, such that $d \ll D$, in the form of a linear function,

$$\mathbf{x} = \mathbf{\Lambda}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{6.1}$$

where $\mathbf{z}$ are the latent variables, $\mathbf{\Lambda}$ are the linear factor loadings, $\boldsymbol{\mu}$ the mean of the data points and $\boldsymbol{\epsilon}$ is an assumed noise model. If the latent variables are assumed to be independent Gaussian distributed with unit variance, $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, and the noise also Gaussian distributed, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{\Psi})$, this induces a distribution for the observations as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})$. Here $\mathbf{\Psi}$ is a positive diagonal matrix which adds variance

along each dimension. Incidentally, if $\boldsymbol{\epsilon}$ is constrained to an isotropic Gaussian, by setting $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$, this resolves to principal component analysis (PCA).

Graphically, this model introduces a continuous intermediary variable, $\mathbf{z}$, which can relate two corresponding data sets of the demonstrator motion, $\mathbf{x}$, and the robot motion, $\mathbf{y}$, as shown in Figure 6.1. Equivalently interpreted parameters, $\boldsymbol{\Lambda}_\mathbf{y}$ and $\boldsymbol{\mu}_\mathbf{y}$ are defined for the manifold in robot coordinates.

In related literature, demonstrations were projected onto a single set of eigenvectors or principal components as a pre-processing stage before learning a GMM. For a mixture of factor analyzers (MFA), on the other hand, each of the K model components has a set of factor loadings and thereby a separate latent space, $\mathbf{z}_{(i)}$ for the ith mixture component. Thus, the coordinate systems $\mathbf{z}_{n_{(i)}}$ and $\mathbf{z}_{n_{(j)}}$ do not necessarily agree even when the two components share significant probability mass for the point $\mathbf{z}_n$. If the coordinates are not consistent then there is clearly a multimodality in the distribution $p(\mathbf{z}|\mathbf{x}_n)$ as shown in Figure 6.2a, where the coordinates of neighbouring components are not guaranteed to align.

To address this dilemma, it was noted in [137] that the MFA model exhibits an invariance for the internal coordinates of each component. That is, applying an arbitrary rotation to the factor loadings, $\boldsymbol{\Lambda}$, does not fundamentally affect the objective function for EM. We use the formulation of the coordinated MFA model as presented in [83] where auxiliary parameters, $\boldsymbol{\kappa}_i$ and $\boldsymbol{\Sigma}_i$ in $p(\mathbf{z}|i) \sim \mathcal{N}(\mathbf{z}|\boldsymbol{\kappa}_i, \boldsymbol{\Sigma}_i)$, are chosen to achieve local coordination between the components. A cost function is constructed to penalize multimodality per data point by evaluating the Kullback-Leibler (KL) divergence between $p(\mathbf{z}|\mathbf{x}_n)$ and a Gaussian distribution, $q(\mathbf{z}|\mathbf{x}_n) \sim \mathcal{N}(\mathbf{z}|\mathbf{z}_n, \boldsymbol{\Sigma}_n)$, centered at each data point. By further introducing a distribution over the mixture components with $q(s|\mathbf{x}_n)$, the penalized log-likelihood function can be

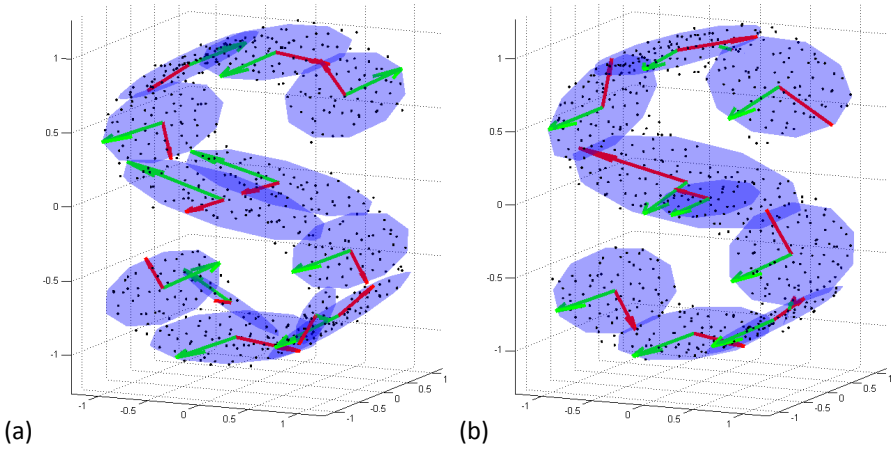

Figure 6.2: Comparison of the local coordinate systems in a mixture of factor analysers, (a), against the coordinated factor analysis model, (b), on a synthetic test data set used for demonstrating manifold learning. Each component is displayed as an ellipsoid where the major axes are scaled to 2 standard deviations. Coordinate systems of neighboring components are aligned in the CFA model, that is, the red and green arrows of neighboring components should point in similar directions, whereas there is no such guarantee for MFA. The data reconstruction error, E_{rec}, for CFA was 1.8, which was less than 1% of the E_{rec} when using PCA.

written as,

$$\mathcal{F} = \sum_{n,s}^{N} \int q(\mathbf{z}, \mathbf{s}|\mathbf{x}_n, \mathbf{y}_n) \log \frac{p(\mathbf{x}_n, \mathbf{y}_n, \mathbf{z}, \mathbf{s})}{q(\mathbf{z}, \mathbf{s}|\mathbf{x}_n, \mathbf{y}_n)} d\mathbf{z}. \tag{6.2}$$

This bounded likelihood can be computed, ignoring constant terms, as

$$\mathcal{F} = \sum_{i=1}^{K} \sum_{n=1}^{N} q_{ni} \left[\mathcal{S}_{ni} - \mathcal{E}_{ni}\right]$$

$$\mathcal{S}_{ni} = \frac{1}{2} \log |\mathbf{\Sigma}_n| - \log q_{ni}$$

$$\begin{aligned}
\mathcal{E}_{ni} = &- \log p(i) + \log |\mathbf{\Sigma}_i| + \frac{1}{2} \mathbf{Tr}\{\mathbf{\Sigma}_i^{-1}(\mathbf{\Sigma}_n + \mathbf{z}_{ni}\mathbf{z}_{ni}^T) \\
&+ \frac{c_n^x}{2}\left[\mathbf{Tr}\{\mathbf{\Sigma}_n \mathbf{\Lambda}_i^{\mathbf{x}T} \mathbf{\Psi}_i^{\mathbf{x}-1} \mathbf{\Lambda}_i^{\mathbf{x}}\} + \log |\mathbf{\Psi}_i^{\mathbf{x}}|\right] \\
&+ \frac{c_n^y}{2}\left[\mathbf{Tr}\{\mathbf{\Sigma}_n \mathbf{\Lambda}_i^{\mathbf{y}T} \mathbf{\Psi}_i^{\mathbf{y}-1} \mathbf{\Lambda}_i^{\mathbf{y}}\} + \log |\mathbf{\Psi}_i^{\mathbf{y}}|\right] \\
&+ \frac{c_n^x}{2}(\mathbf{x}_{ni} - \mathbf{\Lambda}_i^{\mathbf{x}} \mathbf{z}_{ni})^T \mathbf{\Psi}_i^{\mathbf{x}-1} (\mathbf{x}_{ni} - \mathbf{\Lambda}_i^{\mathbf{x}} \mathbf{z}_{ni}) \\
&+ \frac{c_n^y}{2}(\mathbf{y}_{ni} - \mathbf{\Lambda}_i^{\mathbf{y}} \mathbf{z}_{ni})^T \mathbf{\Psi}_i^{\mathbf{y}-1} (\mathbf{y}_{ni} - \mathbf{\Lambda}_i^{\mathbf{y}} \mathbf{z}_{ni}),
\end{aligned} \tag{6.3}$$

with $q_{ni} = q_n(i)$, $\mathbf{x}_{ni} = \mathbf{x}_n - \boldsymbol{\mu}_i^{\mathbf{x}}$, $\mathbf{y}_{ni} = \mathbf{y}_n - \boldsymbol{\mu}_i^{\mathbf{y}}$ and $\mathbf{z}_{ni} = \mathbf{z}_n - \boldsymbol{\kappa}_i$. The logical indices, c_n^x and c_n^y, relate a corresponding data point n if $c_n^x = c_n^y = 1$. The update equations to optimize this objective function based on EM are listed in Algorithm (2) and are also reported in [83]. The expressions for the mapping to $\mathbf{y}$ coordinates ($\boldsymbol{\mu}^{\mathbf{y}}$,$\boldsymbol{\Lambda}^{\mathbf{y}}$,$\boldsymbol{\Psi}^{\mathbf{y}}$) are equivalent to those listed for $\mathbf{x}$.

Upon achieving a local optimum we have a function which maps points in data space to consistent coordinates in the latent space, $f_{\mathbf{x}\to\mathbf{z}}$, and a function which maps points in the latent space back into the data space, $f_{\mathbf{z}\to\mathbf{x}}$. A trajectory in $\mathbf{z}$ traverses a manifold in $\mathbf{x}$ as indicated by the consistency of neighbouring component coordinates in Figure 6.2b. The expected value of $\mathbf{z}$ given a data point $\mathbf{x}$ is

$$\langle p(\mathbf{z}|\mathbf{x})\rangle = \sum_{i=1}^{K} \gamma_i \left[\boldsymbol{\kappa}_i + \mathbf{V}_i^{\mathbf{x}} \boldsymbol{\Lambda}_i^{\mathbf{x}T} \boldsymbol{\Psi}_i^{\mathbf{x}-1} \mathbf{x}_{ni}\right] \tag{6.4}$$

where $\mathbf{V}_i^{\mathbf{x}} = (\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Lambda}_i^{\mathbf{x}T} \boldsymbol{\Psi}_i^{\mathbf{x}-1} \boldsymbol{\Lambda}_i^{\mathbf{x}})^{-1}$. Reciprocally, the expected value of a data point given a point in the latent coordinate space is

$$\langle p(\mathbf{x}|\mathbf{z})\rangle = \sum_{i=1}^{K} \hat{\gamma}_i \left[\boldsymbol{\mu}_i^{\mathbf{x}} + \boldsymbol{\Lambda}_i^{\mathbf{x}} \mathbf{z}_{ni}\right], \tag{6.5}$$

where $\hat{\gamma}_i$, or $\hat{\alpha}_i$ for the HMM, refers to the responsibility computed in the latent space as opposed to the data space[1]. Analogous functions exist for $f_{\mathbf{z}\to\mathbf{y}}$ and $f_{\mathbf{y}\to\mathbf{z}}$. The performance of the embedding function can be assessed with the reconstruction error by computing

$$E_{rec} = \sum_{n=1}^{N} \|\mathbf{x}_n - f_{\mathbf{z}\to\mathbf{x}}(f_{\mathbf{x}\to\mathbf{z}}(\mathbf{x}_n))\|. \tag{6.6}$$

In order to use the GMM non-linear dynamical system method we estimate the

[1]The notation $\langle\cdot\rangle$ refers to the expected value operator.

Algorithm 2: Coordinated Factor Analysis

E-step:

Compute $\mathcal{E}_{ni}$, $\forall\ n = 1 \dots N,\ i = 1 \dots K$

$$q_{ni} = \frac{e^{\mathcal{E}_{ni}}}{\sum_{i=1}^{K} e^{\mathcal{E}_{ni}}}$$

$$\mathbf{V}_{ni} = \mathbf{\Sigma}_i^{-1} + c_n^x \mathbf{\Lambda}_i^{\mathbf{x}T} \mathbf{\Psi}_i^{\mathbf{x}-1} \mathbf{\Lambda}_i^{\mathbf{x}} + c_n^y \mathbf{\Lambda}_i^{\mathbf{y}T} \mathbf{\Psi}_i^{\mathbf{y}-1} \mathbf{\Lambda}_i^{\mathbf{y}}$$

$$\mathbf{\Sigma}_n^{-1} = \sum_{i=1}^{K} q_{ni} \mathbf{V}_{ni},$$

$$\mathbf{z}_n = \mathbf{\Sigma}_n \sum_{i=1}^{K} q_{ni} \big[\mathbf{V}_{ni} \boldsymbol{\kappa}_i + c_n^x \mathbf{\Lambda}_i^{\mathbf{x}T} \mathbf{\Psi}_i^{\mathbf{x}-1} \mathbf{x}_{ni} + c_n^y \mathbf{\Lambda}_i^{\mathbf{y}T} \mathbf{\Psi}_i^{\mathbf{y}-1} \mathbf{y}_{ni} \big]$$

M-step:

$$p(i) = \frac{1}{N} \sum_{i=1}^{K} q_{ni}$$

$$\boldsymbol{\kappa}_i = \sum_{n=1}^{N} \tilde{q}_{ni} \mathbf{z}_n, \text{ where } \tilde{q}_{ni} = \frac{q_{ni}}{\sum_{\tilde{n}=1}^{N} q_{\tilde{n}i}}$$

$$\mathbf{\Sigma}_i = \sum_{n=1}^{N} \tilde{q}_{ni} [\mathbf{z}_{ni} \mathbf{z}_{ni}^T + \mathbf{\Sigma}_n]$$

$$\mathbf{\Lambda}_i^{\mathbf{x}} = \Big(\sum_{n=1}^{N} q_{ni}^{\mathbf{x}} \mathbf{x}_{ni} \mathbf{z}_{ni}^T - \mathbf{x}_{ni}^{\mathbf{x}} \mathbf{z}_{ni}^{\mathbf{x}\,T} \Big) \times \Big(\sum_{n=1}^{N} q_{ni}^{\mathbf{x}} [\mathbf{z}_{ni} \mathbf{z}_{ni}^T + \mathbf{\Sigma}_n] - \mathbf{z}_{ni}^{\mathbf{x}} \mathbf{z}_{ni}^{\mathbf{x}\,T} \Big)^{-1},$$

where $q_{ni}^{\mathbf{x}} = \dfrac{c_n^x q_{ni}}{\sum_{\tilde{n}=1}^{N} c_n^x q_{\tilde{n}i}}$, $\mathbf{z}_i^{\mathbf{x}} = \sum_{n=1}^{N} q_{ni}^{\mathbf{x}} \mathbf{z}_{ni}$, $\mathbf{x}_i^{\mathbf{x}} = \sum_{n=1}^{N} q_{ni}^{\mathbf{x}} \mathbf{x}_{ni}$

$$\boldsymbol{\mu}_i^{\mathbf{x}} = \sum_{n=1}^{N} q_{ni}^{\mathbf{x}} \mathbf{x}_n - \mathbf{\Lambda}_i^{\mathbf{x}} \mathbf{z}_i^{\mathbf{x}}$$

$$[\mathbf{\Psi}_i^{\mathbf{x}}]_{jj} = \sum_{n=1}^{N} q_{ni}^{\mathbf{x}} \Big[(\mathbf{x}_{ni} - \mathbf{\Lambda}_i^{\mathbf{x}} \mathbf{z}_{ni})_j^2 + (\mathbf{\Lambda}_i^{\mathbf{x}} \mathbf{\Sigma}_n \mathbf{\Lambda}_i^{\mathbf{x}T})_{jj} \Big]$$

component velocity in the low dimensional coordinate space as

$$\langle \dot{\mathbf{z}} \rangle_i = \frac{1}{T} \sum_{n=1}^{N} \tilde{q}_{ni} (\mathbf{z}_n - \mathbf{z}_{n-1}). \tag{6.7}$$

Each model component covariance can then be expressed as

$$\boldsymbol{\Sigma}_{\mathbf{z}_{(i)}} = \begin{bmatrix} \langle \mathbf{z}\mathbf{z}^T \rangle_i & \langle \mathbf{z}\dot{\mathbf{z}}^T \rangle_i \\ \langle \dot{\mathbf{z}}\mathbf{z}^T \rangle_i & \langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}_{(i)}} & \boldsymbol{\Sigma}_{\mathbf{z}\dot{\mathbf{z}}_{(i)}} \\ \boldsymbol{\Sigma}_{\dot{\mathbf{z}}\mathbf{z}_{(i)}} & \boldsymbol{\Sigma}_{\dot{\mathbf{z}}\dot{\mathbf{z}}_{(i)}} \end{bmatrix} \tag{6.8}$$

where

$$\boldsymbol{\Sigma}_{\mathbf{z}\dot{\mathbf{z}}_{(i)}} = \left\langle \mathbf{z}\dot{\mathbf{z}}^T \right\rangle_i = \sum_{n=1}^{N} \tilde{q}_{ni} (\mathbf{z}_n - \boldsymbol{\kappa}_i)(\dot{\mathbf{z}}_n - \langle \dot{\mathbf{z}} \rangle_i)^T,$$

$$\boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}_{(i)}} = \left\langle \mathbf{z}\mathbf{z}^T \right\rangle_i = \boldsymbol{\Sigma}_i. \tag{6.9}$$

Therefore an equivalent non-linear dynamical system to Equation (5.5) is expressed in the low dimensional manifold coordinates as

$$\dot{\mathbf{z}} = \sum_{i=1}^{K} \alpha_i (\langle \dot{\mathbf{z}} \rangle_i + \boldsymbol{\Sigma}_{\dot{\mathbf{z}}\mathbf{z}_{(i)}} (\boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}_{(i)}})^{-1} (\mathbf{z} - \boldsymbol{\kappa}_i)). \tag{6.10}$$

6.2.1 Estimating Model Complexity

An advantage of the CFA model over a GMM with full covariance matrices is that it significantly reduces the number of parameters to estimate. In a GMM, $K(1 + D + \frac{1}{2}D(D+1))$ parameters are required (the priors π, mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ are of size 1, D, $\frac{1}{2}D(D+1)$ respectively). For CFA, however, $K(1 + 2D + d(D + \frac{1}{2}(d+3)))$ parameters are learned (the sizes of π, $\boldsymbol{\mu}$, $\boldsymbol{\Psi}$, $\boldsymbol{\Lambda}$, $\boldsymbol{\kappa}$, $\boldsymbol{\Sigma}$ are 1, D, D, dD, d, $\frac{1}{2}d(d+1)$ respectively). Although the major contributor is the dimension of the data, a model with a high number of components typically achieves a higher likelihood without introducing a penalty for model complexity. Penalizing models with uncoordinated

components naturally reduces the space of feasible solutions, however, we additionally incorporate a model complexity penalty in the form of the Bayesian Information Criterion (BIC).

A set of heuristics are used to manage model selection, which bear similarity heurisitics in [96]. A list of components is created, sorted by the inner summation of $\mathcal{F}$, and each component on the list is tested for a possible split. After each split the algorithm optimises the parameters until any of the component responsibilities over the data are not changing more than a minimum tolerance value. This 'reponsibility flux' is computed from a normalised difference between the responsibility before and after the last update for each component. The algorithm terminates once all components are checked.

To split a candidate component a sample is taken from the distribution and treated as a direction vector to partition the data belonging to the component into two new components. These groups form new responsibilities for the two components and the parameters for the partitions are subsequently found through an iteration of the M-step. If the split increases $\mathcal{F}$ a new list is generated, again ordered with $\mathcal{F}$, and the algorithm continues, otherwise it restores the previous model and tries to split the next component.

To remove a component it must be responsible for ≤ 2 data points ($\sum_{n=1}^{N} \tilde{q}_{ns}$), or have a noise variance, Ψ, below a minimum threshold, which corresponds to converging to a single data point, or have a significantly large ratio between two singular values in the latent dimensional space. The process is summarized in Algorithm (3).

Algorithm 3: Model Selection CFA

```
list = {1}, j = 0, K = 1
while j < length(list) do
    Split Component list(j + 1)
    while responsibility flux > tolerance do
        Compute E-step
        Run Forward-Backward algorithm
        Remove degenerate components
        Compute M-step
    end
    if F̃ > F then
        Reject split
        j = j + 1
    else
        Accept split
        Generate new list
        j = 0
    end
end
```

6.3 Robot Correspondence Mapping

The target 19DOF humanoid robot has 6DOF in each leg, 3DOF for each arm and 1DOF for the torso and the kinematic structure is defined in Appendix B. In order to learn a correspondence mapping between the human and robot we must first obtain a subset of corresponding data points, $\mathbf{y}$, for the robot using a motion retargeting scheme. The correspondence between the datasets can then be indicated by $c^{\mathbf{x}}$ and $c^{\mathbf{y}}$ as in the previous section. The human kinematic structure consists of a set of joint position vectors for 28 joints and end-points as labelled in Figure 6.3a. The target posture is first represented as a set of key position and orientation vectors which have analogues in the robot structure. These include the positions of the center of the torso, elbow joints, hands, knee joints and feet, and the orientations normal to the palms of the hands and to the heel of the foot, as displayed by the vectors in Figure 6.3.

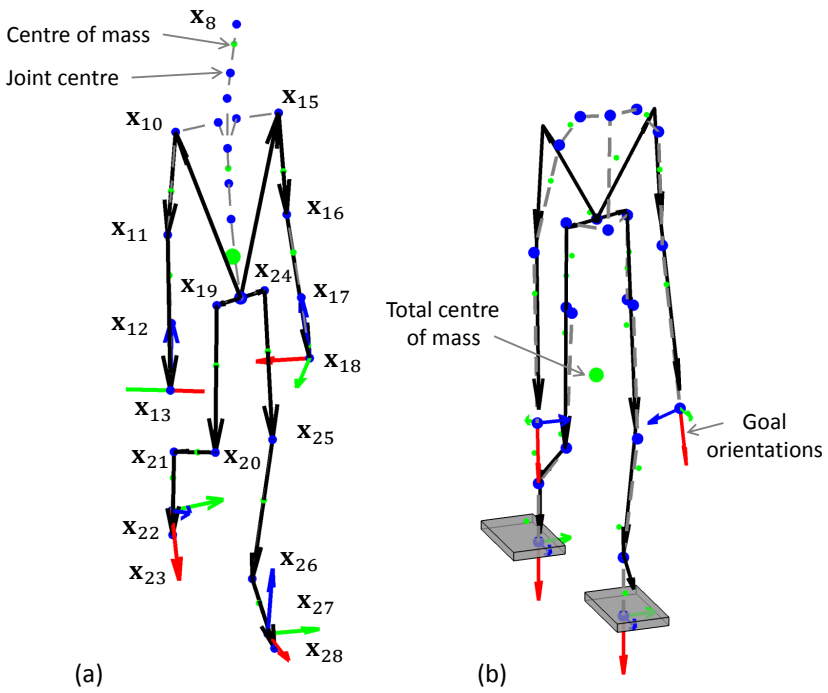

Figure 6.3: Depiction of the scaled positional targets (black vectors) for the humanoid robot, shown in relation to the human coordinate system, (a), and robot coordinates, (b). The joints of each body are shown in blue, the links are dashed gray lines, the link center of mass for the robot is shown in green and total center of mass is shown by a larger marker.

Since the dimensions of the target robot are known, each position in the human model is scaled to the length of an equivalent position vector in the robot frame. As a consequence, the target robot positions are in proportion to the workspace of the robot. An inverse kinematics function, to return a set of joint angles that achieve the desired positions, is executed to obtain a matching robot posture. The function considers the smoothness of the joint angle trajectory and avoids the physical constraints of the robot.

6.4 Robot Constraints

We use a task prioritized control [138] to find a trajectory that satisfies motor constraints by projecting the desired trajectory onto the nullspace of a constraint Jaco-

bian, $\mathbf{J_C}$, which considers the angle limits and avoids impending collision.

$$\hat{\dot{\mathbf{y}}} = \mathbf{J}^{\dagger}\dot{\mathbf{y}} + (\mathbf{I} - \mathbf{J}_{\mathbf{C}}^{\dagger}\mathbf{J_C})\dot{\mathbf{y}}_{des} \tag{6.11}$$

where $\hat{\dot{\mathbf{y}}}$ refers to the positions as opposed to the generalised coordinates of the robot, $\mathbf{y}$. The error between the set of target mapped human positions and the robot joint positions, as a function of the generalized coordinates, is minimized along with the velocity in joint space subject to angle limit and collision constraints of the robot.

Controlling a humanoid robot imposes additional constraints on the system. Since a humanoid robot is underactuated, having 6 unactuated DOF at the robot base, the mechanical stability of the frame further restricts the motion that can be safely imitated. The static stability of a robot posture is determined by the location of the body center of mass in relation to the ground contact forces whereas the dynamic stability is typically controlled by positioning the zero moment point (ZMP) within the ground contact plane.

To ensure that the robot maintains balance in relation to positioning of the ZMP, one can modulate the synthesized trajectory using the center of mass Jacobian with a similar prioritized control scheme [75]. This method translates the base frame and Jacobian to the contact frame, which would for instance be a point on the robot foot plane in the single contact phase, in order to control the position of CoM and ZMP from a static frame. Combining manifold control with the stability algorithm will modulate the robot posture along the manifold coordinates or in some cases depart from the manifold based on the satisfaction of the balance criteria.

Measurement in the body frame of reference and the contact forces is required to assess the structural stability. The orientation of the robot was estimated from an inertial measurement unit (IMU) constituting a tri-axial accelerometer and gyroscope. The sensor was mounted on the torso of the robot, as indicated in the kinematic

diagram in Figure 6.3b. An IMU is generally used to estimate an incline rather than a 3D orientation due to gravitational acceleration vector compensating for the gyroscope bias drift around horizontal axes. A Kalman filter was used to estimate the inclination [139] of the robot torso. This orientation was compared to the estimate derived from the ground contact plane, assuming all contact forces are acting in the horizontal plane. The sensing capability includes an orientation sensor fusing accelerometer and gyroscopic data with a Kalman filter as described in [139], and a set of 4 force sensing resistors on the base of each foot plate. A thin cushioning is taped over the sensor so that it slightly protrudes from the plate and makes contact with the surface and a threshold on the sensor value is obtained experimentally. The convex hull of the deteceted contact points is then identified as the stability region.

6.5 Results

In this section, we describe the results of applying this technique to data sets collected by a motion capture system and reconstructed in a humanoid robot. The data sets include a walking gait from a single person, multiple walking styles, and a motion involving stepping forward and reaching with both arms to an object.

6.5.1 Parameter Initialization

Learning the coordination of the model can be highly sensitive to the initialization of $\mathbf{z}$ so, as suggested in [83], we use Local Linear Embedding (LLE) [131] to initialize $\mathbf{z}$ and the modification outlined in [140] when we have a corresponding data set of robot retargeted points. This technique only requires the selection of the number of nearest neighbours to form a connective grid and this was set to 100 data points. The results were sensitive to this neighbourhood size but initialisations between approximately 50 and 200 obtained similar results.

For the purposes of illustration we selected $d = 2$, however, an intrinsic dimensionality can be estimated by further applying variational Bayesian technology [96].

6.5.2 Single walk data set

Applying Algorithm (3) to the gait data from a single subject resulted in a model of 23 components with a log-likelihood and reconstruction error across algorithm epochs shown in Figure 6.4a and 6.4b. The number of components in the best model are shown in Figure 6.4c and the corresponding number of model parameters in Figure 6.4d. Optimizing over $\mathcal{F}$ leads to a restriction on the number of model components and a reduction in the error. The reconstruction error for the model was 30% of the error when applying PCA on average (27% on robot data, 33% on human joint angles) and the number of model parameters is approximately one order of magnitude lower than when directly applying GMM. Figure 6.5 displays the time series trajectory of the data as projected into the latent coordinate system together with the simulated trajectory given by Equation (6.10). As a reference, the human posture and corresponding robot pose from Section 6.3 are shown above the signals. Although the trajectory matches the demonstrations, the velocity of the system can be modified to either speed up or slow down the robot motion. The model generates 'similar' motion for 0.2-3 times the original velocity, which may be of use in other reproduction tasks, but varying the velocity to this degree would likely destabilize an under-actuated humanoid robot.

One can compare the structure of latent space as derived from CFA to the eigenvectors when applying PCA by projecting the data into the corresponding spaces. In Figure 6.6a the walking data is projected along the two eigenvectors associated with the two largest eigenvalues. Compared to Figure 6.6b showing the CFA latent points, the PCA trajectory is highly non-linear (overlaps and re-traces) in two regions which

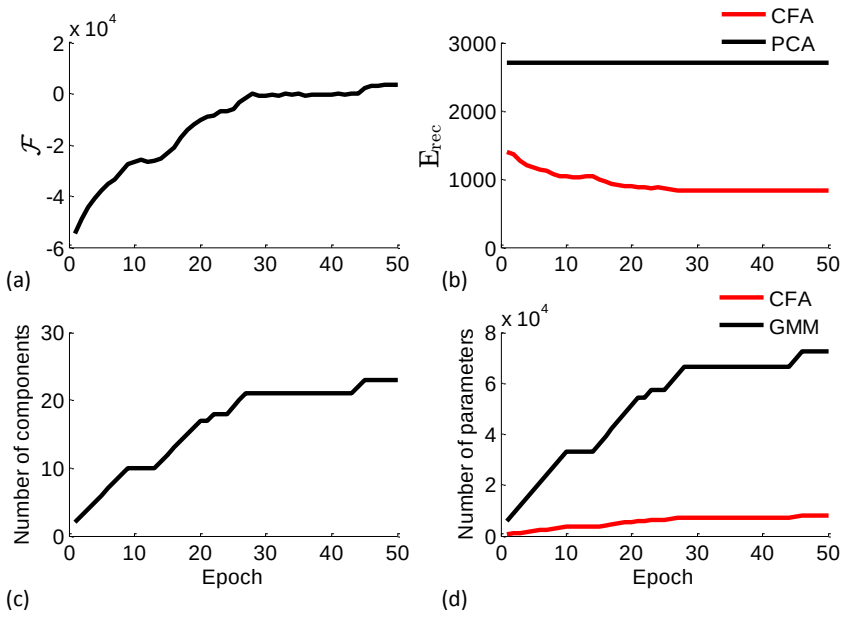

Figure 6.4: (a) Lower-bounded approximate log-likelihood across epochs. (b) The number of components in the best model after each epoch. (c) Reconstruction error evaluated after each epoch ($E_{rec} = \|\mathbf{x} - f_{\mathbf{z}\to\mathbf{x}}(f_{\mathbf{x}\to\mathbf{z}}(\mathbf{x}))\|$).

can affect the stability of the subsequent simulation obtained from learning a GMM. In Figure 6.6a and b the simulation of the system is shown in red. The model parameters, covariance $\mathbf{\Sigma_{zz}}$ and mean $\boldsymbol{\kappa}$, are depicted as ellipsoids and the velocity $\langle \dot{\mathbf{z}} \rangle_k$ is the vector associated each ellipsoid as was explained in Figure 5.2. The reconstruction error for each model in Figure 6.6, which can be calculated for all data points, is shown in Figure 6.7, partitioned into groups of joints. In total, E_{rec} using CFA resulted in 30% of the error level of PCA and according to Figure 6.7 CFA models the trajectory of the leg angles significantly more accurately than PCA.

Evidently, from Figure 6.6 the system follows a periodic orbit trajectory throughout $\mathbf{z}$ when initialized at a data point from the demonstrations. If the system deviates from this trajectory or is started in an unlikely position in the latent space (or a unobserved posture) the control law in Section 5.3.1 returns the system to the intended trajectory. To test the system we initialized the dynamic model in a number of lo-

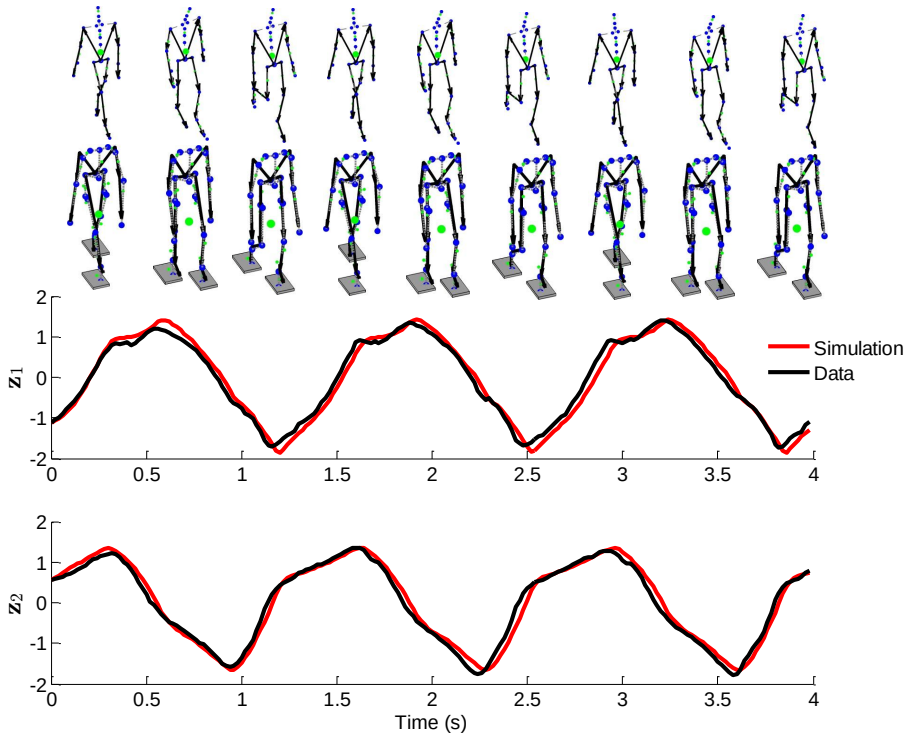

Figure 6.5: The postural trajectory as represented in the low dimensional space, $\mathbf{z}$. The data projected into $\mathbf{z}$ is compared to the trajectory from simulation according to Equation (6.10).

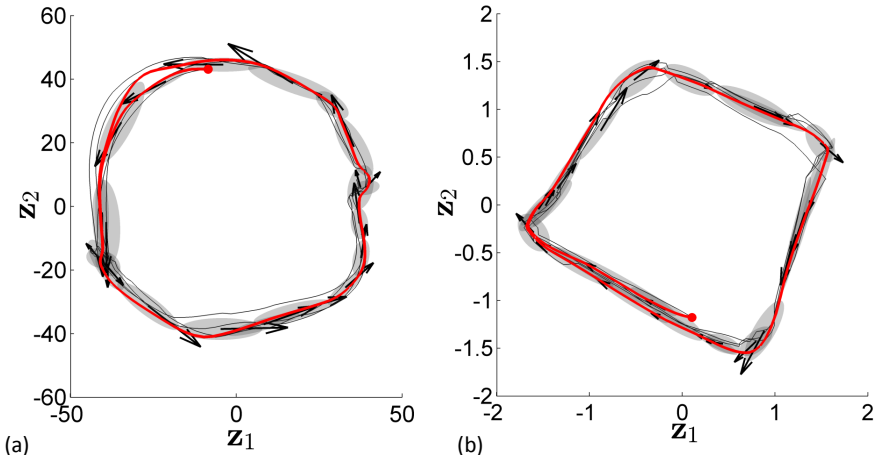

Figure 6.6: The observed demonstrations mapped to the latent space $\mathbf{z}$ for PCA (a) and CFA (b). The learned model parameters of the mixture model where each ellipsoid corresponds to $\boldsymbol{\Sigma}_{\mathbf{zz}}$, each vector represents $\boldsymbol{\mu}_{\dot{\mathbf{z}}}$, and the red line is an example simulation.

cations in Figure 6.8. Each subsequent trajectory returned smoothly to the periodic orbit. Due to the availability of explicit mapping functions in this model we can sam-

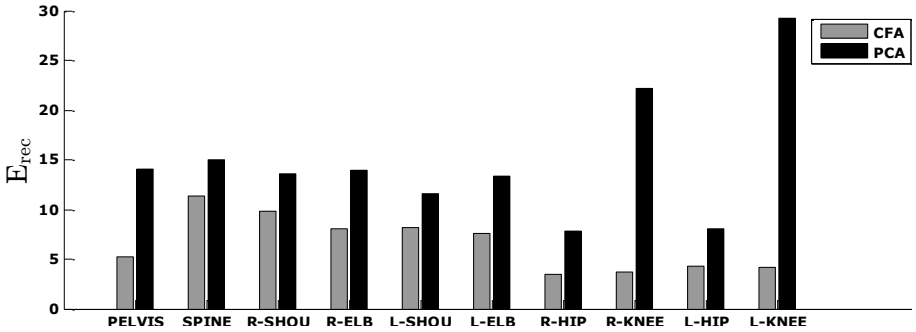

Figure 6.7: Average reconstruction error separated into groups of joints. The spine includes all angles from the sacrum to the neck, L-SHOU indicates the left shoulder, L-ELB is the left elbow. The error is displayed as a percentage of the average range of motion for the joints, therefore a value of 10% for a joint that exhibited a range of 30° averaged an error of 3°

ple postures throughout the latent variable space and inspect how well they correlate with the data points as human postures. In Figure 6.8 we display the corresponding posture for a number of locations in the latent variable space. It is clear that the variance between postures along the line perpendicular to the trajectory are very similar, yet locations far from the demonstrations may vary significantly in posture, even to the extent of violating physical constraints of the body.

Figure 6.9a depicts the region $\mathcal{D}$, the contour of minimum data likelihood, along with an infeasibility region, $\mathcal{C}$ (in red), constructed from the anatomical bounds of the human body. The regions intersect in two stages of the gait cycle, when the knee angle of either leg is at full extension, due to the extrapolation of the component eigenvectors. Moreover, examining the robot angle limits in relation to the manifold, in Figure 6.9b reveals additional regions of intersection due to constraints on the robot hip angles despite taking into account the variance Ψ around the manifold. It is necessary to revise the set $\mathcal{D}$ to exclude $\mathcal{C}$ to avoid sampling infeasible points and to ensure that any samples depart sufficiently from the manifold in order to satisfy the constraints. This is handled by projecting the trajectory onto the nullspace of the constraint Jacobian as mentioned in Section 6.4.

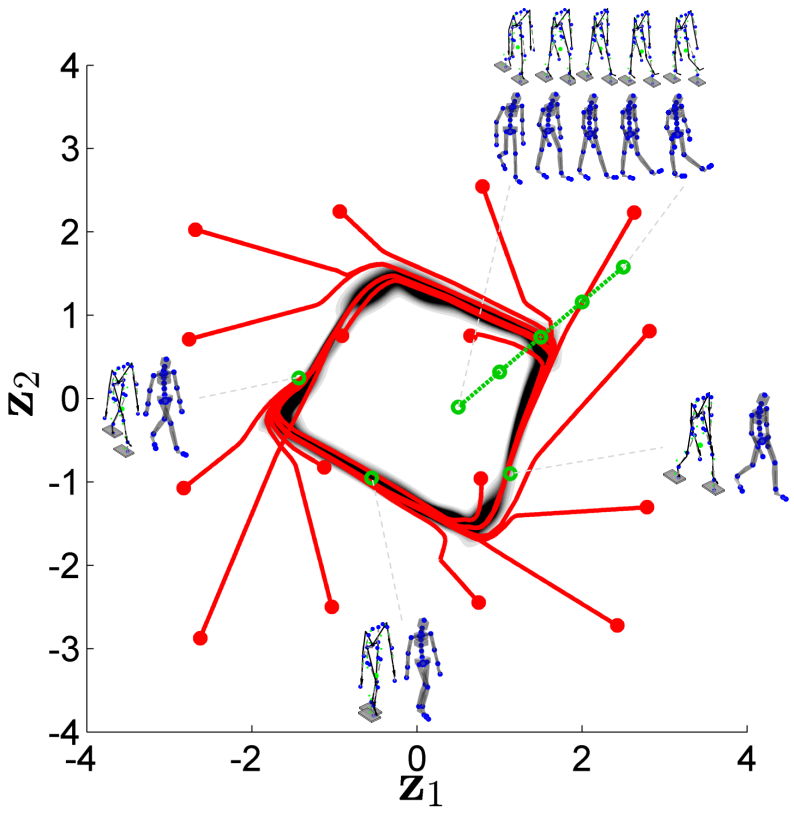

Figure 6.8: Response of the system learned from the walking cycle. Multiple initializations (circles) of the system are tested and each trajectory (red) converges to a periodic orbit. The likelihood of observing a data point is visualized by the grayscale contours. A number of samples (green) from the latent space are reconstructed in the data space by displaying the expected pose of the human and robot. The samples along the green line extend into regions of low likelihood and consequently deviate from the observed gait. The top right sample depicts an over extended knee angle, an infeasible posture.

6.5.3 Multiple walking styles

Demonstrating one style of gait may be sufficient to train a robot, however, observing a greater variety of motion improves the flexibility of the robot. Walking at different speeds typically involves adapting the stride length and arm swing range rather than simply scaling the velocity of a standard trajectory. We combined two styles of walking in the training set including the data from the previous section and a slower gait of another 16 cycles. This produced a model of 16 components as is shown in Figure 6.10, where the minimum likelihood region $\mathcal{D}$ is shown along with the combined

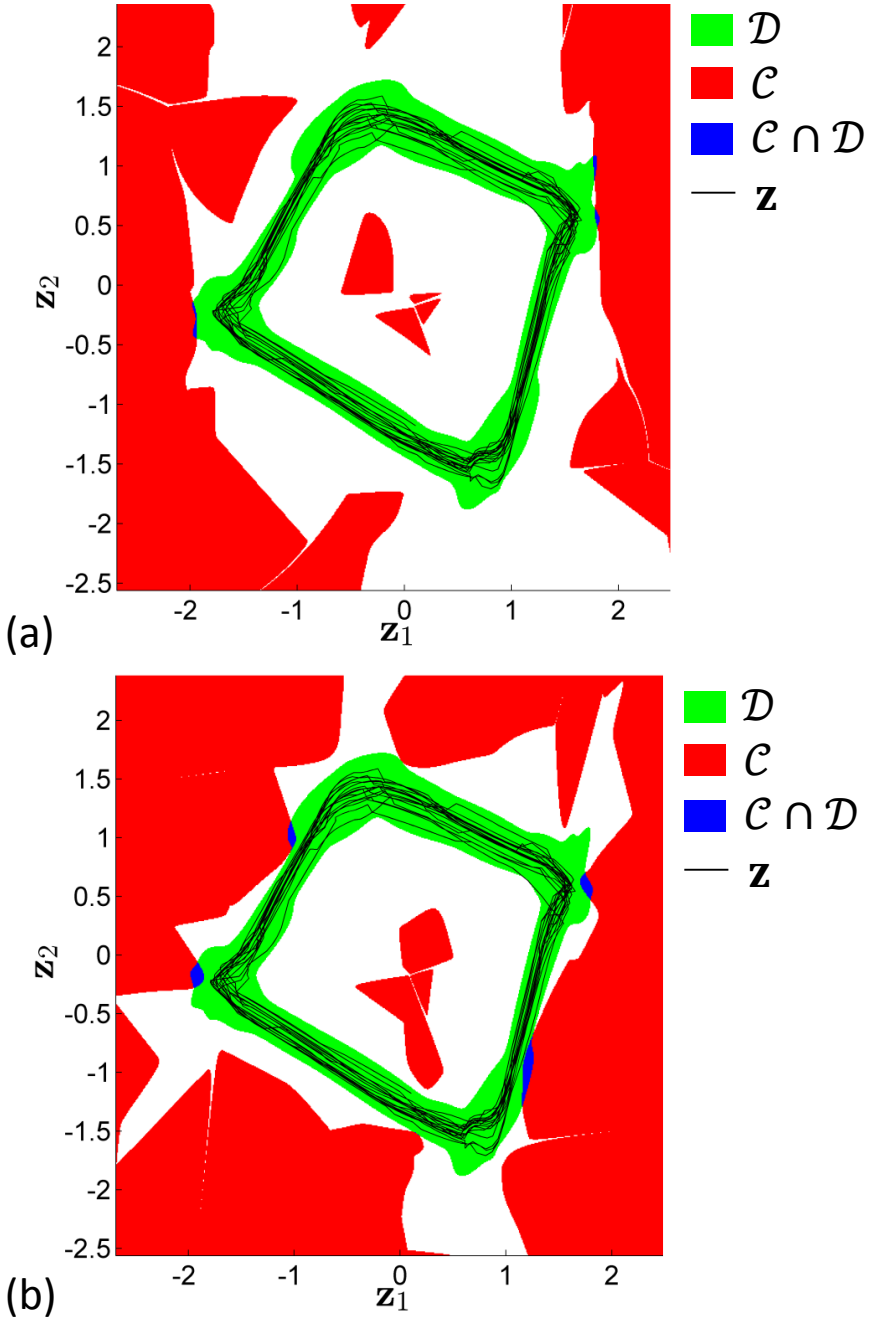

Figure 6.9: The region of likelihood greater than the minimum data likelihood, $\mathcal{D}$, shown along with the data points in latent space and the projected region of infeasible postures based on angle limits, $\mathcal{C}$. The intersection of these regions heralds a departure from the manifold. In (a), the infeasibility constraints shown are derived from anatomical limits on human posture, whereas in (b), the constraints are derived from robot angle limits.

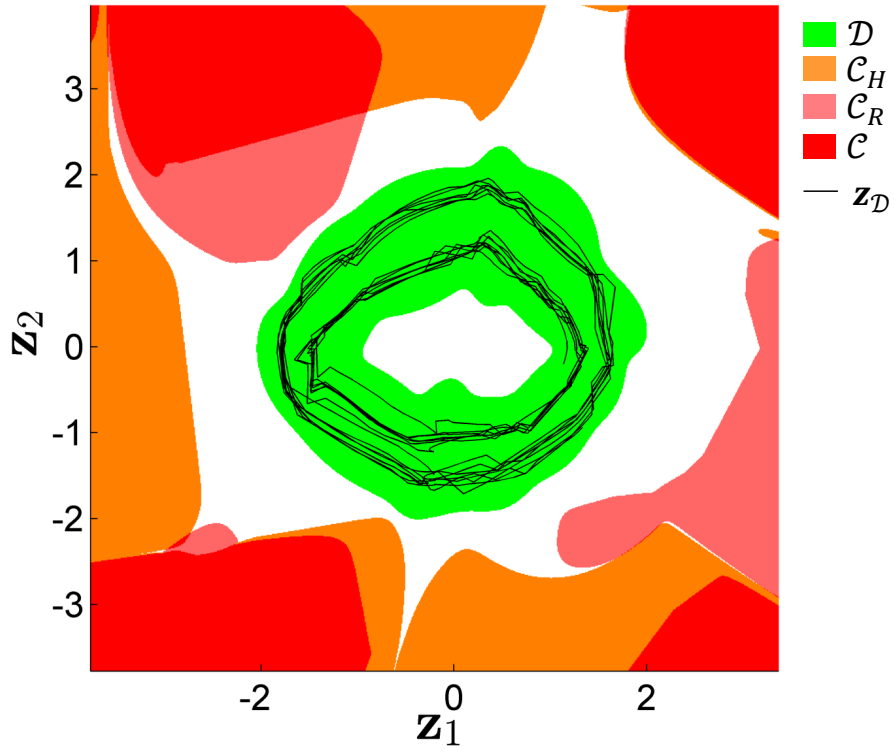

Figure 6.10: Low dimensional manifold coordinates of model learned with multiple gait styles. The region of likelihood greater than the minimum data likelihood, $\mathcal{D}$, shown along with the data points in latent space and the projected region of infeasible postures based on human anatomical limits, $\mathcal{C}_H$, robot angle limits, $\mathcal{C}_R$, and the intersection of the limits, $\mathcal{C}$. The trajectory of observations, $\mathbf{z}_t$, is shown in black.

constraints of the human and robot motion. Including the additional observations increases the likelihood of points perpendicular to the trajectory meaning a greater postural variance is accessible to the system. The projected data points form two cycles within $\mathcal{D}$ and correspond to the two walking speeds. It also reduces the encroachment of constraints, in this case $\mathcal{C}_H$ for the human and $\mathcal{C}_R$ for the robot, into the region of data likelihood as was seen in the previous model. This is due to a reduction in scale of the factor loading eigenvalues as influenced by observing a slower walk. Figure 6.11 displays simulations of the resultant system which exhibit the intended periodic trajectory and samples from this space which demonstrate the range of motion captured by the model. In practice, a robot will need to transition between

walking and related behaviours so expanding the range of feasible postures represented by a walking model presents greater opportunities for smoothly transitioning to a stopping or grasping action when the goal location is achieved.

The simulated trajectory was transferred to the robot resulting in the motion captured in Figure 6.12. Due to limitations in communication speed of the test robot we did not utilise the real-time balance control features and controlled the robot directly from the manifold dynamics. The robot gait trajectory was only unstable on one part of the cycle when a foot is lifted and matched with the simulation.

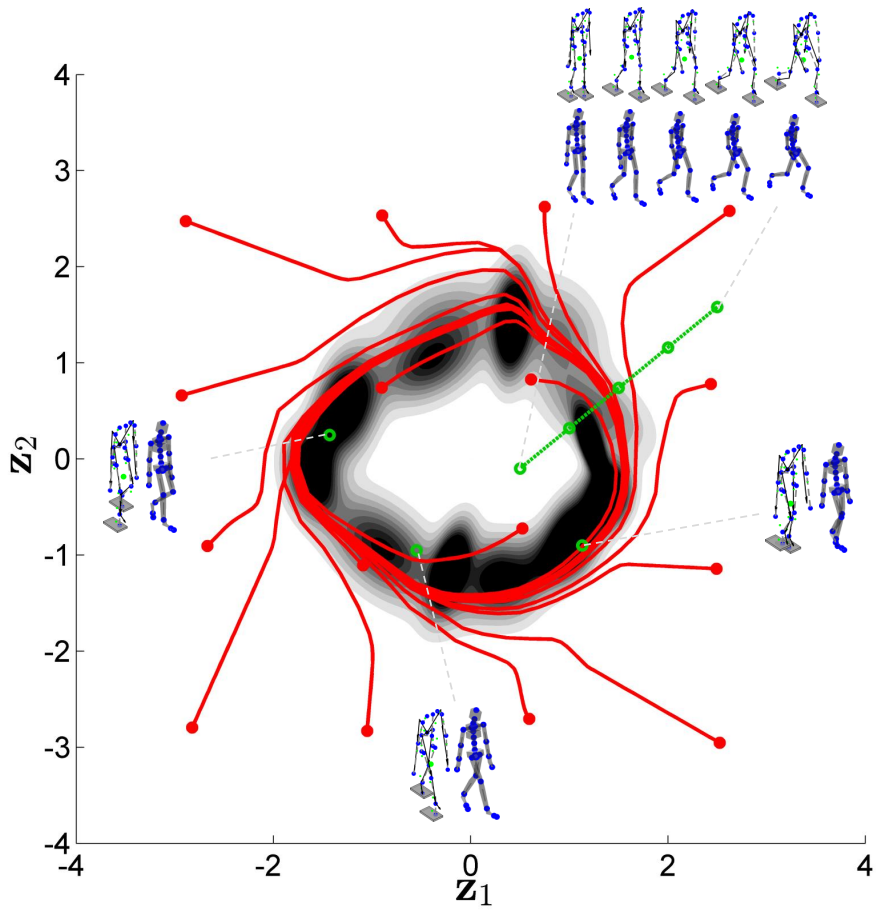

Figure 6.11: The likelihood of the model learned from two walking data sets is shown here in the low dimensional coordinates. A series of samples are drawn from the model latent space and shown as reconstructed in the data space by displaying the corresponding the pose of the human body. The contours indicate confidence levels of the model and the red trajectory is the converging periodic orbit initialised from a number of starting locations.

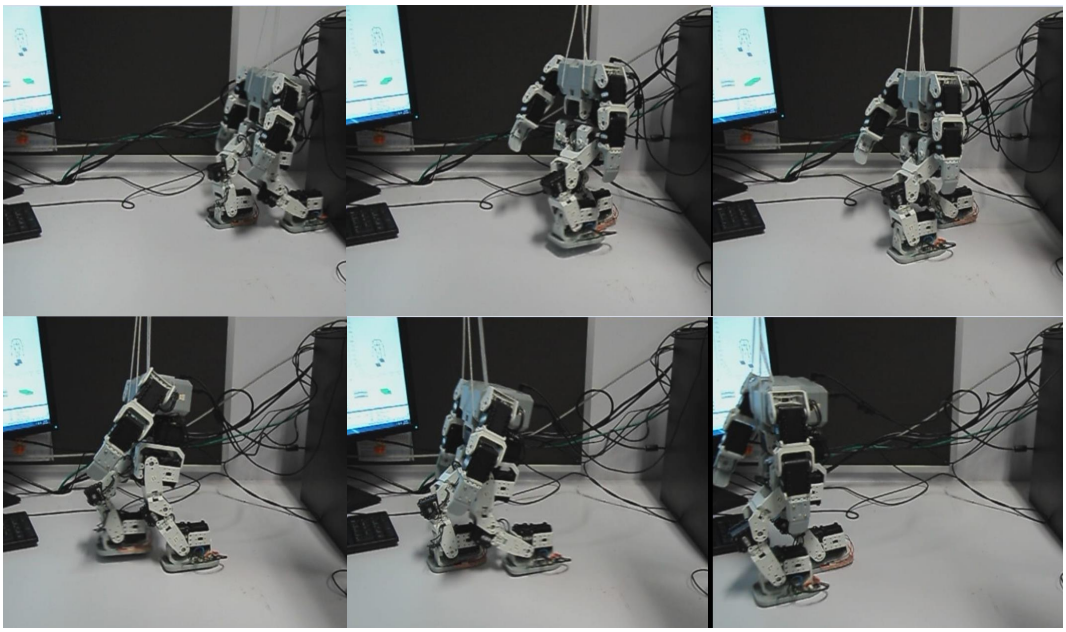

Figure 6.12: An experiment to control a 19-DOF robot using the proposed model. The gait was generated from the model shown in Figure 6.11

6.5.4 Transition between manifolds

The final experiment involves a candidate demonstration for transitioning between a walking cycle and interacting with the environment. This action consists of taking a step forward and reaching out to an object at rest on a table. One difference in the step phase is that the arms were already raised. Six demonstrations were recorded that were independent of the walking cycle recordings. The learned model comprised of 13 components and exhibits a variance as shown in Figure 6.13a by the likelihood contours and samples of human and robot postures. A large variance is present in the final reaching stage of the motion since there are multiple ways of reaching for an object if the position varies between demonstrations. Further environmental

constraints based on sensing the object location need to be considered to provide a final target pose.

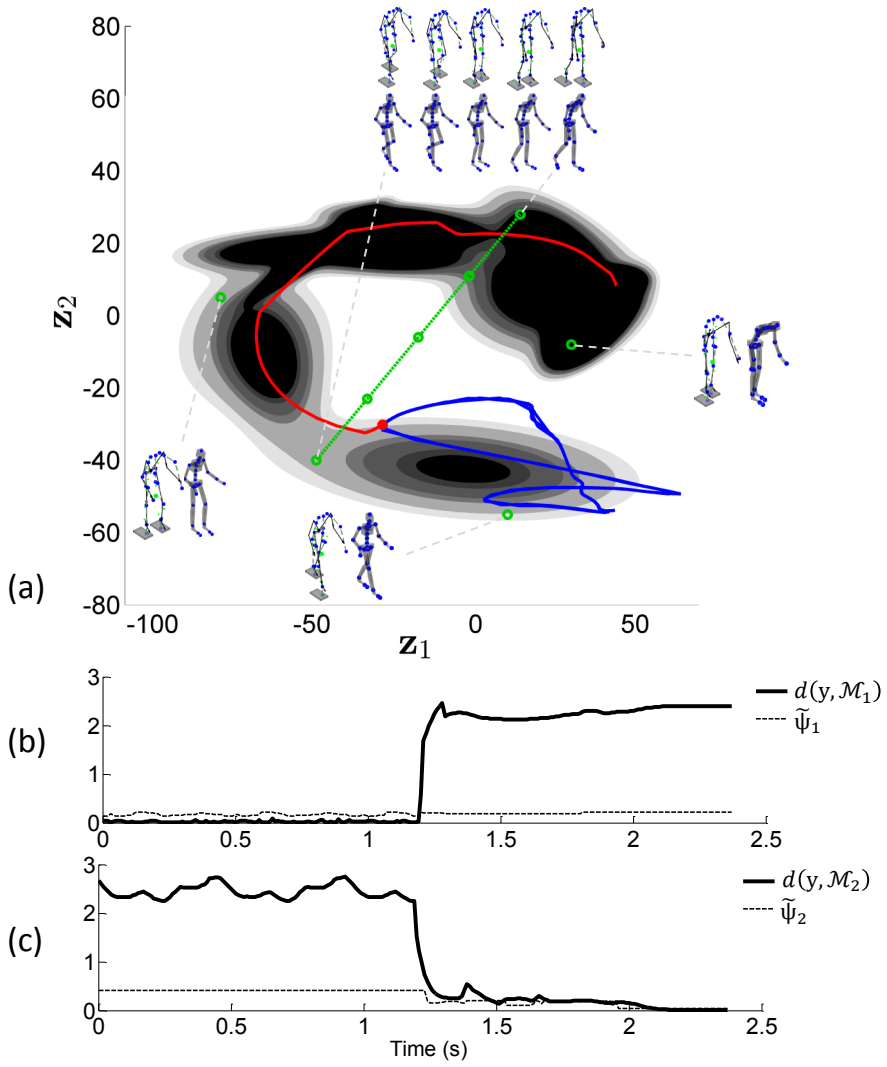

Figure 6.13: Simulation of a transition between two models where one is a set of walking cycles, $\mathcal{M}_1$, and the other a reaching task, $\mathcal{M}_2$. (a) Initial motion is governed by $\mathcal{M}_1$ from Figure 6.11 and is projected onto $\mathcal{M}_2$ (path in blue). After switching to the reaching model the trajectory is shown in red. A series of samples are also drawn from $\mathcal{M}_2$ and shown as reconstructed in the data space by displaying the corresponding pose of the human and robot. The contours indicate confidence levels of the model. (b) The distance from the current state of the robot and the expected value on the manifold learned from walking. (c) Equivalent distances from current state to manifold $\mathcal{M}_2$.

For the transition between two manifolds, $\mathcal{M}_1$ and $\mathcal{M}_2$, one must consider the distance between the current system state in data space and the expected state if pro-

jected onto the target manifold. Let $\mathbf{y}$ be the current state, $\tilde{\mathbf{y}}_{\mathcal{M}_j} = \langle p(\mathbf{y}|\langle p(\mathbf{z}|\mathbf{y})\rangle)\rangle_{\mathcal{M}_j}$ is the expected value according to the dynamic system driven by $\mathcal{M}_j$. Two methods of quantifying the distance to a manifold and were considered. Computing the likelihood of the data point given the target manifold and Euclidean distance to the expected value on the manifold. The Euclidean distance to a particular manifold j will therefore be defined as $d(\mathbf{y}, \mathcal{M}_j) = \|\mathbf{y} - \tilde{\mathbf{y}}_{\mathcal{M}_j}\|$ and the manifold variance is given by $\tilde{\mathbf{\Psi}}_j = \sum_s \alpha_s \|\mathbf{\Psi}_s\|$, for $\mathcal{M}_j$.

We considered a transition from the walking model in Figure 6.11 to the model derived from the reaching demonstrations. Initially the system is governed by the walking periodic orbit ($\mathcal{M}_1$) and when projected onto the reaching model latent space ($\mathcal{M}_2$) it follows the blue trajectory in Figure 6.13a. The corresponding distances to each manifold are displayed in Figure 6.13b and Figure 6.13c. Initially, during the walking motion, $d(\mathbf{y}, \mathcal{M}_1) < \tilde{\mathbf{\Psi}}_1$, but $d(\mathbf{y}, \mathcal{M}_2) > \tilde{\mathbf{\Psi}}_2$, which indicates there are no postures shared by both manifolds. However, we trigger the transition on the minimum of $d(\mathbf{y}, \mathcal{M}_2)$ throughout the walking cycle to produce a smoother transition. The remaining trajectory is then driven by $\mathcal{M}_2$ and corresponds to the red trajectory in Figure 6.13a. Using the manifold likelihood to quantify the divergence from the manifold was unsuccessful and led to a switching posture which caused significant pause in the motion.

6.6 Chapter Summary

In this chapter we have proposed an extension to the non-linear dynamical system control with mixture models by learning a underlying manifold which unifies the mixture model components along the their major eigenvectors. The model is constructed in such a way that the system dynamics act along the manifold coordinates rather than directly in the original data coordinate system. Given retargeted data points for

a robot a correspondence mapping is learned by constraining the shared points from both domains to map to the same locations on the manifold. Physical constraints of the robot embodiment and environment can be transformed into the manifold coordinates to influence the dynamical system. The method was tested by learning a model based upon human motion capture data and synthesising trajectory from the manifold dynamics to control a humanoid robot. The robot imitated a human walking gait and a transition from walking to stopping and reaching for an object.

Chapter 7

Conclusions and Recommendations of Future Work

7.1 Introduction

The complexity involved in the deployment of control skills in robots is a common barrier to the pervasiveness of robots in society today. This thesis explores numerous challenges in developing learning algorithms for imitative robots such that intuitive and interactive programming techniques can be widely adopted.

Imitative motion primitive control algorithms center around constructing observational models of actions that have competitive recognition capacity and accurate reproducibility. This approach of robot programming by demonstration, which aims to accelerate robot learning, is often dependent on segmenting each of the demonstrations in advance. In Chapter 3 an analysis was presented on the sensitivity of automatic segmentation of human movement. The investigation compared the clinical practices of detecting biomechanical changes in repeated motion evaluations to the outcomes of a segmentation algorithm and are shown to be sensitive to shifts in joint angle profiles that correspond to accessory muscle activations. Further to this

study, an algorithm was investigated for the clustering of time series segments. A collection motion capture action categories were then segmented and clustered using motion primitive components in Chapter 4. The resultant clusters of time series were matched based on dynamic programming distance metrics. Additional analysis on the recognition performance involved cross-validation across different demonstrators.

Reproducing trajectories from the motion primitive model presents various challenges of stability. The maximum likelihood algorithm for learning the model parameters is insensitive to these considerations and in Chapter 5 a method for producing stable periodic or non-periodic trajectories was proposed. Trajectories were shown to converge in an experiment involving the demonstration of a series of two dimensional paths. This motion reproduction method, however, is often sensitive to the dimensionality of the model, especially when extended to the control of a humanoid robot. An extension was proposed to the non-linear dynamical system motion primitive model by assuming that the motion is locally linear and resides on a manifold. This framework is also extended to address the correspondence problem whereby joint space representations for both the human and robot map to the same low dimensional coordinates. By generating trajectories from a stable dynamical system acting along the manifold, learned from observing demonstrations, we are able to endow a humanoid robot with control skills.

The contributions in this thesis include the automatic segmentation and clustering of time series segments, the extension of learning non-linear system controllers of into hidden Markov models, modifying the trajectory online in response to predicted critical collisions and physical constraints, and extending the motion primitive dynamical system controllers into high degree of freedom, redundant robots using dimensionality reduction. These contributions further advance the field of human-robot interaction and imitative robots towards the goal of fully autonomous learning robots guided by

human instruction. Clearly, there are still important areas to research in order to reach this goal. The rest of this chapter is devoted to outlining potential research directions based on the content of this thesis.

7.2 Future Work Recommendations

Investigations in this area of research could follow a number of directions. Some of the plausible avenues include the following:

- In conjunction with the dynamical system acting on the manifold in Chapter 6 it is important for a humanoid to maintain balance control. Full body balance control of the humanoid robot can be achieved by using the nullspace of the center of mass Jacobian, J_{com}, to position the ZMP [75]. This would allow one to constrain all motion generated from the manifold to conform within bounds for center of mass and ZMP to create a robot that could learn and reproduce dynamic movement skills whilst maintaining balance control.

- By placing flexible prior distributions over the model parameters in CFA one can perform variational Bayesian learning to obtain an informative distribution over the parameters. This enables an estimate for the number of components, K, and to estimate the intrinsic dimensionality, d, with automatic relevancy determination (ARD) by optimising a set of hyper-parameters.

- Incorporating the non-linear dynamic system stability constraints into the model learning algorithm would circumvent the necessity for an additional compensatory stage to stabilise the system. This could be rectified by a carefully designed prior probability distribution over the covariance matrices or by investigating the application of transverse contraction analysis [129] to this design challenge.

- Additional optimisation could prepare the system for transitioning between manifolds in efficient manner. Regions on each manifold could be coordinated such that a continuous path on the model coordinates connects the closest components in the each of the models. This is achievable due to the flexibility introduced by the auxiliary variables and would then simplify merging models in an incremental manner. The system controlling transitions would be dependent on a more abstract description of the robot task goals which may be composed of a transitioning between a sequence of motion primitive modules.

- Implementing a sensor suit system with new wireless sensor units would streamline the data gathering process in a decision support system for regular use in a clinic as indicated in the second case study of Chapter 3. In recording a larger database of lifting actions one could have a larger repository of example detrimental lifting postures, which is limited by the final postures of each evaluation. In the experiment conducted in this thesis there were clearly multiple ways of exhibiting fatigue which made it difficult to characterise a small cohort of subjects.

- An underlying fatigue status could inferred from observations of postural differences, heart rate and other clinical observations by proposing a probabilistic model which combines all of the observations for prediction. These predictions could then inform clinical decisions more directly in a functional capacity assessment setting or indicate a level of intervention for assistive robotics. Related models have been proposed in [112].

- Assessing the segmentation accuracy of a system should also ideally take into account the variance of the ground truth one compares against. Collecting additional observations on the class labels that could be assigned to recordings

would improve the reliability of the evaluation. However, this process can become complicated when there is a greater disagreement among observers over similarity of actions.

- Implementing observer state models would enable estimating contact forces with the environment while reproducing a demonstrator action. If the environment sensors are inaccurate and the robot contacts an unforeseen objects it would then be able to modulate the generated path based on the estimated force.

References

[1] Xsens Technologies. MVN User Manual and Technical Documentation, 2013. http://www.xsens.com/en/general/mvn. xi, 42, 43, 44, 115

[2] Andreas Buja, Deborah F. Swayne, Michael L. Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008. xiii, 76

[3] P. De Leva. Adjustments to zatsiorsky-seluyanov's segment inertia parameters. *Journal of Biomechanics*, 29(9):1223–1230, 1996. xviii, 44, 45, 79

[4] Andrew Isaac and Claude Sammut. Goal-directed learning to fly. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 258–265. AAAI Press, 2003. 1, 23

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 1371–1394. Springer Berlin Heidelberg, 2008. 1, 25

[6] Dorian Suc, Ivan Bratko, and Claude Sammut. Learning to fly simple and robust. In *European Conference on Machine Learning*, pages 407–418, 2004. 1, 12, 23

[7] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010. 1, 24

[8] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990. 2

[9] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1398–1403, 2002. 2, 18, 25

[10] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura. Embodied symbol emergence based on mimesis theory. *The International Journal Robotics Research*, 23(4-5):363–377, 2004. 2, 22, 24, 33

[11] G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: A neurophysiological study. *Experimental Brain Research*, 91:176–180, 1992. 3, 13

[12] Richard A. Schmidt. *Motor control and learning: A behavioural emphasis*. Human Kinetics Press, Champaign, IL, USA, 5 edition, 2011. 11

[13] D. Suc and I. Bratko. Skill modeling through symbolic reconstruction of operator's trajectories. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 30(6):617–624, 2000. 12, 23

[14] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, 1431:537–547, 2003. 12, 28, 33

[15] M. Field, Z. Pan, D. Stirling, and F. Naghdy. Human motion capture sensors and analysis in robotics. *Industrial Robot: An International Journal*, 38(2):163–171, 2011. 12

[16] J.M. Kilner and R.N. Lemon. What we know currently about mirror neurons. *Current Biology*, 23(23):1057 – 1062, 2013. 13

[17] E. Oztop, M. Kawato, and M.A. Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006. 13

[18] D. L. Wardman, J. L. Taylor, and R. C. Fitzpatrick. Effects of galvanic vestibular stimulation on human posture and perception while standing. *Journal of Physiology*, 551(3):1033–1042, September 2003. 14

[19] Paul Bach y Rita and Stephen W. Kercel. Sensory substitution and the humanmachine interface. *Trends in Cognitive Sciences*, 7(12):541 – 546, 2003. 14

[20] E. Marder and R. L. Calabrese. Principles of rhythmic motor pattern generation. *Physiological Reviews*, 76(3):687–717, 1996. 14

[21] Arthur D Kuo. The relative roles of feedforward and feedback in the control of rhythmic movements. *Motor control*, 6(2):129145, April 2002. 14

[22] J. Nassour, P. Henaff, F.B. Ouezdou, and G. Cheng. Multi-layered multi-pattern CPG for adaptive locomotion of humanoid robots. *Biological Cybernetics*, 108(3):291–303, 2014. 14

[23] Herman van der Kooij, Ron Jacobs, Bart Koopman, and Frans van der Helm. An adaptive model of sensory integration in a dynamic environment applied to human stance control. *Biological cybernetics*, 84(2):103–115, 2001. 14

[24] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, March 2010. 16

[25] R. Li, T. Tian, and S. Sclaroff. Divide, conquer and coordinate: Globally coordinated switching linear dynamical system. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):654–669, April 2012. 16, 29

[26] Ruei-Sung Lin, Che-Bin Liu, Ming-Hsuan Yang, Narendra Ahuja, and Stephen Levinson. Learning nonlinear manifolds from time series. In Ale Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision ECCV 2006*, volume 3952 of *Lecture Notes in Computer Science*, pages 245–256. Springer Berlin Heidelberg, 2006. 16

[27] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society. 16

[28] Jungong Han, Ling Shao, Dong Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, 43(5):1318–1334, Oct 2013. 16

[29] AG. Cutti, A Giovanardi, L. Rocchi, and A Davalli. A simple test to assess the static and dynamic accuracy of an inertial sensors system for human movement analysis. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 5912–5915, August 2006. 16

[30] D. Roetenberg, H.J. Luinge, C. T M Baten, and P.H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 13(3):395–405, Sept 2005. 16

[31] Ascension Technologies. MotionStar , 2014. http://www.ascension-tech.com/. 17

[32] J. Moldenhauer, I Boesnach, T. Beth, V. Wank, and K. Bos. Analysis of human motion for humanoid robots. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 311–316, April 2005. 17

[33] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović. Practical motion capture in everyday surroundings. *ACM Trans. Graph.*, 26(3), July 2007. 18

[34] M. Stikic, D. Larlus, S. Ebert, and B. Schiele. Weakly supervised recognition of daily life activities with wearable sensors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2521–2537, December 2011. 19, 22, 96

[35] J.A. Ward, P. Lukowicz, G. Troster, and T.E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1553–1567, October 2006. 19, 22, 90

[36] J. Barbic, A. Safonova, C. Pan, J. Faloutsos, J. K. Hodgins, and N.S. Pollard. Segmenting motion capture data into distinct behaviors. In *Graphics Interface*, pages 185–194, 2004. 19, 85

[37] J. Kohlmorgen and S. Lemm. A dynamic hmm for on-line segmentation of sequential data. In *Neural Information Processing Systems (NIPS)*, pages 793–800, 2001. 19

[38] D. Kulic and Y. Nakamura. Scaffolding on-line segmentation of full body human motion patterns. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2860–2866, September 2008. 20

[39] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Neural Information Processing Systems (NIPS)*, pages 457–464, 2008. 20, 67, 98

[40] S. Chiappa and J. Peters. Movement extraction by detecting dynamics switches and repetitions. In *Neural Information Processing Systems (NIPS)*, pages 388–396, 2010. 20

[41] D. Kulic, W. Takano, and Y. Nakamura. Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *The International Journal of Robotics Research*, 27(7):761–784, 2008. 20, 22, 24, 97

[42] Feng Zhou, F. De la Torre, and J.K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):582–596, 2013. 20, 38, 69

[43] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, April 2007. 20, 22, 24, 96

[44] S.M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *Robotics, IEEE Transactions on*, 27(5):943 –957, oct. 2011. 20, 25, 96, 102, 106, 109, 113

[45] A. Fod, M.J. Mataric, and O.C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, January 2002. 20, 79

[46] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006. 21, 22

[47] G.N. Pradhan and B. Prabhakaran. Indexing 3-d human motion repositories for content-based retrieval. *Information Technology in Biomedicine, IEEE Transactions on*, 13(5):802–809, September 2009. 21

[48] M. Müller, A. Baak, and H. Seidel. Efficient and robust annotation of motion capture data. In *Symposium on Computer Animation*, pages 17–26, 2009. 21

[49] K. Altun, B. Barshan, and O. Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010. 22

[50] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *UbiComp 2008: Ubiquitous Computing, 10th International Conference, Seoul, Korea, Proceedings*, pages 10–19, September 2008. 22

[51] C. Sun, D. Stirling, and F. Naghdy. Human behaviour recognition with segmented inertial data. In *Australasian Conference on Robotics and Automation (ACRA)*, December 2006. 22

[52] T. Abbas and B.A. MacDonald. Robust trajectory segmentation for programming by demonstration. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 1204–1209, Sept 2009. 22

[53] F. Meier, E. Theodorou, F. Stulp, and S. Schaal. Movement segmentation using a primitive library. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3407–3412, September 2011. 22

[54] S. Muench, J. Kreuziger, M. Kaiser, and R. Dillmann. Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *In Proceedings of the 24th International Symposium on Industrial Robots*, pages 685–693, 1994. 23

[55] A. Makarovic. A qualitative way of solving the pole balancing problem. In J. E. Hayes, D. Michie, and Ė. Tyugu, editors, *Machine Intelligence 12*, pages 241–258. Clarendon Press, New York, NY, USA, 1991. 23

[56] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15, Intelligent Agents [St. Catherine's College, Oxford, July 1995]*, pages 103–129, Oxford, UK, UK, 1999. Oxford University. 23

[57] G. M. Shiraz and Claude Sammut. Combining Knowledge Acquisition and Machine Learning to Control Dynamic Systems. In *International Joint Conference on Artificial Intelligence*, pages 908–913, 1997. 23

[58] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. 24

[59] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann. Imitation learning of dual-arm manipulation tasks in humanoid robots. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 40–47, Dec 2006. 24

[60] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi. Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):1039–1052, 2012. 24

[61] D. Kulic, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012. 24

[62] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, Xiao-Yu Fu, K. Goldberg, and P. Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2074–2081, 2010. 24

[63] M. Hersch, F. Guenter, S. Calinon, and A. Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *Robotics, IEEE Transactions on*, 24(6):1463–1467, Dec 2008. 25

[64] S.M. Khansari-Zadeh and A. Kronander, K. Billard. Learning to play minigolf: A dynamical system-based approach. *Advanced Robotics*, 26(17):1967–1993, 2012. 25, 106

[65] SeyedMohammad Khansari-Zadeh and Aude Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454, 2012. 25, 27, 107

[66] S.P. Chatzis, D. Korkinof, and Y. Demiris. A nonparametric bayesian approach toward robot learning by demonstration. *Robotics and Autonomous Systems*, 60(6):789–802, June 2012. 25

[67] V. Kruger, V. Tikhanoff, L. Natale, and G. Sandini. Imitation learning of non-linear point-to-point robot motions using dirichlet processes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2029–2034, May 2012. 25

[68] S. Calinon, F. D'halluin, D.G. Caldwell, and A.G. Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 582–588, Dec 2009. 25, 102

[69] P. Azad, T. Asfour, and R. Dillmann. Toward an unified representation for imitation of human motion on humanoids. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2558–2563, 2007. 26

[70] BEHZAD DARIUSH, MICHAEL GIENGER, ARJUN ARUMBAKKAM, YOUDING ZHU, BING JIAN, KIKUO FUJIMURA, and CHRISTIAN GOERICK. Online transfer of human motion to humanoids. *International Journal of Humanoid Robotics*, 06(02):265–289, 2009. 27

[71] Jur van den Berg, Pieter Abbeel, and Kenneth Y. Goldberg. LQG-MP: optimized path planning for robots with motion uncertainty and imperfect state information. *International Journal of Robotic Research*, 30(7):895–913, 2011. 27

[72] M. Vukobratovic and B. Borovac. Zero-moment point thirty five years of its life. *International Journal of Humanoid Robotics*, 01(01):157–173, 2004. 27, 46

[73] Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990. 27

[74] Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA*, pages 1404–1409, 2002. 27

[75] A. Gams, J. van den Kieboom, F. Dzeladini, A. Ude, and A.J. Ijspeert. Real-time full body motion imitation on the coman humanoid robot. *Robotica*, FirstView:1–13, 6 2014. 27, 132, 148

[76] D. Stonier and Jong-Hwan Kim. Zmp analysis for realisation of humanoid motion on complex topologies. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, volume 1, pages 247–252, 2006. 27

[77] Ian R. Manchester and Jack Umenberger. Real-time planning with primitives for dynamic walking over uneven terrain. *CoRR*, abs/1310.7062, 2013. 27

[78] Shinichiro Nakaoka, Atsushi Nakazawa, Kazuhito Yokoi, and Katsushi Ikeuchi. Leg motion primitives for a dancing humanoid robot. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pages 610–615, 2004. 28

[79] P. Kormushev, D.N. Nenchev, S. Calinon, and D.G. Caldwell. Upper-body kinesthetic teaching of a free-standing humanoid robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3970–3975, May 2011. 28

[80] S. Hyon, J. Moren, and G. Cheng. Humanoid batting with bipedal balancing. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 493–499, Dec 2008. 28

[81] Albert Mukovskiy, Jean-Jacques E. Slotine, and Martin A. Giese. Dynamically stable control of articulated crowds. *Journal of Computational Science*, 4(4):304 – 310, 2013. 28

[82] Sylvain Calinon and Aude Billard. Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm. In *Proceedings of the 22nd international conference on Machine learning*, pages 105–112. ACM, 2005. 29

[83] Jakob Verbeek. Learning nonlinear image manifolds by global alignment of local linear models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1236–1250, Aug. 2006. 29, 122, 124, 126, 133, 175

[84] Ioannis Havoutis and Subramanian Ramamoorthy. Motion planning and reactive control on learnt skill manifolds. *The International Journal of Robotics Research*, 32(9-10):1120–1150, 2013. 29, 122

[85] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 29, 122

[86] Odest Chadwicke Jenkins and Maja J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 56–, New York, NY, USA, 2004. ACM. 29

[87] A.P. Shon, K. Grochow, and R. P N Rao. Robotic imitation from human motion capture using gaussian processes. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 129–134, 2005. 29, 122

[88] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):283–298, 2008. 29

[89] S.K. Banala, Seok Hun Kim, S.K. Agrawal, and J.P. Scholz. Robot assisted gait training with active leg exoskeleton (alex). *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 17(1):2–8, Feb 2009. 32

[90] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977. 34, 103

[91] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989. 35

[92] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978. 36

[93] Christopher S Wallace and David M Boulton. An information measure for classification. *The Computer Journal*, 11(2):185–194, 1968. 36

[94] J. Oliver, T. Roush, P. Gazis, W. Buntine, R. Baxter, and S. Waterhouse. Analysing rock samples for the mars lander. In *Knowledge Discovery and Data Mining (KDD)*, pages 299–303, 1998. 36

[95] D. Asheibi, A. Stirling and D. Sutanto. Analyzing harmonic monitoring data using supervised and unsupervised learning. *Power Delivery, IEEE Transactions on*, 24(1):293–301, January 2009. 36

[96] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003. 36, 129, 134

[97] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. 37

[98] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Neural Information Processing Systems (NIPS)*, pages 921–928. MIT Press, 2002. 37

[99] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, Feb 1978. 38

[100] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 39, 90

[101] D. Roetenberg, P.J. Slycke, and P.H. Veltink. Ambulatory position and orientation tracking fusing magnetic and inertial sensing. *Biomedical Engineering, IEEE Transactions on*, 54(5):883–890, May 2007. 42

[102] P. Sardain and G. Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(5):630–637, Sept 2004. 46

[103] M. Field, D. Stirling, M. Ros, and Z. Pan. Inertial sensing for human motor control symmetry in injury rehabilitation. In *Advanced Intelligent Mechatronics,*

2013. AIM 2013. IEEE International Conference on, pages 1470–1475, July 2013. 47, 71

[104] S. Das, L. Trutoiu, A. Murai, D. Alcindor, M. Oh, F. De La Torre, and J. Hodgins. Quantitative measurement of motor symptoms in parkinson's disease: A study with full-body motion capture data. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 6789–6792, 2011. 47

[105] D. Kulic, G. Venture, and Y. Nakamura. Detecting changes in motion characteristics during sports training. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 4011 –4014, September 2009. 47

[106] T.G. Mayer, D. Barnes, N.D. Kishino, G. Nichols, R.J. Gatchel, H. Mayer, and V. Mooney. Progressive isoinertial lifting evaluation. i. a standardized protocol and normative database. *Spine*, 13(9):993–997, 1988. 53

[107] H. Lygren, T. Dragesund, J. Joensen, T. Ask, and R.M. Nilssen. Test-retest reliability of the progressive isoinertial lifting evaluation (pile). *Spine*, 30(9):1070–1074, 2005. 53

[108] H. Wind, V. Gouttebarge, P. Kuijer, J. Sluiter, and M. Frings-Dresen. Effect of functional capacity evaluation information on the judgment of physicians about physical work ability in the context of disability claims. *International Archives of Occupational and Environmental Health*, 82(9):1087–1096, 2009. 53

[109] D. Stirling, F. Naghdy, G. Naghdy, M. Field, R. Arunglabi, and D. Kilpatrick. Objective functional capacity assessment using inertial sensor. In *Healthcare*

Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on, pages 272–277, July 2011. 55, 71

[110] Gert S. Faber, Idsart Kingma, and Jaap H. van Dien. Bottom-up estimation of joint moments during manual lifting using orientation sensors instead of position sensors. *Journal of Biomechanics*, 43(7):1432 – 1436, 2010. 55

[111] J. Shippen. Biomechanical analysis of entry, egress and loading of a passenger vehicle with rear hinged rear doors. *International Journal for Traffic and Transport Engineering*, 2(2):107–117, 2012. 57, 62

[112] M. Karg, G. Venture, J. Hoey, and D. Kulic. Human movement analysis as a measure for fatigue: A hidden markov-based approach. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 22(3):470–481, May 2014. 67, 149

[113] P. Beaudoin, S. Coros, M. van de Panne, and P. Poulin. Motion-motif graphs. In *Symposium on Computer Animation*, pages 117–126, 2008. 69

[114] M. Field, D. Stirling, Z. Pan, M. Ros, and F. Naghdy. Recognizing human motions through mixture modeling of inertial data. *Pattern Recognition*, 48(8):2394–2406, 2015. 70

[115] I.C. Gyllensten and A.G. Bonomi. Identifying types of physical activity with a single accelerometer: Evaluating laboratory-trained algorithms in daily life. *Biomedical Engineering, IEEE Transactions on*, 58(9):2656–2663, 2011. 71, 95

[116] M. Brand and A. Hertzmann. Style machines. In *SIGGRAPH*, pages 183–192, 2000. 74

[117] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 75, 176

[118] Carl Edward Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 554–560, December 1999. 75

[119] M Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):381–396, March 2002. 75

[120] P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI Press/MIT Press, 1996. 75

[121] S. Kullback. *Information Theory and Statistics*. John Wiley and Sons, New York, 1959. 75

[122] Eamonn Keogh, Themistoklis Palpanas, Victor B. Zordan, Dimitrios Gunopulos, and Marc Cardle. Indexing large human-motion databases. In *In Proc. 30th VLDB Conf*, pages 780–791, 2004. 79

[123] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957. 85

[124] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, Xiao-Yu Fu, K. Goldberg, and P. Abbeel. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *Robotics and Automa-*

tion (ICRA), 2010 IEEE International Conference on, pages 2074 –2081, May 2010. 97

[125] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25(2):328–373, 2013. 101

[126] Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683 – 696, 1998. 110

[127] Jean-Jacques E Slotine and Weiping Li. Applied nonlinear control. *NJ: Prantice-Hall, Englewood Cliffs*, 1991. 113

[128] Robotis. Robotis - Dynamixel, 2014. http://www.robotis.com/xe/dynamixel_en. 115

[129] I.R. Manchester, M.M. Tobenkin, and J. Wang. Identification of nonlinear systems with stable oscillations. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5792–5797, 2011. 120, 148

[130] E. Gribovskaya, S.M. Khansari-Zadeh, and A. Billard. Learning non-linear multivariate dynamics of motion in robotic manipulators. *The International Journal of Robotics Research*, 30(1):80–117, 2011. 121

[131] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 122, 133

[132] Matthew Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 961–968, 2002. 122

[133] Neil D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Neural Information Processing Systems (NIPS)*, 2003. 122

[134] Piotr Dollár, Vincent Rabaud, and Serge Belongie. Non-isometric manifold learning: Analysis and an algorithm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 241–248, New York, NY, USA, 2007. ACM. 122

[135] M. Field, D. Stirling, Z. Pan, and F. Naghdy. Learning trajectories for robot programing by demonstration using a coordinated mixture of factor analyzers. *Cybernetics, IEEE Transactions on*, PP(99):1–1, 2015. 122

[136] Z. Ghahramani and M.J. Beal. Variational inference for bayesian mixtures of factor analysers. In *In Advances in Neural Information Processing Systems 12*, pages 449–455. MIT Press, 2000. 123

[137] S. Roweis, L.K. Saul, and G.E. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pages 889–896. MIT Press, 2002. 124

[138] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212, 1990. 131

[139] H.J. Luinge and P.H. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and Computing*, 43(2):273–282, 2005. 133

[140] J. Ham, D. D. Lee, and L. K. Saul. Learning high-dimensional correspondences from low-dimensional manifolds. In *Workshop on The Continuum from La-*

beled to Unlabeled Data in Machine Learning and Data Mining at Twentieth International Conference on Machine Learning, pages 34–39, 2003. 133

[141] Robert J. Schilling. *Fundamentals of Robotics: Analysis and Control*. Simon & Schuster Trade, 1st edition, 1996. 171, 173

Appendix A

Robot arm description

In Chapter 5 an experiment is presented using a 6-DOF robot arm. In this appendix we describe the structure of this equipment.

According to the axis label assignment and notation reported in [141] the robot structure corresponds to Figure A.1 and the associated Denavit-Hartenberg parameters are listed in Table A.1

Table A.1: D-H parameters for 6-DOF robot.

θ	**a**	**d**	α	θ_0
q_1	0	d_1	$-\frac{\pi}{2}$	0
q_2	0	d_1	$\frac{\pi}{2}$	$\frac{\pi}{2}$
q_3	a_1	0	0	$\frac{\pi}{2}$
q_4	a_2	0	0	0
q_5	0	0	$\frac{\pi}{2}$	0
q_6	0	d_3	0	0

$$ {}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T \tag{A.1} $$

$$ {}^0_6T = \begin{bmatrix} R & p \\ \mathbf{0} & 1 \end{bmatrix} $$

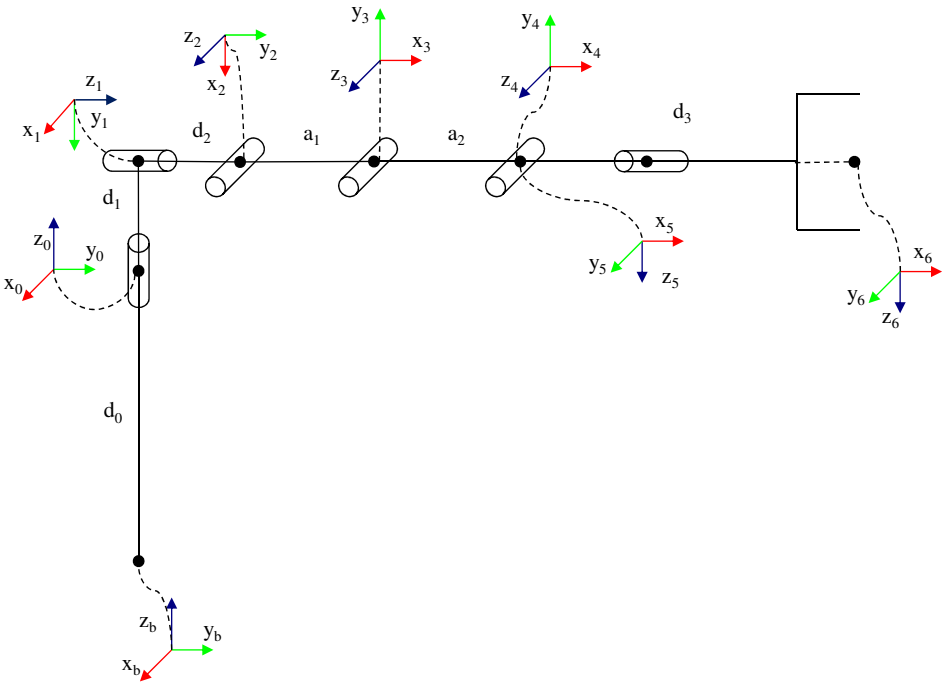

Figure A.1: Kinematic description of 6-DOF robot.

$$
{}^{0}_{6}R = \begin{bmatrix}
c_1 s_2 s_6 - c_6(s_1 s_{345} - c_1 c_2 c_{345}) & c_1 s_2 s_6 + s_6(s_1 s_{345} - c_1 c_2 c_{345}) & s_1 c_{345} + c_1 c_2 s_{345} \\
s_1 s_2 s_6 + c_6(c_1 s_{345} + s_1 c_2 c_{345}) & s_1 s_2 c_6 - s_6(c_1 s_{345} + s_1 c_2 c_{345}) & s_1 c_2 s_{345} - c_1 s_{345} \\
c_1 s_6 + s_2 c_6 c_{345} & c_2 c_6 - s_2 s_6 c_{345} & \frac{1}{2}(c_{2345} - c_{2-3-4-5})
\end{bmatrix}
$$

$$
{}^{0}_{6}p = \begin{bmatrix}
d_3(s_1 c_{345} + c_1 c_2 s_{345}) - d_2 s_1 - a_2(s_1 s_{34} - c_1 c_2 c_{34}) + a_1(c_1 c_2 c_3 - s_1 s_3) \\
d_2 c_1 - d_3(c_1 c_{345} - s_1 c_2 s_{345}) + a_2(c_1 s_{34} + s_1 c_2 c_{34}) + a_1(s_1 c_2 c_3 - c_1 s_3) \\
d_1 + d_0 - s_2(d_3 s_{345} + a_1 c_3 + a_2 c_{34})
\end{bmatrix}
$$

where $c_{23} = \cos(q_2 + q_3)$ and $s_{31} = \sin(q_3 - q_1)$.

Appendix B

Humanoid Robot Description

In Chapter 6 an experiment is presented using a 19-DOF humanoid robot. In this appendix we describe the structure of this robot.

Table B.1: DH parameters for 19DOF humanoid robot.

Axis	θ	**a**	**d**	α	θ_0
0	0	a_1	0	$\frac{\pi}{2}$	0
1	q_1	-a_2	0	0	$-\frac{\pi}{2}$
1	$\frac{\pi}{2}$	$-a_3$	d_1	π	0
2	q_2	a_4	$-d_2$	$\frac{\pi}{2}$	0
3	q_3	a_5	0	0	$-\frac{\pi}{2}$
4	q_4	a_6	0	$-\frac{\pi}{2}$	0
1	$-\frac{\pi}{2}$	a_3	$-d_1$	0	0
5	q_5	$-a_4$	$-d_2$	$-\frac{\pi}{2}$	0
6	q_6	a_5	0	0	$\frac{\pi}{2}$
7	q_7	a_6	0	$-\frac{\pi}{2}$	0
0	$\frac{\pi}{2}$	$-a_7$	0	0	0
8	q_8	0	$-d_3$	$\frac{\pi}{2}$	0
9	q_9	0	d_4	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$
10	q_{10}	a_8	0	0	$\tan^{-1}(\frac{a_{82}}{a_{81}})$
11	q_{11}	a_9	0	0	$-2\tan^{-1}(\frac{a_{82}}{a_{81}})$
12	q_{12}	0	0	$\frac{\pi}{2}$	$\tan^{-1}(\frac{a_{82}}{a_{81}})$
13	q_{13}	a_{10}	0	0	0
0	$\frac{\pi}{2}$	a_7	0	0	0
14	q_{14}	0	$-d_3$	$\frac{\pi}{2}$	0
15	q_{15}	0	d_4	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$
16	q_{16}	a_8	0	0	$-\tan^{-1}(\frac{a_{82}}{a_{81}})$
17	q_{17}	a_9	0	0	$2\tan^{-1}(\frac{a_{82}}{a_{81}})$
18	q_{18}	0	0	$\frac{\pi}{2}$	$-\tan^{-1}(\frac{a_{82}}{a_{81}})$
19	q_{19}	a_{10}	0	0	0

According to the axis label assignment and notation reported in [141] the robot

structure corresponds to Figure B.1 and the associated Denavit-Hartenberg parameters are listed in Table B.1.

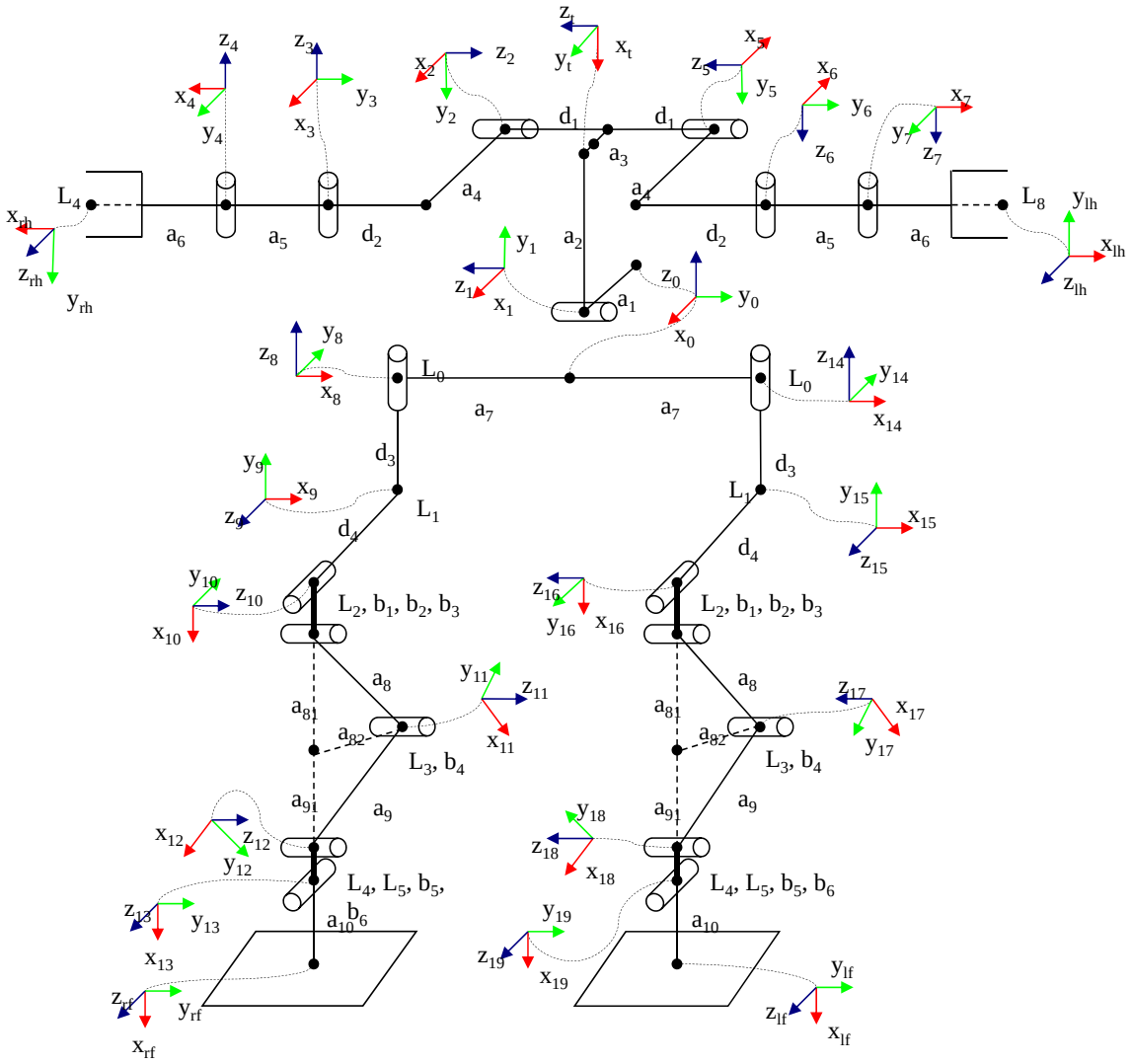

Figure B.1: Kinematic description of 19-DOF humanoid robot.

Appendix C

Derivation of Coordinated Mixture of Factor Analysers Algorithm

In this section we provide a detailed derivation of the coordinated mixture of factor analysers (CFA) algorithm equations, which supplements the original work in [83]. This provides a basis for the manifold learning equations outlined in Chapter 6.

Firstly, one defines the probability distribution of the model. The data is denoted by $\mathbf{x}$ and $\mathbf{z}$ denotes the factor analyser latent coordinates as related by

$$\mathbf{x} = \mathbf{\Lambda z} + \boldsymbol{\mu} + \epsilon. \tag{C.1}$$

Consider a mixture of factor analysers where the prior probability of any one component is π_s. There are K components in the model indexed by $s \in \{1 \dots K\}$.

The mixture of factor analysers probabilistic model can be written as

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, s) &= p(s)p(\mathbf{z}|s)p(\mathbf{x}|\mathbf{z}, s) \\ p(s) &= \pi_s \\ p(\mathbf{z}|s) &= \mathcal{N}(\mathbf{z}|\boldsymbol{\kappa}_s, \boldsymbol{\Sigma}_s) \\ p(\mathbf{x}|\mathbf{z}, s) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_s + \boldsymbol{\Lambda}_s(\mathbf{z} - \boldsymbol{\kappa}_s), \boldsymbol{\Psi}_s). \end{aligned} \tag{C.2}$$

To compute $p(\mathbf{x}|s)$ and $p(\mathbf{z}|\mathbf{x}, s)$ one can apply Bayes theorem where

$$p(\mathbf{x}|s)p(\mathbf{z}|\mathbf{x}, s) = p(\mathbf{x}|\mathbf{z}, s)p(\mathbf{z}|s), \tag{C.3}$$

and apply the linear conditional Gaussian relation below, for which the proof is detailed in [117].

Given a marginal Gaussian distribution for $\mathbf{x}$ and a conditional Gaussian distribution for $\mathbf{y}$ given $\mathbf{x}$ in the form

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}), \end{aligned}$$

the marginal distribution of $\mathbf{y}$ and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ are given by

$$\begin{aligned} p(\mathbf{y}) &= \mathcal{N}(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T) \\ p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}(\mathbf{x}|\boldsymbol{\Sigma}\mathbf{A}^T\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Sigma}\boldsymbol{\Lambda}\boldsymbol{\mu}, \boldsymbol{\Sigma}), \end{aligned}$$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \mathbf{A}^T\mathbf{L}\mathbf{A})^{-1}$.

Applying the above relation to the model under scrutiny, we equate $\boldsymbol{\mu} = \boldsymbol{\kappa}_s$, $\boldsymbol{\Lambda}^{-1} = \boldsymbol{\Sigma}_s$, $\mathbf{A} = \boldsymbol{\Lambda}_s$, $\mathbf{b} = \boldsymbol{\mu}_s - \boldsymbol{\Lambda}_s\boldsymbol{\kappa}_s$, $\mathbf{L}^{-1} = \boldsymbol{\Psi}_s$, and denote $\mathbf{V}_s = (\boldsymbol{\Sigma}_s^{-1} + \boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s)^{-1}$.

Now,

$$\begin{aligned} p(\mathbf{x}|s) &= \mathcal{N}(\mathbf{x}|\mathbf{\Lambda}_s\boldsymbol{\kappa}_s + \boldsymbol{\mu}_s - \mathbf{\Lambda}_s\boldsymbol{\kappa}_s, \mathbf{\Lambda}_s\mathbf{\Sigma}_s\mathbf{\Lambda}_s^T + \mathbf{\Psi}_s) \\ &= \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_s, \mathbf{\Lambda}_s\mathbf{\Sigma}_s\mathbf{\Lambda}_s^T + \mathbf{\Psi}_s) \end{aligned}, \tag{C.4}$$

and

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}, s) &= \mathcal{N}(\mathbf{z}|\mathbf{V}_s\mathbf{\Lambda}_s\mathbf{\Psi}_s^{-1}(\mathbf{x} - \boldsymbol{\mu}_s + \mathbf{\Lambda}_s\boldsymbol{\kappa}_s) + \mathbf{V}_s\mathbf{\Sigma}_s^{-1}\boldsymbol{\kappa}_s, \mathbf{V}_s) \\ &= \mathcal{N}(\mathbf{z}|\boldsymbol{\kappa}_s + \mathbf{V}_s\mathbf{\Lambda}_s\mathbf{\Psi}_s^{-1}(\mathbf{x} - \boldsymbol{\mu}_s), \mathbf{V}_s). \end{aligned} \tag{C.5}$$

$$p(\mathbf{x}) = \sum_{s=1}^{K} p(s)p(\mathbf{x}|s) \tag{C.6}$$

Log-likelihood of the data is then given by

$$\mathcal{L} = \sum_{n=1}^{N} \log p(\mathbf{x}_n). \tag{C.7}$$

Furthermore, marginalising over the component responsibilities gives the following distribution for the manifold mapping,

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) &= \sum_{s=1}^{K} p(s|\mathbf{x})p(\mathbf{z}|\mathbf{x}, s) \\ &= \sum_{s=1}^{K} \frac{p(\mathbf{x}|s)p(s)}{p(\mathbf{x})} p(\mathbf{z}|\mathbf{x}, s). \end{aligned} \tag{C.8}$$

If we propose an approximate distribution for the manifold centred over each data point,

$$q(\mathbf{z}|\mathbf{x}_n) = \mathcal{N}(\mathbf{z}|\mathbf{z}_n, \mathbf{\Sigma}_n), \tag{C.9}$$

we can then use the Kullback-Leibler divergence between the approximate and true

distribution,

$$\mathcal{D}(q(\mathbf{z}|\mathbf{x}_n)\|p(\mathbf{z}|\mathbf{x}_n)) = \int q(\mathbf{z}|\mathbf{x}_n) \log \frac{q(\mathbf{z}|\mathbf{x}_n)}{p(\mathbf{z}|\mathbf{x}_n)} d\mathbf{z}, \tag{C.10}$$

to introduce a penalised log-likelihood function. This will penalise the log-likelihood function if the mapping distribution over each point becomes non-Gaussian or multi-modal. Let the function be defined as,

$$\begin{aligned}
\mathcal{L}' &= \sum_{n=1}^{N} \Big[\log p(\mathbf{x}_n) - \mathcal{D}(q(\mathbf{z}|\mathbf{x}_n)\|p(\mathbf{z}|\mathbf{x}_n)) \Big] \\
&= \sum_{n=1}^{N} \Big[\log p(\mathbf{x}_n) - \int q(\mathbf{z}|\mathbf{x}_n) \log \frac{q(\mathbf{z}|\mathbf{x}_n)}{p(\mathbf{z}|\mathbf{x}_n)} d\mathbf{z} \Big] \\
&= \sum_{n=1}^{N} \Big[\log p(\mathbf{x}_n) - \Big(\int q(\mathbf{z}|\mathbf{x}_n) \log q(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} \Big) + \int q(\mathbf{z}|\mathbf{x}_n) \log p(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} \Big] \\
&= \sum_{n=1}^{N} \Big[\int q(\mathbf{z}|\mathbf{x}_n) \log p(\mathbf{x}_n) d\mathbf{z} - \Big(\int q(\mathbf{z}|\mathbf{x}_n) \log q(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} \Big) + \int q(\mathbf{z}|\mathbf{x}_n) \log p(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} \Big] \\
&= \sum_{n=1}^{N} \Big[- \Big(\int q(\mathbf{z}|\mathbf{x}_n) \log q(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} \Big) + \int q(\mathbf{z}|\mathbf{x}_n) \log (p(\mathbf{z}|\mathbf{x}_n) p(\mathbf{x}_n)) d\mathbf{z} \Big] \\
&= \sum_{n=1}^{N} \Big[\mathcal{H}(q(\mathbf{z}|\mathbf{x}_n)) + \int q(\mathbf{z}|\mathbf{x}_n) \log p(\mathbf{x}_n, \mathbf{z}) d\mathbf{z} \Big]
\end{aligned} \tag{C.11}$$

Using $\int q(\mathbf{z}|\mathbf{x}_n) d\mathbf{z} = 1$, product rule and defining $\mathcal{H}(q(\mathbf{z}|\mathbf{x}_n)) = -\int q(\mathbf{z}|\mathbf{x}_n) \log q(\mathbf{z}|\mathbf{x}_n) d\mathbf{z}$. The function $\mathcal{H}(.)$ is known as the entropy of a distribution.

With a distribution over the mixture components $q(s|\mathbf{x}_n)$ to bound the mixture

log-likelihoods:

$$\begin{aligned}\log p(\mathbf{x}_n, \mathbf{z}) &\geq \log p(\mathbf{x}_n, \mathbf{z}) - \mathcal{D}(q(s|\mathbf{x}_n)\|p(s|\mathbf{x}_n, \mathbf{z})) \\ &= \log p(\mathbf{x}_n, \mathbf{z}) - \Big(\sum_{s=1}^{K} q(s|\mathbf{x}_n)\log q(s|\mathbf{x}_n) - \sum_{s=1}^{K} q(s|\mathbf{x}_n)\log p(s|\mathbf{x}_n, \mathbf{z})\Big) \\ &= \mathcal{H}(q(s|\mathbf{x}_n)) - \sum_{s=1}^{K} q(s|\mathbf{x}_n)\log p(\mathbf{x}_n, \mathbf{z}, s)\end{aligned} \tag{C.12}$$

We now insert the bounded $\log p(\mathbf{x}_n, \mathbf{z})$ into Equation (C.11) to produce

$$\begin{aligned}\mathcal{L}' \geq \Phi &= \sum_{n=1}^{N}\Big[\mathcal{H}(q(\mathbf{z}|\mathbf{x}_n)) + \int q(\mathbf{z}|\mathbf{x}_n)\Big[\mathcal{H}(q(s|\mathbf{x}_n)) - \sum_{s=1}^{K} q(s|\mathbf{x}_n)\log p(\mathbf{x}_n, \mathbf{z}, s)\Big]d\mathbf{z}\Big] \\ &= \sum_{n=1}^{N}\Big[\mathcal{H}(q(s|\mathbf{x}_n)) + \mathcal{H}(q(\mathbf{z}|\mathbf{x}_n)) + \sum_{s=1}^{K} q(s|\mathbf{x}_n)\int q(\mathbf{z}|\mathbf{x}_n)\log p(\mathbf{x}_n, \mathbf{z}, s)d\mathbf{z}\Big]\end{aligned} \tag{C.13}$$

To obtain an explicit form for the log-likelihood one can substitute the corresponding relations in Equation (C.2) into this equation

$$\begin{aligned}\log p(\mathbf{x}_n, \mathbf{z}, s) &= \log(p(\mathbf{z}|s)) + \log(p(\mathbf{x}_n|\mathbf{z}, s)) + \log p(s) \\ \log(p(\mathbf{z}|s)) &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_s| - \frac{1}{2}(\mathbf{z} - \boldsymbol{\kappa}_s)^T\boldsymbol{\Sigma}_s^{-1}(\mathbf{z} - \boldsymbol{\kappa}_s) \\ \log(p(\mathbf{x}|\mathbf{z}, s)) &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Psi}_s| \\ &\quad - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_s - \boldsymbol{\Lambda}_s(\mathbf{z} - \boldsymbol{\kappa}_s))^T\boldsymbol{\Psi}_s^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_s - \boldsymbol{\Lambda}_s(\mathbf{z} - \boldsymbol{\kappa}_s))\end{aligned} \tag{C.14}$$

If all terms independent of $\mathbf{z}$ are taken outside the integral in Equation (C.13), the remaining terms in the integral associated with $\log(p(\mathbf{z}|s))$ are of the following form

$$\int q(\mathbf{z}|\mathbf{x}_n)\Big[\mathbf{z}^T\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\kappa}_s + \boldsymbol{\kappa}_s^T\boldsymbol{\Sigma}_s^{-1}\mathbf{z} + \mathbf{z}^T\boldsymbol{\Sigma}_s^{-1}\mathbf{z}\Big]d\mathbf{z}, \tag{C.15}$$

The expectation of $\mathbf{z}$ is $\mathbb{E}(\mathbf{z}) = \mathbf{z}_n$ and by using cyclic property of the trace, $\mathbf{Tr}(\mathbf{z}^T\boldsymbol{\Sigma}_s^{-1}\mathbf{z}) = \mathbf{Tr}(\boldsymbol{\Sigma}_s^{-1}\mathbf{z}\mathbf{z}^T)$, the expectation of $\mathbf{z}\mathbf{z}^T$ is $\mathbb{E}(\mathbf{z}\mathbf{z}^T) = \mathbf{z}_n\mathbf{z}_n^T + \boldsymbol{\Sigma}_n$ and finally this implies that $\mathbb{E}(\mathbf{z}^T\boldsymbol{\Sigma}_s^{-1}\mathbf{z}) = \mathbf{z}_n^T\boldsymbol{\Sigma}_s^{-1}\mathbf{z}_n + \mathbf{Tr}(\boldsymbol{\Sigma}_s^{-1}\boldsymbol{\Sigma}_n)$.

Reassembling this term then results in the following,

$$\int q(\mathbf{z}|\mathbf{x}_n)\log(p(\mathbf{z}|s))d\mathbf{z} = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_s| - \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T)), \quad \text{(C.16)}$$

where $\mathbf{z}_{ns} = \mathbf{z}_n - \boldsymbol{\kappa}_s$.

The same procedure can be followed for $\log(p(\mathbf{x}|\mathbf{z}, s))$ where the quantities to be integrated are

$$\begin{aligned}\int q(\mathbf{z}|\mathbf{x}_n)\Big[&- \mathbf{z}^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\mathbf{x}_{ns} - \mathbf{x}_{ns}^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\mathbf{z} \\ &- \mathbf{z}^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\boldsymbol{\kappa}_s - \boldsymbol{\kappa}_s^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\mathbf{z} + \mathbf{z}^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\mathbf{z}\Big]d\mathbf{z}\end{aligned} \quad \text{(C.17)}$$

with $\mathbf{x}_{ns} = \mathbf{x}_n - \boldsymbol{\mu}_s$. Using same property, namely $\mathbb{E}(\mathbf{z}^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\mathbf{z}) = \mathbf{z}_n^T\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\mathbf{z}_n + \mathbf{Tr}(\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\boldsymbol{\Sigma}_n)$, we can obtain,

$$\begin{aligned}\int q(\mathbf{z}|\mathbf{x}_n)\log(p(\mathbf{x}|\mathbf{z}, s))d\mathbf{z} = &-\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_s| - \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\boldsymbol{\Sigma}_n) \\ &- \frac{1}{2}(\mathbf{x}_{ns} - \boldsymbol{\Lambda}_s\mathbf{z}_{ns})^T\boldsymbol{\Psi}_s^{-1}(\mathbf{x}_{ns} - \boldsymbol{\Lambda}_s\mathbf{z}_{ns}),\end{aligned} \quad \text{(C.18)}$$

$$\mathcal{F} = \sum_{n=1}^{N}\sum_{s=1}^{K} q_{ns}[\mathcal{S}_{ns} - \mathcal{E}_{ns}]$$

$$\mathcal{S}_{ns} = \frac{1}{2}\log|\boldsymbol{\Sigma}_n| - \log q_{ns}$$

$$\begin{aligned}\mathcal{E}_{ns} = &\log p(s) - \frac{D+d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_s| - \frac{1}{2}\log|\boldsymbol{\Psi}_s| \\ &- \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T)) - \frac{1}{2}\mathbf{Tr}(\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s\boldsymbol{\Sigma}_n) \\ &- \frac{1}{2}(\mathbf{x}_{ns} - \boldsymbol{\Lambda}_s\mathbf{z}_{ns})^T\boldsymbol{\Psi}_s^{-1}(\mathbf{x}_{ns} - \boldsymbol{\Lambda}_s\mathbf{z}_{ns})\end{aligned} \quad \text{(C.19)}$$

We maximise the log-likelihood $\mathcal{F}$ by finding a gradient with respect to each parameter, $\{\log q_{ns}, \mathbf{z}_n, \boldsymbol{\Sigma}_n, \boldsymbol{\mu}_s, \boldsymbol{\kappa}_s, \boldsymbol{\Sigma}_s, \boldsymbol{\Lambda}_s, \boldsymbol{\Psi}_s\}$, and setting it to zero. This leads to the following expressions per parameter,

$$\frac{\partial \mathcal{F}}{\partial q_{ns}} = \sum_{n=1}^{N}\sum_{s=1}^{K}(\log q_{ns} - \mathcal{E}_{ns}) = 0$$
$$q_{ns} = \frac{e^{\mathcal{E}_{ns}}}{\sum_{s=1}^{K} e^{\mathcal{E}_s}}$$

$$\frac{\partial \mathcal{F}}{\partial \mathbf{z}_n} = \sum_{n=1}^{N} q_{ns}(\boldsymbol{\Lambda}_s^T \boldsymbol{\Psi}_s^{-1}(\mathbf{x}_{ns} + \boldsymbol{\Lambda}_s\boldsymbol{\kappa}_s - \boldsymbol{\Lambda}_s\mathbf{z}_n) - \boldsymbol{\Sigma}_s^{-1}\mathbf{z}_{ns}) = 0$$
$$\mathbf{z}_n = \sum_{n=1}^{N} q_{ns}((\boldsymbol{\Sigma}_s^{-1} + \boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s)^{-1}\boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}(\mathbf{x}_{ns} + \boldsymbol{\kappa}_s))$$

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\Sigma}_n} = \sum_{n=1}^{N} q_{ns}(\boldsymbol{\Sigma}_n^{-1} - \boldsymbol{\Sigma}_s^{-1} - \boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s) = 0$$
$$\boldsymbol{\Sigma}_n^{-1} = \sum_{n=1}^{N} q_{ns}(\boldsymbol{\Sigma}_s^{-1} + \boldsymbol{\Lambda}_s^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{\Lambda}_s)$$

$$\boldsymbol{\kappa}_s = \sum_{n=1}^{N} q_{ns}\mathbf{z}_n$$

$$\boldsymbol{\mu}_s = \sum_{n=1}^{N} q_{ns}(\mathbf{x}_n - \boldsymbol{\Lambda}_s\mathbf{z}_{ns})$$

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\Sigma}_s} = \sum_{n=1}^{N} q_{ns}[\boldsymbol{\Sigma}_s^{-1}(\boldsymbol{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T)\boldsymbol{\Sigma}_s^{-1} - \boldsymbol{\Sigma}_s^{-1}] = 0$$
$$\boldsymbol{\Sigma}_s = \sum_{n=1}^{N} q_{ns}[\boldsymbol{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T]$$

$$\frac{\partial \mathcal{F}}{\partial \mathbf{\Lambda}_s} = \sum_{n=1}^{N} q_{ns}(\mathbf{\Psi}_s^{-1}(\mathbf{x}_{ns} - \mathbf{\Lambda}_s \mathbf{z}_{ns})\mathbf{z}_{ns}^T - \mathbf{\Psi}_s^{-1}\mathbf{\Lambda}_s \mathbf{\Sigma}_n) = 0$$

$$= \sum_{n=1}^{N} q_{ns}(\mathbf{x}_{ns}\mathbf{z}_{ns}^T - \mathbf{\Lambda}_s(\mathbf{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T))$$

$$\mathbf{\Lambda}_s = \sum_{n=1}^{N} q_{ns}[\mathbf{x}_{ns}\mathbf{z}_{ns}^T(\mathbf{\Sigma}_n + \mathbf{z}_{ns}\mathbf{z}_{ns}^T)^{-1}]$$

$$[\mathbf{\Psi}_s^{\mathbf{x}}]_{jj} = \sum_{n=1}^{N} q_{ni}^{\mathbf{x}}\left[(\mathbf{x}_{ns} - \mathbf{\Lambda}_s^{\mathbf{x}}\mathbf{z}_{ns})_j^2 + (\mathbf{\Lambda}_s^{\mathbf{x}}\mathbf{\Sigma}_n\mathbf{\Lambda}_s^{\mathbf{x}T})_{jj}\right]$$

where the jj subscript refers to the entries along the diagonal of the matrix. These equations can be extended to the correspondence algorithm detailed in Algorithm 2 in Chapter 6.