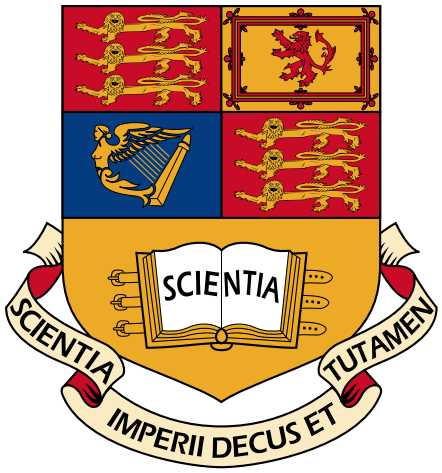
Imperial College London

Individual Project Report

Large-Scale Extraction, Recognition and Prediction of Movement Primitives

Author:
David Haber
dh909@imperial.ac.uk
CID: 006080000

Supervisor:
Dr. A. Aldo Faisal
a.faisal@imperial.ac.uk



18 June 2013

Submitted in partial fulfilment of the requirements for the MEng degree, Imperial College London, Department of Computing.

"The brain is encased in the head, the part of the body which in most walking, flying or swimming animals is the leading end of the moving body."

(V. Braitenberg)

Abstract

Processing neurological data streams is often infeasible when using current state-of-the-art Bayesian methods that make inference computationally complex. Here, we first develop a novel Big Data approach to segmenting and recognising complex time-series for neurological purposes. We subsequently apply our method to real hand movement data that was recorded during both everyday tasks and reconstruction of Lower Paleolithic stone tools. Our data sets contain over one million samples in high dimensional space which allows us to show that the method scales linearly in the number of data points and is near constant in dimensionality. We demonstrate that our method is very robust to noise and achieves an accuracy of over 86% on an artificial data set.

Exploiting the segments obtained and correlation structures between individual fingers, we conceptualise an improved neuroprosthetic controller. Our design overcomes the limitations due to signal extraction currently affecting the potential usage of prosthetics in everyday activities. We propose a two-step process using mixture models and linear regression to predict movement behaviour of injured or lost body limbs, while observing limbs that are still intact. Our results show that we can achieve statistically significantly higher accuracy by taking into account a segmented data set instead of considering unsegmented data, supporting a modular approach to movement generation.

Acknowledgements

First and foremost, I would like to say thank you to my *mother*, who has devoted every single day of her life over the past eighteen years to her children. Your energy, strong will and courage inspires me every day and I hope that one day I can live up to what you have given me on the way.

I would like to thank my supervisor and personal tutor, *Dr Aldo Faisal*, for supporting me throughout my time at Imperial College, providing invaluable advice in many situations and supervising this project. This project would not have achieved this much without the continuous support of *Andreas Thomik*. Thank you for your tremendous effort and time invested to help me with this project, always offering constructive and honest feedback and helping me publish the contents of this work. I have learned a lot from both of you and hope that I have the chance to work with you on interesting projects in the future!

I acknowledge *Professor Daniel Rueckert*'s feedback on my interim report and recognise *Jovana Belic*'s work for collecting the data sets that were used extensively throughout this project.

Dr Silvia Chiappa, Microsoft Research, Cambridge, and *Professor Jan Peters*, Computer Science Department, Technische Universität Darmstadt, kindly provided us with valuable information and source code which supported this project.

Finally, I would like to say thank you to all my *friends* for their continuous encouragement, support and patience throughout my studies.

Contributions

This project contributes two major pieces of work which were developed in the Brain & Behaviour Lab, Department of Computing, Imperial College London between October 2012 and June 2013. Firstly, we devised a novel computational method that allows for processing and segmenting of large neurological data sets. Secondly, we developed a concept for an improved neuroprosthetic controller that exploits the segments generated in the first method and correlation between finger joints to predict missing movement behaviour in real-time.

The similarity metrics Goodness of Fit and Maximum Cross-Correlation presented in this report were previously developed by Andreas Thomik, Department of Bioengineering, Imperial College London. In all other cases it is clearly indicated where previously existent work was used.

The contents of chapter 3 have been submitted for publication under the title *"Simple Unsupervised Time Series Segmentation for Processing High-Dimensional Neurological Data Streams in Linear Time"* (Haber, Thomik, and Faisal 2013) for the Neural Information Processing Systems (NIPS) conference (5-8 December 2013, Lake Tahoe, Nevada, United States). The work presented in chapter 4 has been submitted for publication under the title *"Real-Time Movement Prediction for Improved Control of Neuroprosthetic Devices"* (Thomik, Haber, and Faisal 2013) for the 6th International IEEE EMBS Conference on Neural Engineering (6-8 November 2013, San Diego, California, United States). As of submission date of this report, both are currently in review.

Contents

1 Introduction

"We have a brain for one reason and one reason only; that is to produce adaptable and complex movements. There is no other reason to have a brain." [1] While this potentially defines its main responsibility, it is still not understood how the brain generates such adaptable and complex movement in the context of our highly redundant musculoskeletal system (Kutch and Valero-Cuevas 2011). After more than 100 years of research (Sherrington 1908), existing theories suggest that we can decompose movement into a finite set of sub-sequences which are concatenated in an order to form complex movement behaviour. This idea becomes particularly interesting if we assume that the order of concatenation is not purely random but follows a particular rule. Such decomposition rules seem to exist in many natural and artificial phenomena (grammar for connecting words of speech, tabs for playing music) and have given rise to similar theories that attempt to explain the seemingly effortless way the brain controls our motor system.

One way to investigate whether the brain uses similar rules of decomposition for the generation of complex movement would be to analyse large sets of movement data collected from humans during everyday activities. While motion capture devices allow us to accurately record movement behaviour, computational techniques for processing and analysing large-scale movement data do not currently exist. A Big Data approach to analysing large neurological data streams would potentially enable us to draw new conclusions about the brain's process to generate versatile movement.

Furthermore, a deeper understanding of this process would allow us to utilise our knowledge to develop improved control mechanisms for neuroprosthetic devices. These would offer the chance for people that suffer from partial loss of physical capabilities to regain naturalistic mobility and dexterity by controlling a prosthetic replacement such as an artificial hand. Current technologies present both invasive (Taylor, Tillery, and Schwartz 2002; Velliste, Perel, and Spalding 2008; Hochberg et al. 2006; Hochberg, Bacher, and Jarosiewicz 2012) and non-invasive (Wolpaw and McFarland 1994; Wolpaw and McFarland 2004; Müller-Putz and Scherer 2005) methods and EMG (Bitzer and Smagt 2006; Kuiken et al. 2007; Kuiken et al. 2009) as means to extract control signals from the brain and nervous system. However, under these methods it is still problematic to extract enough information such that accurate control is guaranteed. Replacing lost limbs with prosthetic devices which provide the same functionality as natural limbs is therefore an open challenge until today. Even the most advanced prostheses are often limited to the most basic grasps.

In this report we present two novel statistical machine learning approaches to movement segmentation and prediction. Our first method (chapter 3) enables a Big Data

[1] Daniel Wolpert, "The Real Reason For Brains", TED Talk 2011

driven approach to movement segmentation and recognition by applying a combination of computationally inexpensive statistical methods to large neurological data sets. Where existing methods become computationally intractable, the linear run-time properties of our method allow for efficient and accurate segmentation of very long and high-dimensional time-series. We evaluate our implementation on three different data sets: an artificially generated data set, hand movement data collected during everyday tasks and on hand data that was recorded during the reconstruction of Lower Paleolithic stone tools. We finally demonstrate that our algorithm is very robust to noise with a consistent segmentation accuracy of 86.5% on the artificial data set.

In the second part of this report (chapter 4), we exploit our segmentation results and typical correlation structures between our individual finger joints and develop a concept for an improved neuroprosthetic controller which complements partially observed body parts with real-time predictions about the behaviour of the missing limbs. For this we developed a computational method which relies on Gaussian mixture models and linear regression to estimate which type of movement is currently observed and predict unobserved behaviour in a two-step process. We will present our test results on a large data set containing hand movements and show that instead of considering the entire, unsegmented data set our method yields statistically significantly higher accuracy by taking a modular approach to predicting unobserved behaviour.

With this project we hope to contribute computational techniques that will in the future allow us to (1) obtain a better understanding of the brain using Big Data approaches and (2) develop an alternative and cost-effective neuroprosthetic device exhibiting more naturalistic and versatile movement behaviour.

2 Background

Daniel Wolpert further argued in his talk that "our - the human - brain, having gone through several phases of evolutionary transformation, is not built to think or to feel but rather to coordinate movement". It should not be without any reason that thinking and feeling are interpreted as being subordinate to movement coordination. Everything requires movement in one form or the other; we eat, we breathe, we speak and we move. In contract to this is the extraordinary complexity of the human body and its musculoskeletal system. With the human hand having 21 degrees of freedom on its own and the muscular-skeletal system ranging above 600 degrees of freedom many people have asked how our brain generates such versatile movement behaviour. Even young children can perform very accurate and goal directed tasks which are impossible to be similarly executed by current technology. Given the remarkably diverse set of movement behaviour common to all healthy humans, it generally appears that the brain manages to control a highly-complex system in a seemingly effortless way where even the most advanced technology fails.

This becomes particularly interesting when we consider that the design of our muscoloskeletal system is assumed to be strikingly redundant and as a result of this the central nervous system has several options when generating movement for a specific task (Kutch and Valero-Cuevas 2011). Experiments performed by Kutch and Valero-Cuevas (2011) have shown that only a small subset of possible forces that can be executed to perform a task is robust to loss of any one muscle. This indicates that the muscular redundancy is necessary, however the idea of redundancy still greatly increases the complexity incurred when generating movement. Current research has put a lot of focus on exploring possible explanations behind the seemingly effortless motor control in the context of a highly complex system. We will present some of the important findings in the following sections.

2.1 Computational Theories of Motor Control

When examining how the brain controls the presented level of complexity, we could imagine two extreme cases: (1) Our nervous system stores a hard-coded command sequence for any possible movement scenario (pairs of starting and target position). Upon execution of a desired movement, a simple look-up would be performed. (2) Our brain computes the motor commands from scratch and on demand for every possible starting and target configuration.

With our current understanding, both approaches seem to become infeasible given the almost infinite number of movement possibilities. (1) would require an equally large

storage component that is capable of providing all motor commands on demand. (2) is impractical for highly complex, non-linear systems, including the human body, as it becomes computationally infeasible.

In the following, we will look at various theories that lie between these two extremes and suggest primitive movement blocks as the foundation for motor control. These theories include *reflexes*, *spinal force fields* and *muscle synergies*.

2.1.1 Reflexes

Reflexes were long seen as building blocks for complex movement. In Sherrington (1908), reflexes are defined as the unit mechanism of the nervous system and act as the transport mechanism for movement commands. A reflex arc consists of three components: a receptor, wich is an organ that initiates movement, a conductor, responsible for transmitting stimuli, and an effector, executing the movement, e.g. muscle cells. Reflexes are classified as either simple or compound. While simple reflexes are interpreted as unit reactions in nervous integration, compound reflexes are made up of simple reflexes and are responsible for co-ordination.

Sherrington's theory is plausible in the context of rapid adaptation when a reaction is provoked. However, due to the nature of reflexes they are unlikely to be responsible for the generation of complex voluntary movement. Nonetheless, during our research this theory was the oldest found, dating back to 1908. Its importance is additionally drawn from the many citations and uses in research until today.

2.1.2 Spinal Force Fields

In Bizzi et al. (1995), experiments were conducted during which researchers microstimulated different sides of the frog's spinal cord (Figure 2.1/**E**). The motor responses were characterised by *force fields*. A force field is described as a mapping that associates each position of the frog's hind limb with a corresponding force generated by the neuro-muscular system. The group found that the majority of fields generated were characterised by a single equilibrium point describing a position without an end-point force. Furthermore, they were able to group these force fields into a smaller number of classes where each class was preferentially evoked from a distinct region in the frog's spinal cord (Bizzi et al. 2002).

The discreteness of force orientations found during these experiments together with their organised mapping in the spinal cord gives rise to a hypothesis that supports a form of modular organisation of the spinal cord motor circuitry.

This is underlined by further experiments in which the group concluded that the described convergent force fields follow a principle of vector summation (Figure 2.1/**A-D**), They simultaneously stimulated two sides of the spinal court and found that the resulting force field is equivalent to the sum of the two individual stimulations of the respective sides.

In both Bizzi et al. (1995) and Bizzi et al. (2002), it is concluded that the vector summation of force fields eliminates much of the complexity between neurons and muscles

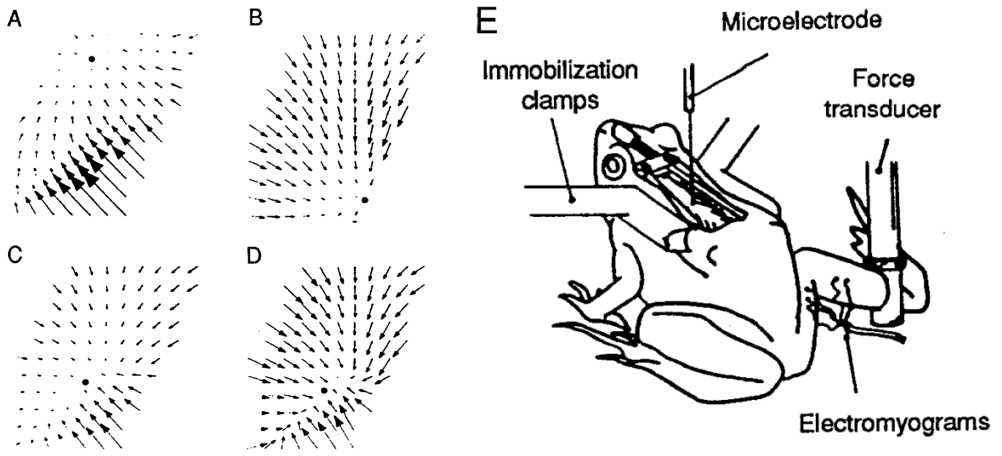


Figure 2.1: Experiments conducted by Bizzi. **(A)-(D)** show visualisation of multiple electrical stimuli. **(A)** and **(B)** are individual fields resulting from separate stimulation of two areas. **(C)** is the predicted field by vectorial summation of **(A)** and **(B)**. **(D)** is the actual field obtained from simultaneous stimulation of the two areas. Note the clear resemblance between **(A)-(B)** and **(C)-(D)**. **(E)** shows apparatus used for experiments. Figure from Bizzi et al. (1995).

and further supports the assumption that the vertebrate's brain exploits modular generation of motor responses. Their results found acknowledgement in a study by Kargo and Giszter (2000).

However, one should note that spinal force fields assume an end-point encoding of movement control. In the context of movement involving differently sized objects, for example grasping a small object followed by grasping a larger object, spinal force fields would imply that different force patterns have to be learned for the respective individual movement. Due to the infinite number of differently shaped objects that humans interact with, this can be considered as impossible in practice.

2.1.3 Muscle Synergies

In further work, the force vectors that characterise the spinal force fields are interpreted as an expression of specific groups of synergistically active muscles (Bizzi et al. 2002). Such *muscle synergies* are investigated in D'Avella et al. (2006); d'Avella, Saltiel, and Bizzi (2003) and it is proposed that they are fundamental building blocks of muscle patterns.

In a first set of experiments (d'Avella, Saltiel, and Bizzi (2003)) kicking, jumping and walking muscle patterns were collected from freely moving frogs. The generation of these muscle patterns was modelled as a linear combination of time-varying synergies and it was argued that given one set of synergies, a variety of different patterns can be generated by only choosing different parameters. It was also found that the synergies were highly

consistent across different frogs and shared across different behaviour. As a result of these experiments, it was interpreted that the synergies are shared among a variety of different tasks.

Similar results were found for fast-reaching movements in different directions (D'Avella et al. 2006). Point-to-point movements were recorded from various human subjects. The group was able to extract movement synergies for each individual subject (Figure 2.2) which explained between 73% and 82% of variation in the data.

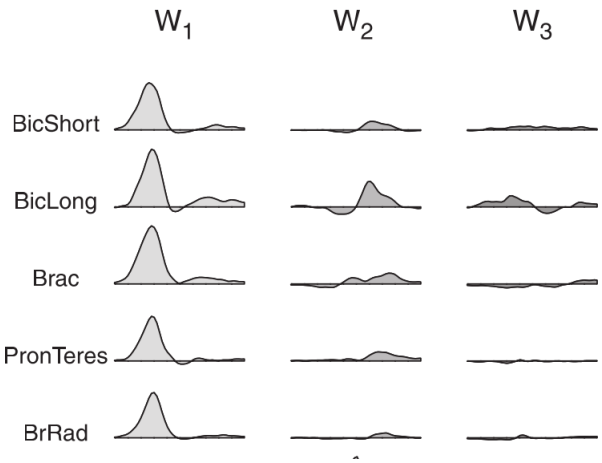


Figure 2.2: Excerpt of time-varying synergies (W1-W3) extracted from human subject. The abbreviations on the left-hand side represent a subset of the muscles observed. Figure from D'Avella et al. (2006).

In the work published and experiments conducted, muscle synergies are all assumed not to be time-invariant and it should be pointed out that the concept of muscle synergies is a rather young theory which still requires better validation in future.

2.2 Exploiting Movement Patterns for Neurotechnology

With regards to hand movements, we can distinguish between *simple* and *complex* behaviour (Faisal et al. 2010). Curling and uncurling a hand, during which the individual joints move in a highly correlated manner, can be classified as simple behaviour. In contrast, expert typing on a keyboard is considered as a complex behaviour where each finger moves independently from the others.

It has been shown that in many everyday situations we move our individual joints in a highly correlated manner (Thomik and Faisal 2012). We term these as *synergistical interaction* between our finger joints. It has been suggested that even the anatomy of our finger muscles has evolved in a way to simplify the control problem of the hand by biomechanically limiting the control of the individual joints (Mason, Gomez, and Ebner 2001). Santello, Flanders, and Soechting (1998) presented that 90% of the variance in

hand configurations could be explained by only 3 principal components [1] when reaching for and grasping a large variety of imagined objects. Similar results were obtained by Mason, Gomez, and Ebner (2001); Gentner and Classen (2006), however it should be noted that all mentioned studies only investigated hand configurations during reaching and grasping movements.

The results found suggest that the brain may use knowledge about such synergistical interaction between individual joints, or more general different parts of the body, to further simplify movement control.

2.3 Artificial and Prothetic Hands

Unlike lizards, that drop off their tails, or other animals that can regrow missing body parts, prosthetic devices are often the only chance for humans to regain mobility and dexterity in everyday life after injury. This project contributes to a larger research area that attempts to find novel ways of generating natural and accurate movement on prosthetic devices. In the following two sections, we will take a look at existing technologies before presenting some of the challenges that have been encountered when developing prosthetics.

2.3.1 Current Technologies

Today, many forms of prosthetics exist. The *hook* is usually seen as the simplest form of a prosthetic device. It is usually affordable, lightweight and robust but has a very different appearance from a natural hand and does not exhibit any movement behaviour. Especially in recent years, more advanced prostheses have been developed, including devices that use electromyography (EMG) signals from muscle contractions to control movement. Some of the most advances prosthetic hands, such as i-limb (Touch Bionics, figure 2.3/**A**) and bebionic3 (bebionic, figure 2.3/**B**), provide spectactular features allowing amputees to regain partial versatility. Their design includes small motors to control the individual fingers and their weight of less than 600 grams allows use in everyday situations [2]. While prostheses have improved greatly, we still face major challenges that have to be overcome before amputees can regain their full dexterity. In addition, highly advanced prostheses are still very costly and not often affordable for the general public. We will summarise some of current challenges in the following section.

2.3.2 Challenges

Despite the great advances that we have seen with recently developed prosthetics, there are two major challenges involved in generating movement on such modern prosthetic hands: (1) A precise signal which encodes the desired movement behaviour has to be

[1] linearly uncorrelated variables, which are often a subset of the total number of variables describing a system, more details can be found in section 3.2.1

[2] Technical specifications from bebionic.com

Figure 2.3: **(A)**: i-limb ultra hand prosthesis (Touch Bionics). **(B)**: bebionic3 prosthesis (RSLSteeper).

generated and transmitted to the device. (2) After the control signal has been processed, the prosthetic device has to generate versatile movements allowing for both firm and soft grasps.

The former focusses on extracting movement commands from the human's brain or nervous system. While we have seen significant improvements by using direct cortical control from both invasive (Taylor, Tillery, and Schwartz 2002; Velliste, Perel, and Spalding 2008; Hochberg et al. 2006; Hochberg, Bacher, and Jarosiewicz 2012) and non-invasive (Wolpaw and McFarland 1994; Wolpaw and McFarland 2004; Müller-Putz and Scherer 2005) electrodes and using sources such as EMG (Bitzer and Smagt 2006; Kuiken et al. 2007; Kuiken et al. 2009), it is still very difficult to extract a meaningful amount of information that can be used for generating artificial movement. This makes prosthetic devices often unsuitable for real-time control. The latter aims at translating the signals into a mechanical force that represents natural and versatile movements. This has been one of the major challenges as, at the same time, prosthetics should be lightweight and wearable.

These problems have yet to be overcome and currently restrict the potential use of neuroprosthetics in everyday activities considerably. As a result they lead to frustration and rejection by individuals using such devices (Roeschlein and Domholdt 1989; Davidson 2002; Cipriani and Zaccone 2008).

2.4 Motion Capture Systems

Capturing real motion data is essential for any further analysis that we want to perform. Without such data we would have to rely on artificially generated movement data which would be (1) hard to generate and (2) not very accurate. We should therefore first take a look at existing systems that enable us to capture motion. Such systems are usually divided into optical and non-optical systems. We require motion capture systems to capture motion accurately, which means they have to feature precise sensors and should

not distort the natural movement to be captured.

Optical Systems

Optical system are often used in combination with optical markers that are attached to a subject. The markers can either be *passive* or *active*. Passive markers usually reflect light, active markers have an illuminated LED source. Markerless optical capture has been developed recently and become more prominent. An example for such a system is given by www.organicmotion.com which allows real-time translation from an actor's performance to a character animation. Interesting work that uses optical motion capturing for movement extraction can be found in Leonardis, Bischof, and Pinz (2006); Rittscher and Blake (1999).

Non-optical Systems

Non-optical systems and in particular so called inertial systems usually consist of a range of sensors (gyroscopes, rotational sensors, etc.) which are attached to the subject in the form of a suit, glove or special piece of clothing. Inertial systems allow for real-time movement capturing without the need for a larger capture area. In contrast to optical systems, they are however usually not suited for determining the absolute position of the subject within an environment.

In the context of this project a CyberGlove I hand glove [3] has been used to collect the required datasets for evaluation (Figure 2.4).

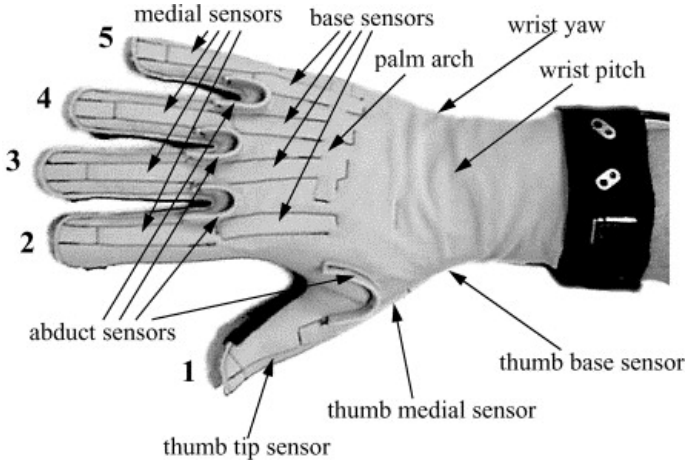


Figure 2.4: CyberGlove I - hand glove used to collect 18-dimensional hand movement data sets. Position of sensors are shown. Image taken from Belić (2010).

The data obtained from the CyberGlove consists of 18-dimensional time-series representing the bending of all joints of the fingers with exception of the distal interphalangeal joint (DIP), as well as flexion/extension and adduction/abduction of the wrist. Every

[3] CyberGlove Systems LLC, San Jose, CA, USA

sensor emits voltage that varies as the sensor is bent. A sampling rate of 80 Hz and a resolution of 8 bit per sensor are used. More information on how the data for this project was obtained can be found in Belić (2010) and in section 3.2.3 of this report.

While the CyberGlove only records hand movements, body suits with inertial sensors such as offered by Animazoo [4] provide the opportunity to record movement from many parts of the body.

Recent Technology

The Canadian startup company "Thalmic Labs" [5] has recently announced their wearable gesture control device "MYO" which recognises movement from lower arm muscle activity. It has gained significant attention through its convenient design and affordable price.

Another company "Leap Motion" [6] has built a device that senses hand movement from a distance. It is not attached to a person's body but rather placed on a table.

With the improvement of "home-use" technology it has recently also become more popular to use normal cameras to recognise movement. Flutter [7], a startup based in Palo Alto, has released an application that recognises hand gestures through a laptop webcam and allows Windows and Mac users to control their music player. While one could argue that such systems are rather inaccurate they allow a wider audience to make use of recently developed algorithms that are able to detect movements.

[4] www.animazoo.com
[5] www.thalmic.com, device not sold as of submission date of this report
[6] www.leapmotion.com
[7] www.flutterapp.com

3 Movement Segmentation from a Big Data Perspective

In section 2.1 we described various computational theories that attempt to explain manages the complexity of motor control. One way of further exploring whether the brain uses primitive building blocks for movement generation would be to analyse large neurological data sets that have been collected from movements of various subjects in everyday situations. If we were able to efficiently and accurately process large scale movement data, we could investigate whether such building blocks exist in the data collected.

In this chapter we will look at existing techniques that attempt to process and segment complex movement data and discuss their limitations. We will then propose our approach to movement segmentation; a simple unsupervised method for accurately segmenting large-scale neurological data sets. (Haber, Thomik, and Faisal 2013). We apply our method, which scales *linearly* in space and time, and is near *constant* in dimensionality of the data, to millions of samples in high dimensional space. These run-time properties make our method highly scalable and suitable for Big Data problems which are often encountered when processing neurological data streams.

3.1 Current Methods

During our research we have found several methods that attempt to process and segment movement data. In this context, algorithms are often classified as either *dictionary-based* or *dictionary-free*.

Dictionary-based methods usually assume that pre-trained motion models exist and that useful movement segments have already been taught in isolation. The individual segments are parametrised and a library of segments is created by estimating the parameters using, for example, linear weighted regression (Meier et al. 2011). These methods then usually start by identifying potential segmentation points from an input data stream which can be done based on the movement's velocity and acceleration profile (Meier, Theodorou, and Schaal 2012). Secondly, they perform time-warping on the segments to obtain a consistent segment duration and finally match the potential segmentation points against available segments in the dictionary in an *offline* (Meier, Theodorou, and Schaal 2012) or *online* (Meier et al. 2011) fashion. The latter method performs simultaneous segmentation and recognition of movement segments which implies sequentially scanning the observed series and matching the current observation with all segments in the dictionary. In essence, while observing motion, it is hypothesised which segment in the library is most likely responsible for the so-far observed trajectory and at which time step it is

going to finish. In Meier et al. (2011) this is done using an Expectation Maximization algorithm.

Estimating the trajectory's goal position online is often difficult due to *co-articulation* which describes that human movement is usually smoothed together and possibly temporally interleaved (Meier, Theodorou, and Schaal 2012). This complicates the process of deciding whether a determined segmentation point corresponds to the actual end-point of the current segment.

In contrast to dictionary-based methods, techniques that do not rely on a dictionary try to infer data segments and their shape directly from the data. This can be done using either a Kalman Filter (Coates, Abbeel, and Ng 2008) or factorial Hidden Markov Models (HMM) (Williams, Toussaint, and Storkey 2006). The former method attempts to train a helicopter to fly a complete aerobatic air show autonomously [1]. The data is captured from multiple sub-optimal expert demonstrations and the learning algorithm presented in Coates, Abbeel, and Ng (2008) finds the most likely hidden trajectory which maximises the joint likelihood of the observed demonstrations. The group reported their results with a standard deviation of 2.3 metres from the optimal trajectory during an autonomous aerobatic show. In Williams, Toussaint, and Storkey (2006), a factorial Hidden Markov Model (HMM) is used to extract primitives from handwriting data without the requirement for any pre-partitioning. In their model, each latent factor is modelled to account for an unobserved movement segment. They further extend their model with a timing module to account for temporal relations when segmenting data. A variational approximation as part of an Expectation Maximisation algorithm is used for learning the model parameters.

We found that dictionary-free models often require more complex probabilistic models for learning and inference. We will present one such approach in the following section and discuss its limitations.

Review of a Probabilistic Approach

As part of this work, we investigated a method proposed by Chiappa and Peters (2010) which attempts to build a probabilistic model allowing inference and learning of *segmentation points* and *shape* of segments. Here, a series of observation is represented by a set of random variables

$$v_{1:T} \equiv v_1, \ldots, v_T \tag{3.1}$$

where $v_t \in \mathbb{R}^V$ and T defines the length of the V-dimensional time-series. The observation is assumed to arise through a noisy transformation and time-warping[2] from a set of hidden trajectories

$$h^{1:S}_{1:M} \equiv h^1_{1:M}, \ldots, h^S_{1:M} \tag{3.2}$$

where $h^i_m \in \mathbb{R}^H$, M is defined as the length of a hidden trajectory, H the dimensionality and S the number of hidden trajectories. Figure 3.1/**Top** shows three basic movement

[1] http://heli.stanford.edu

[2] We use the terms "time-warping" and "resampling" interchangeably in this work.

trajectories (hidden trajectories) of different length that are assumed to form het concatenated observation series in 3.1/**Bottom**. The continuous random variables defined in 3.1 and 3.1 are represented in the Belief network shown in figure 3.2. The two additional sets of discrete random variables shown there, $\sigma_{1:T}$ and $z_{1:T}$, model segment length (minimum and maximum duration of an action), time-warping and which of the hidden trajectories is responsible for an observed segment.

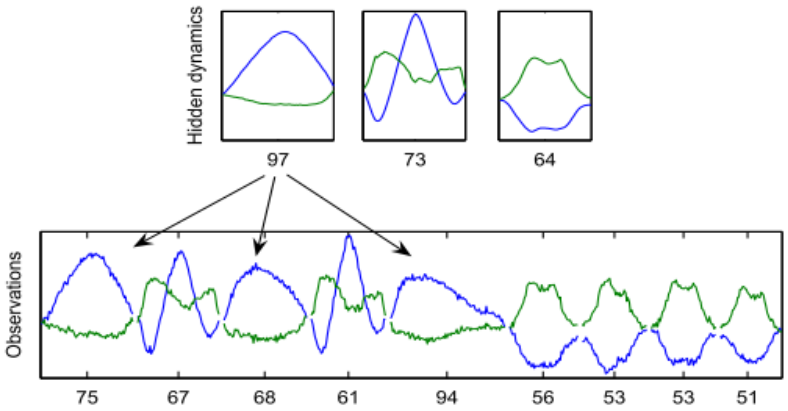


Figure 3.1: **Top**: Hidden trajectories that represent different movement types. **Bottom**: Observed time-series is assumed to be generated from hidden trajectories through a noisy transformation. Figure from Chiappa and Peters (2010).

Different forms of an Expectation Maximisation (EM) algorithm are used to learn the model parameters. Upon convergence of the EM algorithm, the method samples the most likely segmentation points from the probabilistic distributions obtained.

The group investigates three different approaches to ensure computational tractability when computing the posterior distributions required for learning and sampling. These include a variational method, maximum a posteriori (MAP) method and Gibbs sampling. They further show that the MAP method outperforms the other two methods and test their method by segmenting table tennis movements that were recorded using a robot arm as input device.

We began by testing the algorithm with 3-dimensional time-series and two hidden trajectories with 16 data points each. We used 15 iterations for the estimation algorithm and the minimum and duration for each segment was set to 7 and 16 respectively. As shown in figure 3.3, it was found that the method performs well on such simple and short data, giving precise estimations for the shape of predefined hidden trajectories.

Following these initial results we wanted to analyse how the algorithm scales with data of slightly higher dimensionality and with more data points. In a second test, we therefore increased the length of the two hidden trajectories to 61 and set the minimum and maximum duration for each segment to 25 and 55 respectively. We found that the

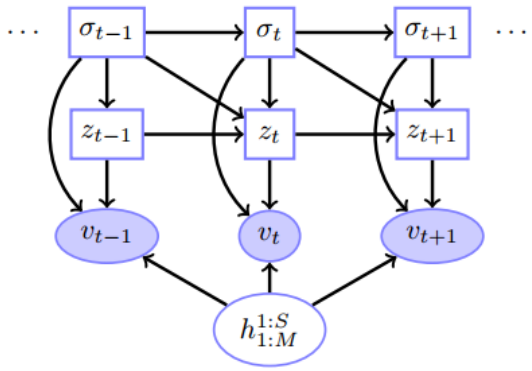


Figure 3.2: Belief network representing the probabilistic model for inference of segmentation and trajectory shape. Model intractability is caused by the interaction between the continuous and discrete latent variables. Figure from Chiappa and Peters (2010).

accuracy of the estimated hidden trajectories decreased significantly (see figure 3.4) while the computational run-time of the algorithm increased notably.

When we increased the dimensionality to 5 while the number of data points and parameters were kept as before, the quality of the estimated segments deteriorated further (see figure 3.5). We eventually observed that the run-time of the algorithm increased exponentially.

We tried to optimise the implementation which was kindly provided by Silvia Chiappa by removing unnecessary parts of the code, pre-allocating data structures and using memory-efficient constructs such as sparse matrices. The performance was however not found to improve significantly. In the context of more complex data, we identified the algorithm's problem in its general approach to performing inference on the probabilistic model. The method presented employs several passes over the data samples and considers *every* single possible segmentation that can be found within a defined time window which is defined by the model's parameters specifying minimum and maximum duration of a segment. The total number of possible segmentation points leads to a *combinatorial complexity* which makes inference and learning generally intractable for movement data of slightly higher complexity and longer duration. The algorithm did therefore not suit our goal of analysing and segmenting complex and long movement time-series as, for example, collected from a motion glove device.

Conclusion

To perform movement segmentation on large neurological data streams, we require a method that is computationally tractable even when applied to very large and complex data sets. We conclude that none of the techniques tested and found as part of our

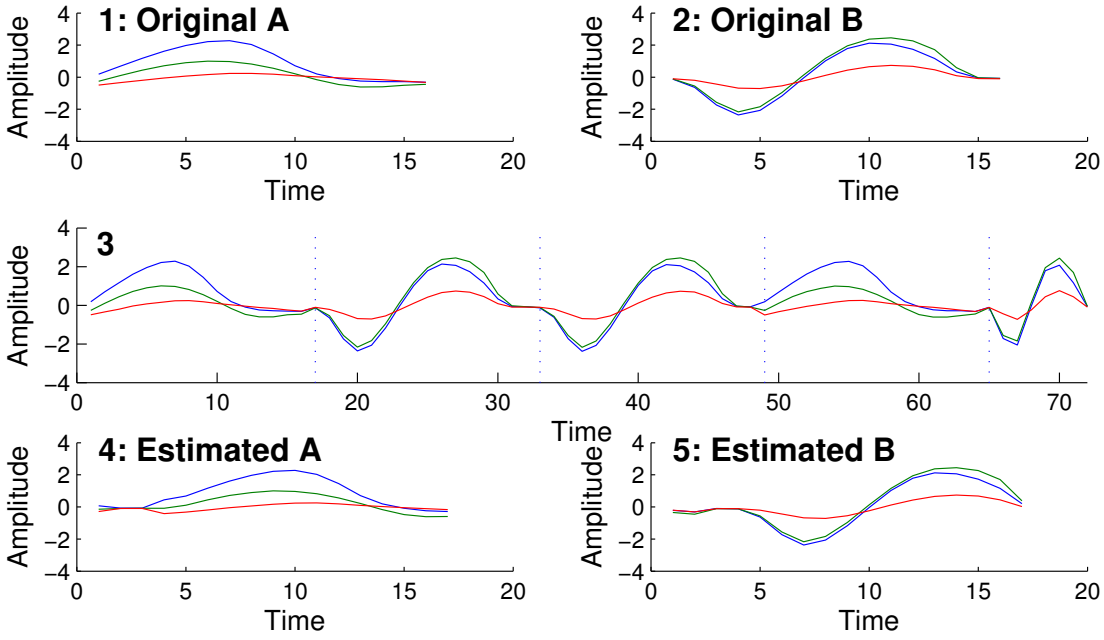


Figure 3.3: Test results from Chiappa and Peters (2010). **(1)-(2)**: Predefined 3-dimensional hidden trajectories (movement segments) of length 16 each. **(3)**: Concatenation of segments of type A and B through noisy transformation. Vertical blue lines indicate original segmentation points. **(4)-(5)**: Estimated trajectories after 15 iterations from **(3)**.

research scale with such large data sets. Many of the methods presented attempt to model all structural aspects of the data with a single model. This makes inference inherently complex and the approach often computationally intractable. We develop an alternative method in the following section.

3.2 Proposed Method

We propose to combine four different, simple and specialised methods for segmenting long and complex neurological data streams. We use (1) Principal Component Analysis (PCA) for data compression, (2) changepoint analysis to determine potential segmentation points in the data, (3) time-series compression to determine the most relevant changepoints and (4) comparison of the extracted segments to determine the subset of generating segments. In the following sections, we will show how the combination of these methods can achieve robust and accurate time-series segmentation and outperform high-concept state-of-the-art algorithms when applied to real world data streams. Finally, we apply our method to over millions of samples of high dimensional space and discuss its run-time and scalability properties.

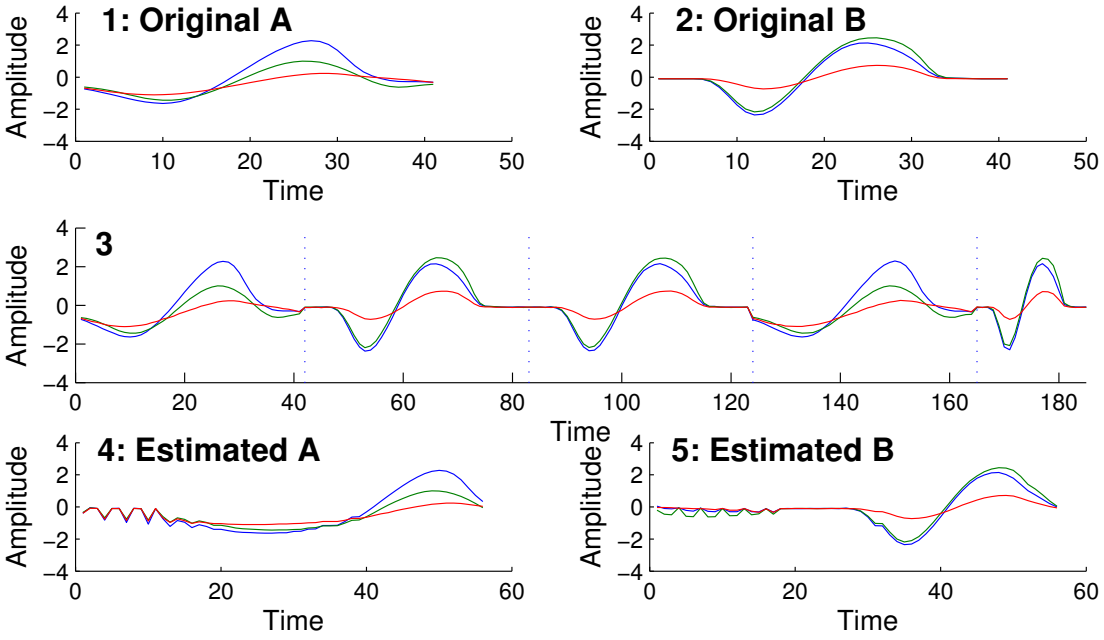


Figure 3.4: Test results from Chiappa and Peters (2010). **(1)-(2)**: Predefined 3-dimensional hidden trajectories (movement segments) of length 41 each. **(3)**: Concatenation of segments of type A and B through noisy transformation. Vertical blue lines indicate original segmentation points. **(4)-(5)**: Extracted trajectories after 15 iterations from **(3)**.

3.2.1 Methodology

Principal Component Analysis

We can exploit the existence of patterns of correlation between joints of fingers during every tasks (see section 2.2) to remove redundant information when processing neurological movement data. This has benefits both in terms of memory and computational costs. *Principal Component Analysis* (PCA), a dimensionality reduction technique which is often used in areas of machine learning and pattern recognition, achieves this by finding a representation of the data with maximum variability in a lower dimensional data space. PCA has as input a set of correlated variables, in this case the joint angles of the hand, and outputs a set of uncorrelated variables of usually lower dimensionality called *principal components* (see Fig. 3.6). More specifically, the D-dimensional input data can be transformed to D'-dimensional space where $D' \leq D$ by calculating the principal components as the eigenvectors of the covariance matrix of the zero mean input. We

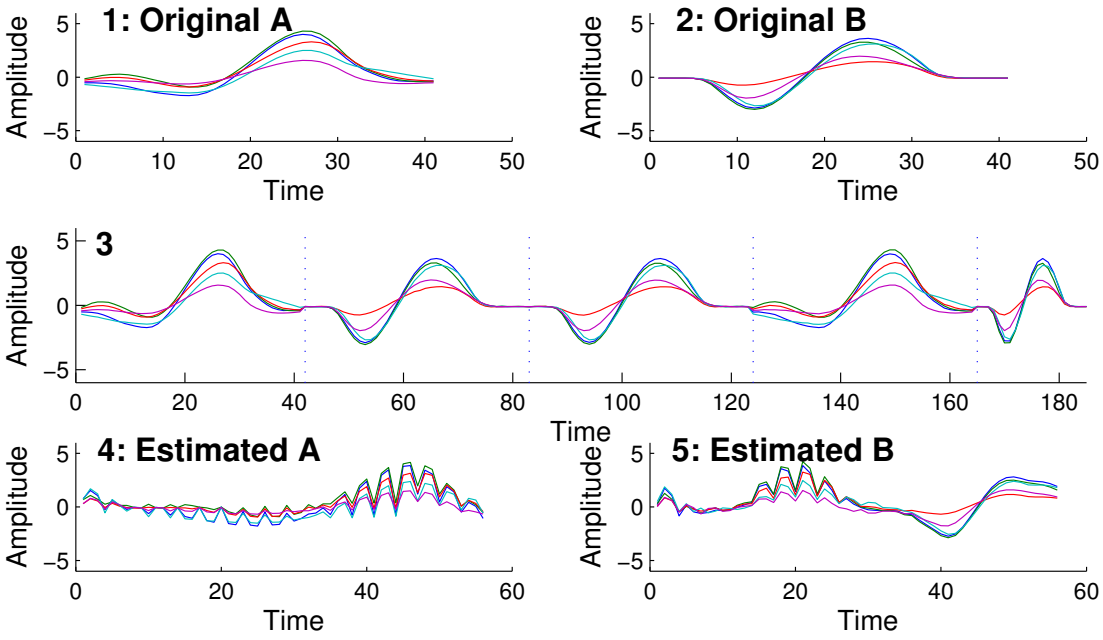


Figure 3.5: Test results from Chiappa and Peters (2010). **(1)-(2)**: Predefined 5-dimensional hidden trajectories (movement segments) of length 41 each. **(3)**: Concatenation of segments of type A and B through noisy transformation. Vertical blue lines indicate original segmentation points. **(4)-(5)**: Extracted trajectories after 15 iterations from **(3)**.

solve

$$\Sigma = \Phi \Lambda \Phi^T \tag{3.3}$$

for the eigenvectors Φ and eigenvalues Λ and achieve dimensionality reduction by usually choosing the D' eigenvectors $\Phi_{PCA} = [\phi_1, \phi_2, \ldots, \phi_{D'}]$ with the largest eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_{D'}]$. In cases of very high dimensionality, one way to efficiently perform PCA is by using the *Karhunen–Loève transform*.

We can define the number of principal components we want to use to represent the data in lower dimension as a set amount of variance observed, however this approach has been shown to behave poorly on noisy data (Delis et al. 2013). Instead, we use enough principal components to account for 90% of the variance observed in the data which was found to produce satisfying results for our data. This means we find a D' such that $D' \leq D$ and

$$\frac{\sum_{i=1}^{D'} \lambda_j}{\sum_{j=1}^{D} \lambda_j} \geq 0.9 \tag{3.4}$$

The output produced by PCA is used for changepoint analysis in the next step.

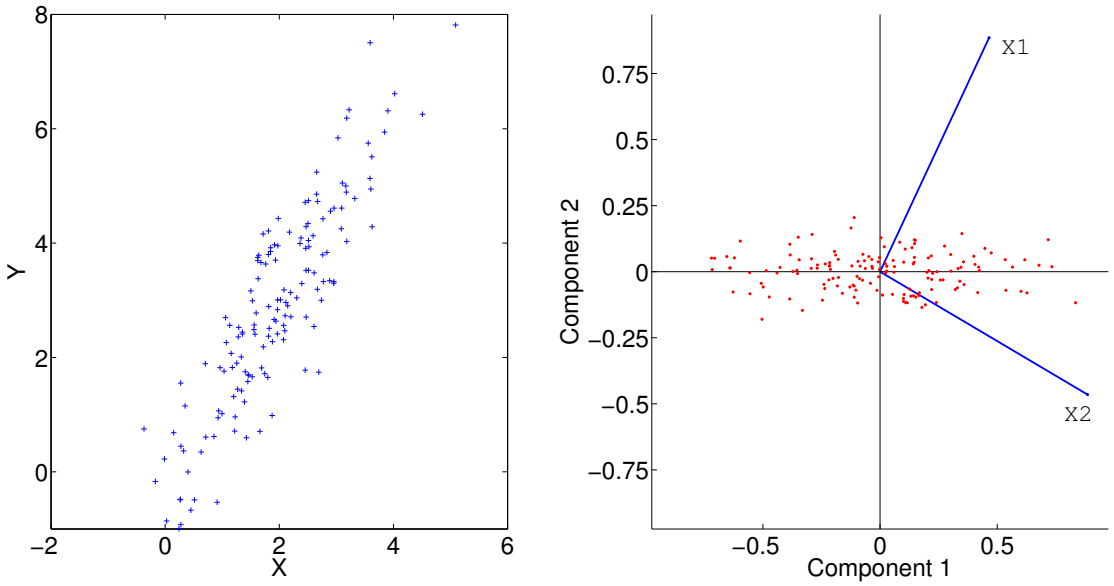


Figure 3.6: **Left**: 150 data points sampled from a multivariate Gaussian distribution. **Right**: Representation of data points in principal component space. In this case, the first principal component (X1) represents more than 95% of the data variability. PCA ensures that all further principal components (X2) are orthogonal to X1.

Changepoint Analysis

A *changepoint* defines abrupt variations in the time-series or alternatively in the generative parameters of sequential data. We developed a Bayesian algorithm for detecting such changepoints that is suitable for very long and high dimensional time-series and forms the core of our proposed method. Here, changepoint detection is applied to to movement data, however it could be slightly adapted and used in a variety of pattern recognition contexts, including the analysis of financial data, EEG data and DNA segmentation.

The algorithm is derived from an implementation suitable for analysis of univariate time-series by Adams and MacKay (2007) which is based on the student's t distribution and scales quadratically in computing time and memory. In this section, we will present several computational issues of the original algorithm when applied to complex movement data and show how we adjusted the implementation for processing high-dimensional neurological data.

When observing a data stream of points x_t $(t = 1, \ldots, T)$, it is assumed that the data in each segment are i.i.d. variables sampled from a probability distribution $P(x_t|\Theta_i)$ with parameter set Θ_i. The algorithm computes the posterior probability over all possible *run-lengths*, that is the time that has passed since the last changepoint at any time step

t

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \tag{3.5}$$

where r_t describes the current run-length and $x_{1:t}$ – the data observed up to the time point t – arises from a multivariate Gaussian distribution $x_{1:t} \sim \mathcal{N}(\mu, \Sigma)$.

The joint distribution over run-length and observed data is given by (Adams and MacKay 2007)

$$P(r_t, x_{1:t}) = \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t}) \tag{3.6}$$

$$= \sum_{r_{t-1}} P(r_t, x_t|r_{t-1}, x_{1:t-1})P(r_{t-1}, x_{1:t-1}) \tag{3.7}$$

$$= \sum_{r_{t-1}} P(r_t|r_{t-1})P(x_t|r_{t-1}, x_t^{(r)})P(r_{t-1}, x_{1:t-1}) \tag{3.8}$$

and the unnormalised evidence for $x_{1:t}$ is defined as

$$P(x_{1:t}) = \sum_{r_t} P(r_t, x_{1:t}) \tag{3.9}$$

Essentially, at every time step we are interested in the most likely run-length r_t of the current segment which is given by

$$r_t = argmax_k(P(x_t|\Theta_k) \mid k = 1 \dots t) \tag{3.10}$$

where Θ_k is the maximum likelihood estimate of the parameters given the last k number of data points. At every time step, the run-length can either increase by 1 and $r_t = r_{t-1} + 1$ or a changepoint occurs and $r_t = 0$.

Furthermore, a prior distribution over changepoint appearance is introduced which is implemented using a hazard function, where $H(\tau) = 1/\lambda$ defines a exponential distribution that is memoryless over time (Adams and MacKay 2007). Alternatives such as using a linear, step function or a Gamma hazard function are possible here. Our method assumes that a changepoint appears at time 0, that is $P(r_0 = 0) = 1$. Alternatively a prior distribution over initial run-length can be used instead.

We first had to convert the univariate implementation provided by Adams and MacKay (2007) to a multivariate Gaussian version which can deal with D-dimensional input vectors x such that

$$x \sim \mathcal{N}(\mu, \Sigma) \tag{3.11}$$

where μ is a $D \times 1$ mean vector and Σ a $D \times D$ dimensional covariance matrix.

The adaptation was slightly more complicated than initially anticipated and particular attention had to be paid to the initialisation and updates of the parameters of the multivariate Gaussian distribution. We found that different initialisation and update rules for mean vectors and covariance matrices had a significant impact on the final result produced by our algorithm. In this section we discuss the rules that yielded the best results on our data sets and briefly hint at possible alternatives.

Firstly, the parameters of the multivariate Gaussian distribution are initialised as follows:

$$\mu_0 = \mu_{prior} \tag{3.12}$$

$$\Sigma_0^{-1} = \Sigma_{prior}^{-1} \tag{3.13}$$

$$\kappa_0 = 1 \tag{3.14}$$

where D is the dimensionality of the input data stream and Σ^{-1} a precision matrix which is defined as the inverse of the covariance matrix. It has to be ensured that the covariance matrix is non-singular and is initialised such that non-zero probabilities are obtained when run-length distributions are computed. We defined Σ_{prior}^{-1} to be the inverse identity matrix scaled by a constant factor. Alternatively, the initial covariance parameters can be sampled from an Inverse Wishart distribution or estimated over part of the data set in question.

At each iteration we update the parameters according to

$$\mu_{t+1}^{(0)} = \mu_{prior} \tag{3.15}$$

$$\mu_{t+1}^{(1:t+1)} = \frac{\kappa_t \times \mu_t^{(0:t)} + x_t}{\kappa_t + 1} \tag{3.16}$$

$$\Sigma_{t+1}^{-1(0)} = \Sigma_{prior}^{-1} \tag{3.17}$$

$$\Sigma_{t+1}^{-1(1:t+1)} = \frac{\kappa_t}{\kappa_t + 1}(x_t - \mu_0)^T(x_t - \mu_0) + \Sigma_t^{-1(0:t)} \tag{3.18}$$

$$\kappa_{t+1} = \kappa_t + 1 \tag{3.19}$$

where $\mu_{t+1}^{(1:t+1)}$ denotes elements 1 to $t+1$ of the mean vector at time $t+1$. Note again, the explicit use of the precision matrix in these equations.

When we tested the original design of the changepoint algorithm by Adams and MacKay (2007), we observed that the algorithm becomes intractable for very long one-dimensional time-series. We found that the algorithm computes the posterior run-length distribution for *all* possible run-lengths at *every* time step, resulting in a quadratic run-time complexity in the number of data points. All run-length probabilities were retained

over the entire computation which led to quadratic memory requirements, as well. For higher dimensional data and using multivariate Gaussian distributions the algorithm exploded both in time and space for time-series in excess of 10,000 data points. In the following we will present how we overcame this problem and reduced the algorithm's complexity to be *linear* over all data points, both in time and space.

Firstly, we reduced the space requirements by only retaining the entire run-length distribution for the *current* time step, instead of all previous time steps, and only storing the *most likely* run-lengths, instead of all possible run-lengths, for every previous time-step. This information was sufficient to determine changepoints and made the algorithm scale *linearly* in space with the number of data points.

Secondly, as previously mentioned, the original implementation calculated a probability distribution over *all* possible run-lengths $r = 1 \dots t$ for *every* time step t. This implies that at every time step we update t mean vectors and t covariance matrices and calculate t run-length probabilities under the multivariate Gaussian model. We discovered that at each time step t, we do not have to compute all possible run-lengths. Instead, it is feasible to compute only the run-lengths over a limited and fixed time window $t_w \ll t$ which, depending on the window size, can increase the performance of the algorithm significantly up to an order of magnitude. One may be tempted to gradually decrease the window size to further reduce run-time costs, however it should be noted that a careful choice of t_w is required. The window has to contain enough data points to provide an accurate estimate of the current statistics. For the tests presented later in this report, we have to ensure that t_w is larger than the number of free parameters in the statistics used. In the case of a multivariate Gaussian distribution, this means that we have to consider at least $2 \times D$ data points for estimating the covariance matrix and D data points for the mean, which in total adds up to $t_w > 3 \times D$ data points, where D is the dimensionality of the data. In practice, we observed that t_w has to be larger than $3 \times D$ in order to give accurate results, however this is the theoretical minimum to avoid overfitting.

Extraction of Change Points

The ideal behaviour of the changepoint algorithm as shown in figure 3.7 is defined to be a linearly increasing run-length followed by a drop to zero, at which time step a changepoint occurs (Adams and MacKay 2007). This is however rarely observed in the case of using a multivariate distribution and calculating run-length probabilities over more complex data. This is assumed to be due to *noise* and *non-stationarity* in the data which induces uncertainties in the run-length estimates produced by the algorithm. A typical output produced by our algorithm as shown in figure 3.8/**Middle** highlights the need for additional processing to extract the most relevant changepoints for segmentation. We therefore extract the most significant changepoints by defining a changepoint as a point where the maximum run-length probability drops significantly (not necessarily to zero), where the drop significance is calculated with a distance function (Fink and Gandhi

2011):

$$dist(r_{t-1}, r_t) = \frac{|r_{t-1} - r_t|}{|r_{t-1}| + |r_t|} \tag{3.20}$$

which is bounded between 0 and 1. The extraction of changepoints becomes more coarse as the threshold above which we considered a drop in run-length as significant is increased.

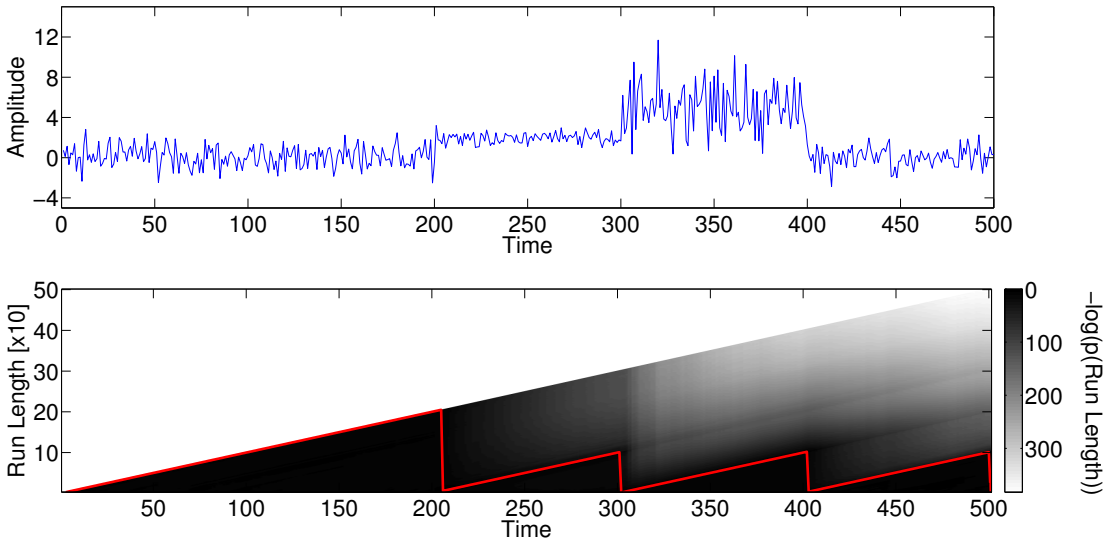


Figure 3.7: **Top**: time-series with changepoints at time-steps 200, 300 and 400. **Bottom**: Run-length distribution $P(r_t|x_{1:t})$ computed by the univariate algorithm presented in Adams and MacKay (2007). Gray shading represents probability distribution over all possible run-lengths at every time-step. Red line indicates most likely run-length at every time-step.

During this step, we further made use of our prior distribution over segment length which provides a more accurate estimate of changepoint appearance within a time-series.

At this stage, we have computed a set of significant changepoints in our observed time-series and have extracted the individual segments between the changepoints from the original data. We are now ready to perform comparisons on pairs of segments and generate clusters that group highly similar movement segments together.

Resampling and Comparison of Segments

Having obtained the estimated movement segments, we can now calculate similarities between all possible pairs of segments. One of the difficulties when testing for similarity between two natural movement segments is the difference in speed at which the respective movements are performed. It has to be taken into account that closing a hand can

be done very quickly and rather slowly, yet representing the same type of movement. To be able to account for different speeds at which we can execute movement, we resample every segment to a default movement duration. This allows us to compute similarity values between two segments of the same length with respect to their actual type. Having calculated the similarity between all possible pairs of segments, we generate movement clusters of segments with the same type. We define two movement segments to be the same if their similarity value is above a defined threshold. By varying this threshold we can produce clusters of different breadth, i.e. the segments contained within a cluster are less similar to each other. In an extreme case all segments will be contained by one cluster.

In the course of this project, we experimented with different types of statistical measures to compute the similarity between two segments. Their accuracy was validated by visualising movement segments on a computer screen (Figure 3.9) and comparing an expert's visual assessment with the values return by the statistical measures. We discovered that the use of different measures affected our results significantly and we therefore discuss our findings with respect to different types of measures in section 3.2.2. Before this, we will briefly explain our clustering procedure.

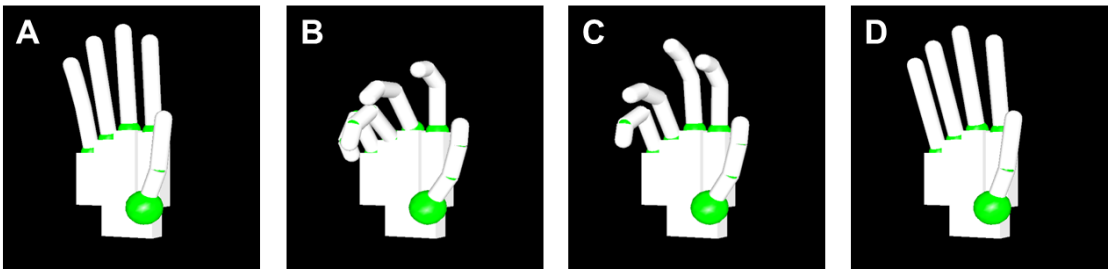


Figure 3.9: (**A-D**): Visualisation of two movement segments that were observed and recorded. The subject closes the hand (**A-B**) and opens it again (**C-D**). Our method finds a changepoint between (**B**) and (**C**) and we obtain two segments.

Clustering

In our analysis we discard both the first and the last segment of each movement sequence as we found that these were often incomplete or erroneous due to calibration and initial misalignments of the hand. As a result these would distort the overall distribution of the clusters without adding any valuable information. In addition to resampling all segments to a default length, we normalise all segments to have amplitudes in the range $[-1, 1]$ by dividing each segment's amplitude by its absolute maximum. We then compute the similarity between every possible pair of segments using, e.g. using Pearson correlation coefficient or maximum cross-correlation. When calculating cross-correlation, we defined a time-shift window of 20% over which the values are calculated to correct slight misalignments between two segments.

Following this, the clusters are computed using an iterative procedure as described in algorithm 1. We start with an arbitrary segment s_1 and find all other segments s_o for which the similarity value between s_1 and s_o is larger than a given threshold t. This step is repeated until the cluster has converged to a stable state.

Algorithm 1: Pseudocode of iterative clustering algorithm

Input: m-by-n binary matrix dMat
Output: *clusters*
$m \leftarrow$ number of rows in dMat;
$n \leftarrow$ number of columns in dMat;
$not\_checked \leftarrow$ n-by-1 array of 1s;
$clusters = \oslash$;
for $seg \leftarrow 1$ **to** n **do**
  if $not\_checked[seg] == 0$ **then**
    continue;
  end
  $tmpSegs \leftarrow seg$;
  $oldSegs \leftarrow 0$;
  while $length(oldSegs) \neq length(tmpSegs)$ **or** $\exists k(oldSegs[k] \neq tmpSegs[k])$ **do**
    oldSegs = tmpSegs;
    $\forall j(dMat(tmpSegs, j) > 0 \rightarrow dMat(tmpSegs, j) \in tmpSegsNew)$;
  end
  $not\_checked[seg] \leftarrow 0$;
  $clusters = clusters \cup tmpSegsNew$;
end

We verified each cluster with visual inspection of the movement segments on a computer screen. It should be mentioned that one slight drawback of this method is the creation of a number singleton clusters where the segment is too different from any other segment. We consider these segments as truly unique.

3.2.2 Measuring Similarity between Movement Segments

Mathematically, a statistical measure describes the similarity between two different sets of variables. In our case we are interested in comparing pairs of time-series describing movement behaviour. More specifically, we would like to ascertain to which extent two time-series represent the same behaviour. We therefore have to ask ourselves first what it means for two movement segments to be equal since we can, for example, imagine a very quick closing and in contrast a very slow closing of a hand, which makes this question not trivial to answer. In addition, a subject can close his/her hand in slightly different ways making it even more difficult to test for similarity.

As for this project, we want to account for different speeds at which movements are performed and slight time shifts between movements occurring within a movement seg-

ment. By varying the threshold above which two segments are considered to be equal, we can deal with movement that is performed slightly differently, but still represents the same type of behaviour. In the following, we will briefly mention the similarity measures that we have tested in combination with the methods presented here.

Euclidean Distance

We can compute the similarity of two D-dimensional movement segments X and Y by calculating

$$d(X,Y) = \sqrt{\sum_{i=1}^{T} (X_i - Y_i)^2} \tag{3.21}$$

for each dimension in D and take the mean over all dimensions. We found that, when comparing two segments using Euclidean distance, the indications whether or not two segments were similar were not very accurate. The rather poor performance of Euclidean distance as a similarity measure for movement sequences is assumed to be due to the high dimensional nature of the data. In addition, it does not take slight time shifts between the movement sequences into account and only considers the distance between the respective dimensions.

Pearson Correlation Coefficient

As an alternative to the Euclidean distance measure we use Pearson correlation coefficient to calculate similarity between two movement segments. It is defined as

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{3.22}$$

where X and Y denote two movement segments again. We found that, when applied to movement segments and verified by visualisation, Pearson correlation coefficient provides a more accurate estimate of similarity between two movement segments than Euclidean distance. It appears to deal more accurately with high-dimensional movement data but does not account for scaling and time shifts between the two segments.

Goodness of Fit

Goodness of Fit (GoF) between two time-series X and Y can be defined as

$$gof_{X,Y} = 1 - \frac{var(X-Y)}{var(X)} \tag{3.23}$$

We additionally introduced a weighting over all dimensions in our GoF measure to discriminate against variables with low variance. This becomes important when our data contains a significant amount of noise. In contrast to Pearson correlation coefficient, the GoF measure takes scaling of the time-series into account but does still not consider time shifts.

Maximum Cross-Correlation

We found that even slightest time shifts between two segments affected the accuracy of the similarity measures presented so far significantly. They were therefore not always useful for comparison of two movement segments and we often obtained more accurate comparison results when using cross-correlation as a similarity measure. Cross-correlation is a measure of correlation between two time-series as a function of a time-lag applied to one of them.

$$\phi_{X,Y}(t) = \int_{-\infty}^{\infty} X(\tau - t)Y(\tau)\,\mathrm{d}\tau \tag{3.24}$$

where τ is a given time-lag parameter and two segments X and Y. We calculate it over a minimum and maximum time-lag corresponding to 20% of the segment length to correct for time-shifts within two segments. We then define the similarity between two segments as the maximum value returned by cross-correlation. One should note that this measure takes all possible shifts within the time-lag window into account and is therefore computationally more expensive than the measures presented previously.

3.2.3 Evaluation

We tested our method on two different types of data; an artificial data set and hand-movement data of multiple subjects performing various everyday tasks.

Artificial Data Set

For the purpose of this work, we generated an artificial data set by repeatedly sampling from the family of functions defined as

$$f = \left(\sum_{i=1}^{N} w_i \phi(t - t_i) \right) + \epsilon \tag{3.25}$$

where $w_i \sim \mathcal{N}(0, I_D)$, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, Φ were chosen to be Gaussian basis functions, N is the number of basis functions, D the dimensionality of the data and t_i can be arbitrarily chosen to constrain the function to a given interval. The segments obtained were then resample to a length which was sampled from $\mathcal{U}(100, 150)$ and randomly concatenated to generate a data set.

Our method was validated on data of varying dimensionality, number of different segment types, and noise levels. Additionally, we changed the parameters of our segmentation method throughout the tests. We measured performance of our method by calculating similarity between the original and extracted segment sequence. The results presented here were produced using Pearson correlation coefficient.

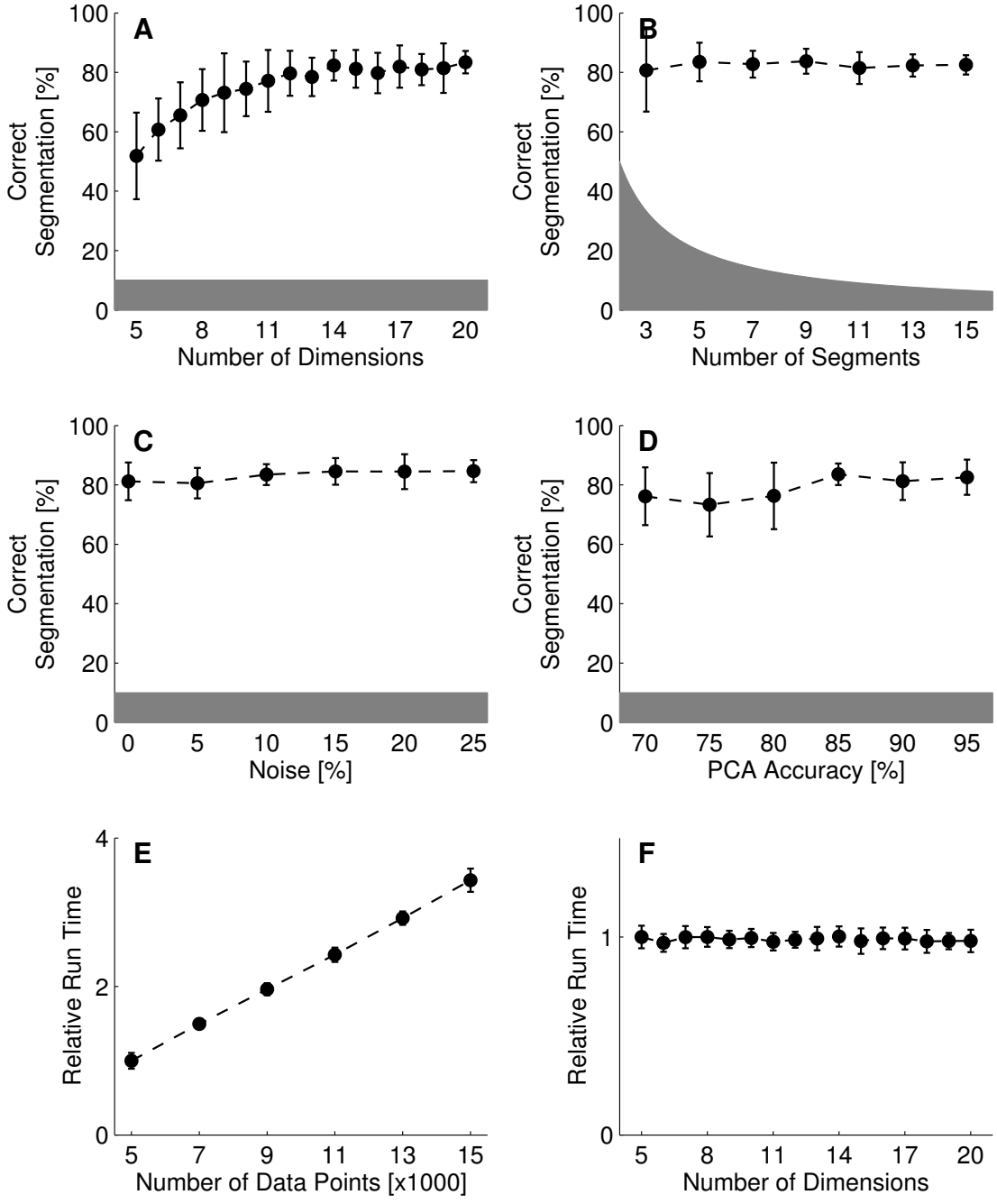


Figure 3.10: **(A)** - **(D)**: Influence of parameters on extraction accuracy. **(A)** dimensionality, **(B)** number of different segments, **(C)** noise levels, **(D)** PCA accuracy. **(E)** - **(F)**: Influence of number of data points **(E)** and number of dimensions **(F)** on run time. Markers indicate mean with standard deviation and grey area represents levels of chance. If no errorbars are shown, they are smaller than the marker. Note that the run-lengths have been normalised to the shortest run-length. See text for details.

Our testing results are shown in Fig. 3.10. It is revealed that our method is extremely robust to most variations in the parameters of the algorithm and data, consistently showing very high accuracy (on average 86.5% when using Pearson correlation coefficient as similarity measure) and no significant decline in performance as the number of different segment types (Fig. 3.10/**B**) or the level of noise (Fig. 3.10/**C**) is increased. The only drop in performance was observed when analysing data of lower dimensionality. This was found to be caused by the PCA step which is performed to reduce the dimensionality of the data. In cases where the original data dimensionality was already low, PCA often reduced the input to one-dimensional data which makes changepoint detection generally hard as the statistics taken into account become very similar across all segments. This decrease in performance can be easily overcome by introducing a lower bound on the number of output dimensions of the PCA step.

We tested the segmentation accuracy of our method by analysing how many different segment types our algorithm extracts from a given data set. This was done by generating clusters of segments that were very similar to each other based on the segments extracted from the input data. These clusters allowed us to identify unique movement types within the extracted segments. We could then compare these unique movement types against the unique segments in the original data and estimate the extraction accuracy of our algorithm. Under Pearson correlation, we defined two segments to be the same if their coefficient was larger than 0.7. This threshold was verified using cross validation and visual inspection of movement segments on a computer screen.

K	**K Extracted**	**Correlation Coefficient**
3	4.60 (± 2.19)	0.81 (± 0.24)
5	6.40 (± 1.48)	0.86 (± 0.18)
7	8.53 (± 1.94)	0.87 (± 0.18)
9	10.60 (± 2.19)	0.85 (± 0.18)
10	11.71 (± 1.84)	0.87 (± 0.17)
11	12.77 (± 1.72)	0.87 (± 0.17)
13	13.83 (± 1.21)	0.88 (± 0.15)
15	14.77 (± 1.79)	0.89 (± 0.15)

Table 3.1: Performance measures for proposed method on artificial data set. K is the number of different *types* of segments in the input data. Values indicated are mean (± SD).

In table 3.1 we compare the number of unique segment types in the input data against the number of different segment types found by our algorithm. We also present the average correlation coefficient found between the original and extracted segments. It can be seen that the number of estimated unique segments is always slightly higher than the number of real segments in the input data. This can be explained by considering that any segment that is "incorrectly" extracted will form a cluster by itself and therefore increase this estimate. Considering that the average number of different segments types

in our data set was 40, we found that our algorithm performs very well, as on average it only finds 1.2 segments more than segments present in the input data.

We were also interested in testing how well our algorithm scales with increasing sequence lengths and number of dimensions. Figures 3.10/**E** and **F** show the relative run-time normalised to the smallest input data set (5000 data points for **E**, 5 dimensions for **F**). It can be seen that our method scales *linearly* with the number of data points and is *constant* in the number of dimensions. The latter is mainly due to the PCA step which ensured that the data dimensionality for changepoint detection was kept relatively low.

Neurotechnology for Kinematics in Daily Tasks

The algorithm was also tested on hand movement data collected during experiments in Belic and Faisal (2011). In these experiments, a left-handed CyberGlove I (see section 2.4 for details on CyberGlove) was worn by subjects who were asked to perform 10 repetitions of 17 different tasks which occur frequently in daily life. The tasks included opening doors, drinking, eating (with and without cutlery) and interaction with common items such as computers and phones. The total length of the data was above six hours, or approximately 1.7 million data points. Using a second-order Savitzky-Golay filter and a running window of 5 data points the data was smoothed before any further analysis to remove discontinuities induced by the glove's A/D converter.

We applied our changepoint algorithm to the data collected by Belic and Faisal (2011) and obtained 3710 segments with a mean run-length of 263 data points which is equivalent to a mean movement duration of 3.28 seconds. It should be noted that when segmenting a movement time-series we always discard the first and the last segment as we found that these are often incomplete or erroneous due to calibration given the data set used here.

Similarly to the tests on the artificial data set, we calculate the similarity between all possible remaining segments by resampling every segment to a default length (see figure 3.11 for a graphical representation of similarity between all pairs of segments within a task) and group similar segments into clusters. We were then interested in investigating whether (1) the segments within a cluster are indeed similar, (2) the singleton clusters are indeed different from all other segments and (3) which movement clusters are large and can therefore interpreted as containing segments that are performed very frequently in the everyday tasks performed.

For (1)-(2), we used the maximum value indicated by cross-correlation over all dimensions (the results with respect to other similarity measures are discussed in section 3.2.2). We visualised the movement segments on a computer screen and discovered that the segments within a cluster were indeed similar. We could control the "breadth" of such a cluster by varying the threshold above which two segments are considered equal and subsequently clustered together. Using a lower threshold, the clusters contained segments of larger variability, and vice versa for a larger threshold. We present the clusters that we obtained with different thresholds in tables 3.2, 3.3 and 3.4. Note that even under a 0.90 threshold, the seven largest clusters account for more than 1.26% of all data

Cluster Id	Cluster Size	Type
161	94	Half opening
509	72	Full hand opening with straight ring-finger
425	29	Half closing
2673	22	Half opening distal and proximal interphalangeal joints
1	15	Full closing
1207	13	Inactive, straight hand
1249	13	Quarter opening with almost straight fingers
...	...	...

Table 3.2: Largest clusters computed with a threshold of 0.85 when using maximum cross-correlation. The segments in these clusters account for 6.95% of all segments found (total number of segments is 3710) and for 5.22% of all data points in the set (total number of data points is 975112).

points in the data set. This is a significant proportion given that the clustering algorithm produced 2792 clusters when the threshold was set to 0.85, 3044 when set to 0.875 and 3281 clusters when the threshold was 0.9.

For (3), we observed that the largest clusters consistently represented the same actions. A cluster containing segments that describe the opening of a hand was always found to be the largest, with 94 segments under a 0.85 threshold, 50 under 0.875 and 15 when the threshold was set to 0.90. By lowering the threshold we obtain larger clusters that contain segments less similar to each other. In this case we can, for example, observe different forms of opening an hand in the same cluster. In contrast, by raising the threshold, the number of segments within each cluster becomes smaller and the movement segments more similar. These results were again verified by visual inspection on a computer screen (see figure 3.9 for an example).

We tested our methodology on a data set containing real movement data collected during everyday tasks. Our algorithm was able to accurately and efficiently process a data set containing roughly one million data points in 18 dimensions. Despite the repetitive nature of the tasks, this data set was particularly useful for testing. The results demonstrate that a large proportion of segments cluster around a small number of different actions which all represented some form of hand opening and closing which emphasises the importance of the grasp movement in many situations.

Lower Paleolithic Stone Toolmaking

We tested our method on a third data set which was recorded during the replication of Lower Paleolithic stone tools (Faisal et al. 2010). Specifically, in these experiments, the

Cluster Id	Cluster Size	Type
223	50	Three-quarter opening
549	29	Full hand opening with straight ring-finger
460	16	Half closing
1	15	Full closing
1303	14	Half opening
1326	13	Inactive, straight hand
1128	12	Closing distal and proximal interphalangeal joints
...	...	...

Table 3.3: Largest clusters computed with a threshold of 0.875 when using cross-correlation. The segments in these clusters account for 4.02% of all segments found (total number of segments is 3710) and for 3.15% of all data points in the set.

production of the Oldowan flake and Late Acheulean handaxe (see figure 3.12), for which historical evidence was found at the beginning and end of Lower Paleolithic period, was studied. This period covers more than 2 million years of human history and is often of great interest to archaeologists and evolutionary researchers.

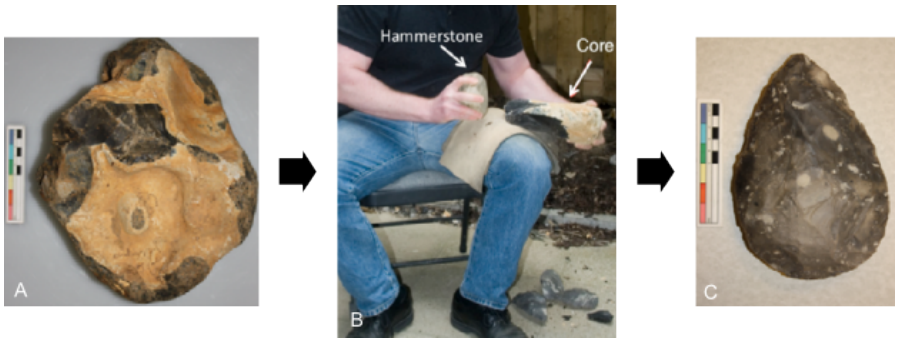


Figure 3.12: **(A)**: A core stone is used for reproduction and struck with a hammerstone **(B)** to form the shape of a Acheulean handaxe **(C)**. Individual figures taken from Faisal et al. (2010).

Here, we considered the movement time-series which was obtained from the CyberGlove I during one of the experiments where a Late Acheulean handaxe was reproduced (see figure 3.13/**Top**). The motion glove was worn underneath a garden glove and approximately 112,000 data points which represent roughly 22 minutes worth of data were recorded. The initial recording frequency was 80 Hz, however the data was later combined

Cluster Id	Cluster Size	Type
594	15	Full opening with straight ring-finger
490	14	Half closing
1	12	Full closing
1217	12	Half opening
1410	11	Half opening with straight index-finger
1952	11	Opening distal and proximal interphalangeal joints
2485	9	Closing distal and proximal interphalangeal joints
...	...	...

Table 3.4: Largest clusters computed with a threshold of 0.90 when using cross-correlation. The segments in these clusters account for 2.26% of all segments found (total number of segments is 3710) and for 1.26% of all data points in the set.

with other recordings and during this process resampled to 150 Hz.

As with the data collected by Belić (2010), we first smoothed the data during a pre-processing step using a second-order Savitzky-Golay filter and a running window of 5 data points. The subsequent analyses were then carried out on the first derivative, i.e. velocity values, of the smoothed input data.

Having computed the run-length distributions over all 112,000 time steps and having performed segmentation based on the calculated changepoints, we obtained 142 segments with a mean length of 663.23 (± 266.53) data points. This is equal to a mean duration of 4.42 (± 1.78) seconds. A video was taken during the reproduction of the tool and we could verify agreement between the segments computed by our method and the hand movements seen in the video.

Four extracted segments are shown in figure 3.13/**(A-D)**. From the video we were able to identify the changepoints **(A-B)** as changing the position of the stone, changepoint **(C)** as a flip over, and changepoint **(D)** as a grasp followed by picking up the stone. We further visualised the segments on a screen and present screenshots in figure 3.14/**(A-D)** accordingly.

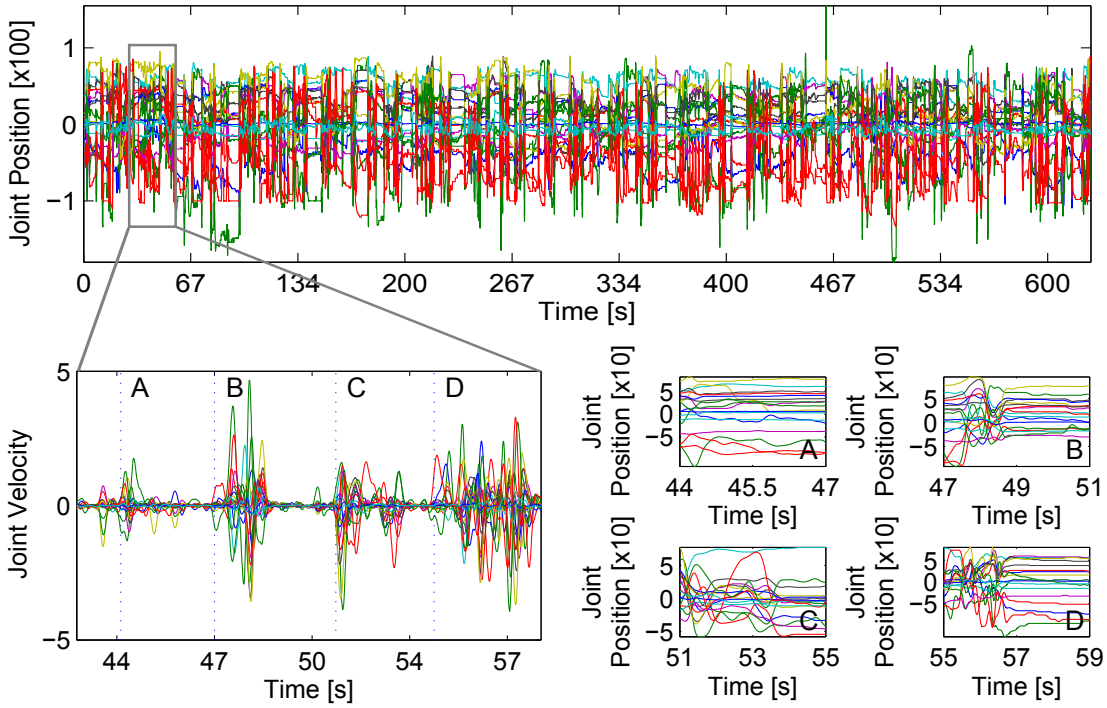


Figure 3.13: **Top**: Movement data collected while motion glove was worn during flint knapping. **Bottom:** Four segments are extracted between seconds 44 and 59. Changepoint detection is p on the joint velocity values.

We were also interested in seeing to which extent the movement segments computed in the previous step are similar to each other. Similarly to our previous tests, we resampled all movement segments to a default length and computed the similarity between all pairs of segments using maximum cross-correlation and Pearson correlation coefficient after having removed all segments with very low variability. Clustering was subsequently performed on the remaining 82 segments from which we generated 60 clusters with a threshold of 0.85 for maximum cross-correlation and 70 clusters with threshold 0.5 under Pearson correlation. It should be noted that, as previously, we obtained many singleton clusters, however in both cases we found two clusters which accounted for 11.30% (maximum cross-correlation) and 9.87% (Pearson correlation) of all data points. We finally visualised the segments within the respective clusters and found that the hand configuration as shown in figure 3.14/**(A)** and 3.14/**(D)** were present in the two largest clusters.

This data set was particularly interesting to us as it did not contain any predefined task repetitions and contained more irregularities than the data sets we had tested previously which made segmentation an even more challenging task. This test revealed that our method manages to produce accurate results even in situations where the patterns in the time-series are less repetitive and the hand movement behaviour varies significantly

between segments.

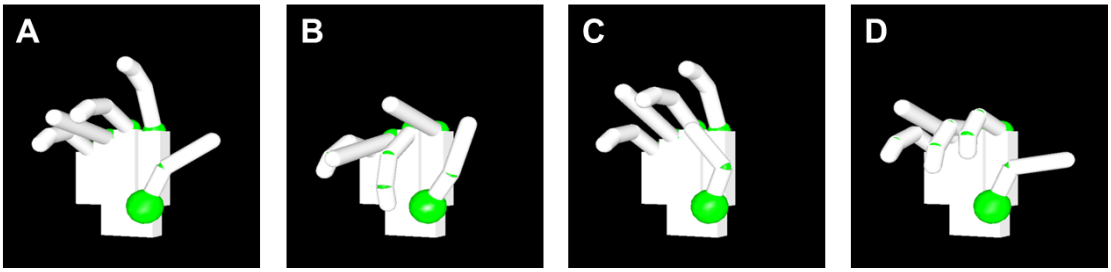


Figure 3.14: Extracted segments visualised on a computer screen. In **(A-B)** the position of the stone is changed, during **(C)** the stone is flipped over and in **(D)** a firm grasp is applied to the stone while being picked up.

3.2.4 Conclusion

We have shown how, by exploiting the covariance structure of the input time-series, Bayesian changepoint detection and temporal correlation, we can perform accurate and highly robust time-series segmentation over millions of high-dimensional samples in *linear* time and space with an increasing number of data samples. Our method is suitable for high-dimensional data streams as, mostly due to PCA, performance scales *constantly* in dimensionality. Here, we have successfully tested our proposed method on artificially generated data, a data set containing hand movements collected during everyday tasks and on movement data recorded during the reproduction of Paleolithic stone tools. We conclude that the run-time properties of our algorithm enable a Big Data approach to neuroprosthetics and brain-machine interfaces to become computationally tractable. We have presented our work in the context of movement data, however, with slight modifications, our implementation could be applied to a variety of types of data.

3.2.5 Future Work

We have successfully developed a method that allows us to accurately segment large movement time-series. Yet, we see a lot of possibilities for improvement for which we will briefly present our ideas in this section

Firstly, our method currently requires an additional pass over the computed run-lengths to extract all *significant* changepoints from the result produced by the changepoint detection algorithm. This is mostly due to the inaccuracies that are sometimes observed in the results of our changepoint detection algorithm. The current implementation uses this extra step to enforce a prior distribution over movement duration which is not considered properly by the changepoint detection algorithm in the first run. We suggest that this prior should be integrated into the changepoint detection step which upon successful integration would allow us to remove the currently required extraction step. We expect

that this would not only simplify the structure of the algorithm but also slightly increase the computational performance of our method.

Secondly, we would like to test further similarity measures and investigate whether we can incorporate an even more accurate measure into our method. One interesting model-based approach which was found but not tested during this project uses semi-continuous Hidden Markov Models (SC-HMMs) and mixture of Gaussians to infer similarity between vector sequences (Rodriguez-Serrano and Perronnin 2012). While this approach seems to be of higher complexity than the similarity measures tested during this project, it may achieve better results.

In some occasions we observed that our algorithm misses a changepoint in cases where the generative process between two segments does not change significantly. As a result it incorrectly combines two movement segments which, in figure 3.15 happens roughly between point 800 and 1000. This becomes particularly evident if the statistics changes from high to low variability in the data. In the case of the Paleolithic stone toolmaking described in section 3.2.3, abrupt changes in hand position (high variability) were often followed by longer resting period (low variability). Our method frequently missed a changepoint in these cases. We found that the structure of the covariance matrix used during changepoint detection influences this heavily, however further investigation is required here.

In this work, we only tested our methodology on an artificial data set and hand movement data. We would like to extend our method to be able to segment and recognise movement patterns in data obtained from a body suit (e.g. ANIMAZOO IGS-150[3]) or other potential non-optical motion capture devices. This would enable us to perform analyses and tests on even more complex data.

Finally, we would like to test our method on a wider variety of data sets that were collected during natural situations.

[3] www.animazoo.com/products/igs-150-range

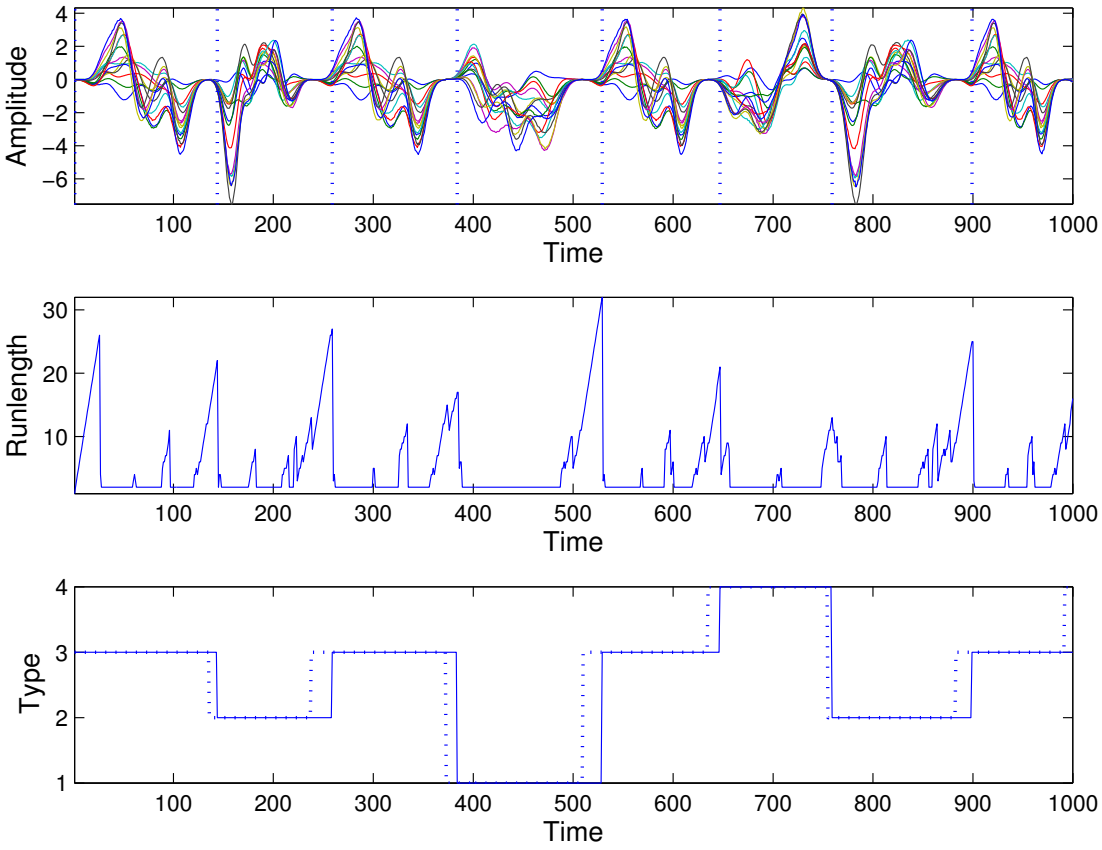


Figure 3.8: **Top**: Artificially generated 15-dimensional time-series which forms input to our algorithm. Computed changepoints are indicated as dotted lines. **Middle**: Computed run-length distribution. The plot shows the most likely run-length at every time step. **Bottom**: Segment classes of original data indicated by dotted line, of extracted data in straight line. Note the constant offset in time between original and extracted recognised classes which we correct by subtracting a constant offset. Here, all segments are correctly extracted and recognised.

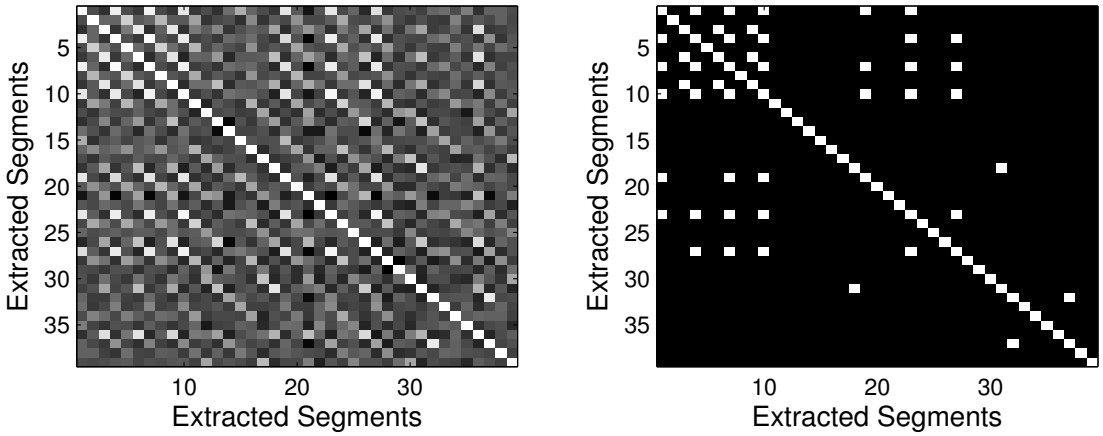


Figure 3.11: Comparison between all pairs of segments for repeated execution of the task "picking up the plastic bottle, simulating the drinking, and returning the bottle to the starting position". **Left**: Similarity based on maximum cross correlation between all possible pairs of the 39 segments found. The brighter the case, the more similar are the two segments. **Right**: All movement segments that we consider as equal using a maximum cross-correlation threshold of 0.85.

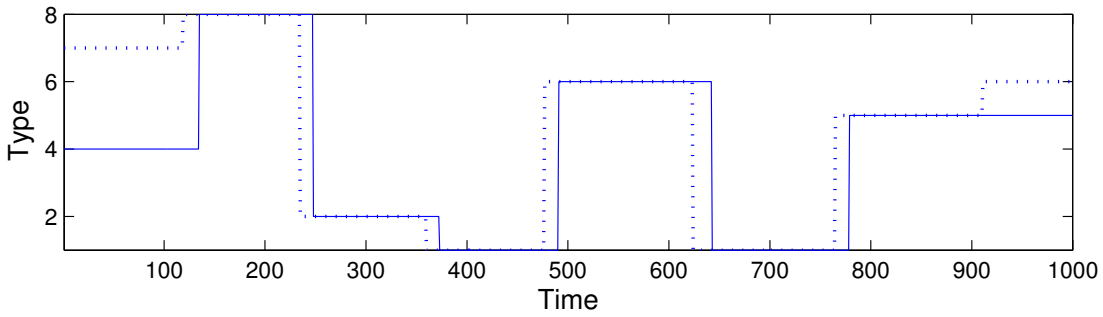


Figure 3.15: The computed segmentation is shown as a straight line. The dotted line indicates original segmentation. We can see a misclassification for the first segment and a missed changepoint between points 800 and 100.

4 Exploiting Movement Patterns for Improved Neuroprosthetic Control

Neuroprosthetics offer the chance for amputees to regain naturalistic mobility and dexterity by controlling a prosthetic device. Yet, even with the most advanced technology available today, the amount of useful control signals that we can extract from the brain or nervous system is still very low and not suitable for real-time prosthetic control (see section 2.3.2). Furthermore, with present methods we have not yet achieved the same versatility and dexterity as we can observe with natural limbs (Abbott and Faisal 2012). As presented before, these current limitations restrict the use of neuroprosthetic devices in everyday activities significantly.

In this chapter, we will propose the design of a novel neuroprosthetic control paradigm (Thomik, Haber, and Faisal 2013) which by observing the behaviour of intact limbs can accurately predict the behaviour of missing limbs (see figure 4.1). To achieve this we will exploit two aspects of natural movement: (1) modularity (see section 2.1), and (2) the stereotypical patterns between joints of fingers during everyday tasks (see section 2.2). We will develop a probabilistic and inference-based method that will use a two-stage approach and demonstrate that this method yields very high statistical accuracy under a large test data set.

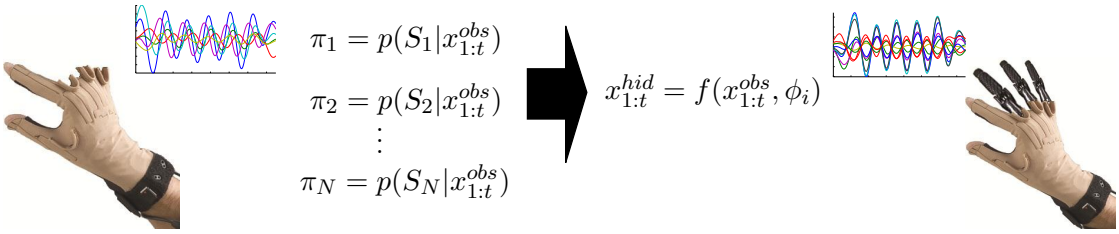


Figure 4.1: Conceptual design of the prosthesis controller. **Left hand side:** A limited amount of information is observed (here, thumb and index finger) and inference is performed to evaluate which movement cluster most likely generated the observed data. **Right hand side:** The parameters of the most likely movement cluster are used to estimate the missing dimensions and the prosthetic replacement is actuated accordingly. Figure from Thomik, Haber, and Faisal (2013)

4.1 Overview

The method presented here is based on a large data set that was also used for the evaluation of our segmentation method presented in section 3.2.3. The data set includes more than 6 hours worth of data from different subjects performing a variety of everyday tasks and was collected using a CyberGlove I motion capture device. We will show how we use this data and the efficient changepoint detection method presented in chapter 3 to generate clusters of similar movement segments. The clusters will subsequently allow us to accurately complement partial observations (e.g. movement of two fingers) and to predict movement behaviour of the unobserved joints (e.g. remaining three fingers) in *real-time* by taking into account the probabilistic structures of the most representative clusters.

Following this, we will demonstrate that we can significantly improve the prediction accuracy for the movement of unobserved joints with significantly less training data and in less time. This is achieved by applying our segmentation method first, instead of considering the entire unsegmented time-series, supporting a modular approach to prediction of unobserved movement behaviour.

4.2 Methodology

In the following sections we will explain our methodology that achieves accurate movement prediction of unobserved joints. We will briefly explain the steps involved and look at optimisations to further improve the performance of our method.

Segmentation & Clustering

We analyse and segment the data set using the method described in section 3.2.3 and Haber, Thomik, and Faisal (2013). We perform all analyses on the velocity values of the movement data. This was found to be more accurate as it explains movement behaviour more accurately than the spatial positions of the joints obtained from the CyberGlove. Using the first derivative of the data also ensures that all our movement segments have a mean close to zero which simplifies analysis.

Before generating a probabilistic representation of the data, we group all movement segments into clusters using the method we described in 3.2.1. Here, we compute the similarity between two movement segments using maximum cross correlation (see section 3.2.2 for definition) and two movement segments are considered to be equal if their maximum cross-correlation value is larger than 0.8. The time-lag used during calculation of cross-correlation was 20% of the segment length which allowed us to correct for slight misalignments of movement within the segments. Again, further details can be found in section and 3.2.1.

Inference

We next fit all generated clusters with a mixture of Gaussians to infer the probability that a given data point belongs to a cluster. A mixture of Gaussian model can be written as a linear sum of Gaussian components in the form (Bishop 2007)

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{4.1}$$

where K denotes the number of components in the mixture model, π_k are the mixture weights, x a D-dimensional input vector, μ_k is a D-dimensional mean vector and Σ_k a D-by-D covariance matrix for each component k ($k = 1 \dots K$) respectively.

In addition, the mixture model introduces the concept of *hidden* or *latent* variables, e.g. a K-dimensional binary vector z, which are unobserved quantities. We will see later that in our context, the latent variables will indicate cluster responsibilities for the data points that we have observed. We can write (Bishop 2007)

$$p(z_k = 1) = \pi_k \tag{4.2}$$

and have to ensure that all π_k are valid probabilities by defining the following constraints

$$\forall k (0 \leq \pi_k \leq 1) \tag{4.3}$$

and

$$\sum_{k=1}^{K} \pi_k = 1 \tag{4.4}$$

Using a mixture of Gaussian model we can then ask which cluster (indicated by latent variable z_k) is most likely responsible for having generated the observed data points x. For this, we first have to fit a distribution over the segments of each cluster generated. An often powerful and computationally efficient method for finding maximum likelihood solutions for the parameters of the models with latent variables is the *Expectation Maximization* (EM) algorithm (further details in Bishop 2007). This method iteratively estimates the values of the latent variables and subsequently maximises the likelihood of the parameters of the probabilistic model until convergence.

It should be noted that other probabilistic models can be used here, however a mixture of Gaussians appears to be suitable as we later want to make statistical inferences about the properties of the *individual* segments by only taking into account the *overall* pooled properties of each cluster

We choose the optimal number of components for our mixture models by computing the Bayesian Information Criterion (BIC) which is often used to select the best model among a finite set of possible models. BIC also avoids overfitting by introducing a penalty term for the overall number of parameters in the model.

The likelihood of the data at time t being generated by a particular cluster S_i is finally given by

$$\pi_i = p(S_t^i|x_{1:t}^{obs}) = \frac{p(x_{1:t}^{obs}|S_t^i) \times p(S_t^i)}{p(x_{1:t}^{obs})} \tag{4.5}$$

where $p(S_t^i)$ can be estimated directly from the training data and $x_{1:t}$ is the data which we have observed so far.

Furthermore, we use a discount factor $df(t) = 1/t^\tau$ where τ is a constant and t a given time-step to discriminate exponentially against past data points. This ensures that more recently observed data points have a larger influence on choosing cluster S_t^i.

At this stage, using π_i we can now infer which cluster should be used for predicting the unobserved dimensions in the next step.

Prediction

To perform prediction of the unobserved dimensions, we fit a regression under a linear combination of Gaussian basis functions between the O observed dimensions x^{obs} and H unobserved dimensions x^{hid} for each cluster offline. The model can be defined as

$$x^{hid}(x^{obs}, w) = w_0 + w_1\phi_1(x^{obs}) + w_2\phi_2(x^{obs}) + \cdots + w_{M-1}\phi_{M-1}(x^{obs}) \tag{4.6}$$

$$= w_0 + \sum_{j=1}^{M-1} w_j\phi_j(x^{obs}) \tag{4.7}$$

where M is the number of basis functions, here chosen to be at most 50, and our Gaussian basis functions for each component i are defined to be

$$\phi_k(x^{obs}) = exp\left(-0.5 \times \frac{(x^{obs} - \mu_k)(x^{obs} - \mu_k)^T}{100\sigma}\right) \tag{4.8}$$

where μ_i is a O dimensional mean vector and σ a $O \times O$ dimensional covariance matrix which is shared amongst all basis functions. Note the constant factor in the denominator is used to widen the Gaussian functions which allows for more robust inference.

The number of such basis functions has to be chosen carefully to avoid overfitting of the parameters due to a large number of degrees-of-freedom. In addition and as previously done, all components share the same covariance matrix which we found gives numerically more stable results during prediction. A pooled covariance matrix is often also used in situations where it is not guaranteed that we observe enough data points to obtain an non-singular estimate for $M - 1$ covariance matrices, but rather only for one covariance matrix.

Upon observing partial movement we then (1) select the most likely cluster from π_i in a winner takes all fashion and (2) estimate the unobserved dimensions on the basis of the regression parameters fitted to that particular cluster using $x^{hid}(x^{obs}, w)$. It should be noted that during this step, the computational run-time is only dependent on the number of clusters that were generated from the extracted segments.

4.3 Evaluation

In this section we present the results of the segmentation, clustering and prediction steps of our proposed method. We then show that by exploiting the unique characteristics of each movement cluster we are capable of predicting the unobserved data much more precisely than by solely taking the entire, unsegmented movement data into consideration.

Obtaining Movement Segments

When segmenting a block of data we removed the first and last segments from the sequence as we often found that these were incomplete or erroneous due to calibration. We obtained 3710 segments with an average movement duration of 3.28 seconds. We subsequently performed clustering of all movement segments as described in section 3.2.1. Using maximum cross-correlation and a threshold of $t = 0.8$, 2279 clusters were generated. It should be noted that a vast majority (88.85%) of these clusters were singleton clusters. Singleton clusters only contain a single segment for which the similarity with any other segment was less than t. We consider such segments as truly unique within our data set.

To limit the influence of the large number of singleton clusters and to simplify the procedure of estimation the regression parameters we generate a prior distribution over cluster size and only consider clusters which account for at least 0.2% of the total number of data points collected. From the 2279 clusters that were produced by our clustering algorithm, only 34 clusters were eventually used to predict the unobserved data dimensions. One should note that with this we take 1.5% of the number of clusters into account but consider 25.55% of all data points in the data set.

Prediction

We estimated the unobserved data during a two-step process in which we (1) select the movement cluster which has most likely generated the data observed and (2) estimate the unobserved dimensions using linear regression which we had fitted to that particular cluster.

We first observe that the statistical nature of our inference approach does not guarantee that we always choose the correct cluster for prediction for consecutive time steps. In some occasions the currently most likely cluster changes between time-steps t and $t+1$ which leads to considerable changes in our estimate for x_t^{hid} and x_{t+1}^{hid}. We overcame such abrupt mistakes by fitting an exponential distribution

$$f(d, \lambda) = \frac{1}{\lambda} e^{-\frac{d}{\lambda}} \tag{4.9}$$

where we define d as the absolute distance between two time-steps $d = |\hat{x}_t - \hat{x}_{t+1}|$ of the unobserved data in our training set and λ the rate parameter which we obtained by fitting the distribution over the entire training data. If the change between any two data points had a likelihood less than 0.1%, we assumed that the inference had failed and repeated the inference on the most likely movement cluster with the current estimate discarded. This

successfully prevented the erroneous jumps in the estimate of the hidden dimensions and additionally refined the inference step for the underlying movement primitive, yielding more accurate predictions for the unobserved dimensions.

Using this method, we were first interested in testing two different scenarios (see figure 4.2): (1) only the *thumb* and *index finger* were observed and the movement of the *middle*, *ring* and *little* finger was reconstructed (Figure 4.2/**A**) and (2) we observe *thumb*, *index* and *middle* finger and the movement of *ring* and *little* finger is predicted (Figure 4.2/**B**). Over all data sets available to us, we obtained an average $R^2 = 0.19(\pm 0.031)$ for (1) and $R^2 = 0.25(\pm 0.013)$ for (2). These results are shown as white bars in figure 4.2. We observed that the Pearson correlation coefficient ρ between the ground truth and the reconstructed dimensions on the same data set was significantly higher than the R^2 values (Figure 4.2/**A**: $\rho = 0.47(\pm 0.013)$ and 4.2/**B**: $\rho = 0.49(\pm 0.014)$). The significant different between the R^2 and ρ values suggests that, while our method correctly identifies the shape of the unobserved time-series, scaling is not perfect. This can be explained by considering that we perform our analysis on the velocity values of the data sets. Movements are generally performed at varying speeds, however the regression only takes into account the average movement velocity of the respective cluster.

Interestingly, the accuracy of our method does not improve significantly when we additionally observe the middle finger in scenario (2) and only predict the remaining two fingers instead of observing only thumb and index finger. This could be explained by assuming that our model is either very good at predicting the behaviour of the middle finger from the thumb and index finger or that when observed, the middle finger adds a substantial amount of noise to the overall observation and distorts our parameter estimates.

In a second test, we wanted to find out how our method performs in comparison to the simpler approach of estimating the movement of the unobserved joints by calculating the regression parameters over *unsegmented* data from 5 of our 7 subjects for which we have data available and estimating the missing dimensions on the remaining two subjects. Figure 4.2 (gray columns) shows that although our novel method only takes into account a fraction of the data available (25.55% of all data points), it performs significantly better than the second, simpler approach which takes into entire data. We performed Student t-tests (see figure 4.2) on the results we obtained for both R^2 and ρ and were able to conclude that our novel method is statistically significantly better with $p < 0.05$. The improvement became particularly evident when we tried to predict the movement of two fingers (see comparison in table 4.1).

Here, we reconstruct the unobserved movement behaviour using a linear regression. Various alternatives including *Gibbs sampling* (Markov Chain Monte Carlo) and *maximum likelihood estimation* (MLE) exist, however both concepts did not seem to be suitable for real-time prediction as their computational run-time complexity is relatively high. The former, while being an improvement over direct sampling, still requires repeated sampling from a multivariate distribution. Similarly, we assumed that obtaining a maximum likelihood estimate in high dimensional space would be computationally too expensive where instant prediction is necessary.

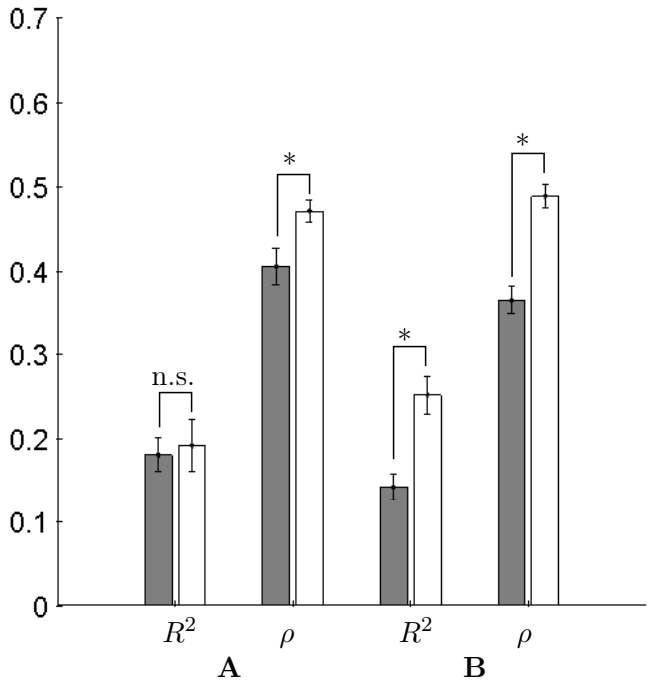


Figure 4.2: Performance of our method (white) against regression over all the observed data (gray). (**A**) Estimate of the trajectory of fingers 3–5 from the observation of fingers 1 and 2 (**B**) Estimate of the trajectory of fingers 4 and 5 from the observation of fingers 1–3. Bars indicate standard error. *:$p < 0.05$. See text for details. Figure from Thomik, Haber, and Faisal (2013).

4.4 Conclusion

We presented a method which is capable of accurately predicting the movement of missing finger joints from the remaining, in case of injury still intact, fingers. Instead of relying on direct information such as EMG of the muscles, we decompose a large set of movement training data into smaller segments (movement primitives) and exploit the typical correlation patterns which are characteristic for the individual segments. We obtain a probabilistic representation of large data sets during a pre-processing step which subsequently allows us to perform movement prediction in *real-time*. Our method was successfully tested by reconstructing the movement of missing fingers on a data set previously used in section 3.2.3 and we were able to show that it is significantly more accurate than the unsegmented approach while at the same time using substantially less training data.

Today, existent mechanisms fail to provide the necessary information throughput when extracting brain signals to allow patients to accurately and naturalistically control a prosthetic replacement that is as complex and versatile as the human hand. We propose to integrate our method into the conceptual design of an improved neuroprosthetic device.

	Clusters	**Entire Data**
Pearson Correlation	0.49 (± 0.014)	0.36 (± 0.091)
Goodness of Fit	0.25 (± 0.013)	0.14 (± 0.085)

Table 4.1: Comparison of prediction accuracy for prediction of fingers 4 and 5 when (1) movement data is segmented and parameters are fitted over clusters and (2) parameters are fitted over entire data set without segmentation and clustering. Values indicate mean (± SD).

Such a device could - on top of decoding intentions from a variety of brain-machine interfaces (BMIs) such as EMG, intra-cortical electrodes or eye-tracking (see section 2.3.2) - make use of prior information about human movement statistics to improve control of a prosthetic device. This is particularly interesting if we further consider for example that it may be possible to predict the movement of the fingers by only observing the movement of shoulder and elbow or predict the behaviour of one hand by observing the other hand.

4.5 Future Work

Despite the promising results that were captured during this project, we were hardly able to scratch the surface and there are many possibilities for further improvements and validation.

Here, we used a linear form of regression to predict unobserved movement behaviour. As an alternative it would be interesting to try a non-linear approximation such as using Gaussian processes for the prediction step.

We further suggest to extend the method presented here to other body parts. To achieve this, relevant data sets would have to be collected in a similar way to the data set obtained by Belić (2010). If we find other parts of the body to interact in a correlated manner during everyday situations, we could then imagine a scenario where, for example, the behaviour of one hand is observed while the movement of the other hand is predicted. Similarly, we could monitor shoulder and elbow muscles and predict the behaviour of the hand. Yet, such data sets do not seem to exist at the moment and it is an open question whether these patterns of correlation can be observed.

5 Conclusion

The initial aim of this project was to develop a computational method that is capable of processing and segmenting complex and long movement time-series. We wanted to use such a method to (1) investigate whether we can find recurring patterns (primitive actions) in everyday movement behaviour collected from humans and (2) utilize our findings and knowledge about human behaviour to design an improved computational method for controlling a more naturalistic and versatile neuroprosthetic device.

During the development of our methodologies we were faced with several computational issues. Initially, we found that none of the existing techniques for movement segmentation and recognition were able to deal with long and high-dimensional time-series data. It appeared that many of the methods use a single and complex probabilistic model for computing segmentation points which complicates inference significantly. In this work, we analysed one existing method (section 3.1) in greater detail and showed that even after several optimisations the computations exploded both in run-time and memory when applied to slightly more complex movement data. After thorough analyses we therefore shifted our focus to a slightly different approach: instead of modelling all our data with a single, complex probabilistic model, we wanted to test whether a smart combination of simple and computationally efficient techniques can be used to perform movement segmentation and recognition on a larger scale and more accurately than previously done.

Resulting from our initial analyses, we developed the method presented in section 3.2; a simple and unsupervised segmentation algorithm that relies on dimensionality reduction techniques, changepoint detection and temporal correlation. The linear run-time properties of the method enabled us to process high-dimensional neurological data sets. We tested our method (section 3.2.3) on artificial data and a large data set containing over one million samples from everyday movement activities and demonstrated that it outperforms the segmentation accuracy and computational efficiency of current state-of-the-art techniques and is suitable for Big Data driven approaches to neuroprosthetics. We further investigated the properties and accuracy of our method by analysing a data set containing recordings from the reproduction of Lower Paleolithic stone tools (section 3.2.3) and demonstrated that we can accurately process complex time-series which are recorded from naturally executed movement activities over a longer period of time.

These properties enabled us to successfully carry out the second part of this project, during which we designed a computational concept for a novel neuroprosthetic controller. Our research showed that until today even the most advanced prosthetics have not been able to demonstrate the same dexterity as the human limb and are often limited to simple grasps. We presented that one of the barriers for fitting amputees with an artificial limb that exhibits truly naturalistic behaviour is often found in the transportation of muscle

contraction signals from the brain to the prosthetic joint. Current techniques including EMG, intra-cortical electrodes or even eye-tracking technologies have not yet managed to overcome or bypass this hurdle.

Similarly to the first part, we attempted a different approach to conceptualising a novel neuroprosthetic device which we described in section 4. Here, we exploited two characteristics of human movement behaviour: (1) modularity, which means that complex movement can be decomposed into simpler segments (i.e. the segments we generated in the first part of this project) and (2) correlation structures between finger joints in everyday movements which were discovered in previous work. We implemented an offline pre-processing step, during which we fitted a probabilistic distribution over clusters containing highly similar movement segments and estimated regression parameters for reconstructing unobserved joint behaviour. We successfully demonstrated that such a probabilistic and inference-based model allows us, by observing a subset of present and still functional joints, to predict the movement behaviour of missing limbs accurately in real-time. We validated our method using the data set from section 3.2.3 and showed that by considering movement segments instead of the entire, unsegmented time-series, our approach yields a significantly higher prediction accuracy while at the same time requiring substantially less training data.

In this report we presented promising results that we hope open new opportunities for further research in the areas of understanding the brain and using our knowledge to build more intelligent machines. We believe that, even though after more than 100 years of research it may seem that we are still at the very beginning of this journey, the combination of computing power and Big Data approaches such as presented in this report will allow us to make much more rapid progress in the near future. We hope to continue contributing to this fascinating research area in the future and perhaps one day can give a truly definite answer to Daniel Wolpert's initial question: "Why do we have a brain?"

References

Abbott, WW and AA Faisal (2012). "Ultra-low-cost 3D gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces." In: *Journal of Neural Engineering* 9.4, p. 046016.

Adams, RP and DJC MacKay (2007). "Bayesian Online Changepoint Detection." In: p. 7. arXiv: 0710.3742.

Belic, J and AA Faisal (2011). "The structured variability of finger motor coordination in daily tasks." In: *BMC Neuroscience*, p. 102.

Belić, Jovana (2010). *Classification and Reconstruction of Manipulative Hand Movements Using the CyberGlove*. Tech. rep. London: Imperial College London, p. 38.

Bishop, CM (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer. ISBN: 0387310738.

Bitzer, S and P van der Smagt (2006). "Learning EMG control of a robotic hand: Towards active prostheses." In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, pp. 2819–2823.

Bizzi, E et al. (1995). "Modular organization of motor behavior in the frog's spinal cord." In: *Trends in Neurosciences* 18.10, pp. 442–446. ISSN: 01662236.

Bizzi, E et al. (2002). "Book Review: Modular Organization of Spinal Motor Systems." In: *The Neuroscientist* 8.5, pp. 437–442.

Chiappa, S and J Peters (2010). "Movement extraction by detecting dynamics switches and repetitions." In: *Advances in neural information processing systems* 23, pp. 388–396.

Cipriani, C and F Zaccone (2008). "On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction." In: *IEEE Transactions on Robotics* 24 (1).

Coates, A, P Abbeel, and AY Ng (2008). "Learning for control from multiple demonstrations." In: pp. 144–151.

d'Avella, A, P Saltiel, and E Bizzi (2003). "Combinations of muscle synergies in the construction of a natural motor behavior." In: *Nature neuroscience* 6.3, pp. 300–308.

D'Avella, A et al. (July 2006). "Control of fast-reaching movements by muscle synergy combinations." In: *The Journal of Neuroscience* 26.30, pp. 7791–810. ISSN: 1529-2401.

Davidson, J (2002). "A survey of the satisfaction of upper limb amputees with their prostheses, their lifestyles, and their abilities." In: *Journal of Hand Therapy* 15.1, pp. 62–70.

Delis, I et al. (2013). "A methodology for assessing the effect of correlations among muscle synergy activations on task-discriminating information." In: *Frontiers in Computational Neuroscience* 7, p. 54. ISSN: 1662-5188.

Faisal, AA et al. (2010). "The manipulative complexity of Lower Paleolithic stone tool-

making." In: *PloS one* 5.11, e13718.

Fink, E and HS Gandhi (2011). "Compression of time series by extracting major extrema." In: *Journal of Experimental & Theoretical Artificial Intelligence* 23.2, pp. 255–270.

Gentner, Reinhard and Joseph Classen (2006). "Modular organization of finger movements by the human central nervous system." In: *Neuron* 52.4, pp. 731–42. ISSN: 0896-6273.

Haber, D, AAC Thomik, and AA Faisal (2013). "Simple unsupervised time series segmentation for processing high-dimensional neurological data streams in linear time." In: *NIPS - in review*.

Hochberg, LR, D Bacher, and B Jarosiewicz (2012). "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm." In: *Nature* 485.7398, pp. 372–375.

Hochberg, LR et al. (2006). "Neuronal ensemble control of prosthetic devices by a human with tetraplegia." In: *Nature* 442.7099, pp. 164–171.

Kargo, WJ and SF Giszter (2000). "Rapid correction of aimed movements by summation of force-field primitives." In: *The Journal of Neuroscience*.

Kuiken, TA et al. (2007). "Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study." In: *The Lancet* 369.9559, pp. 371–380.

Kuiken, TA et al. (2009). "Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms." In: *JAMA: the journal of the American Medical Association* 301.6, pp. 619–628.

Kutch, JJ and FJ Valero-Cuevas (2011). "Muscle redundancy does not imply robustness to muscle dysfunction." In: *Journal of biomechanics* 44.7, pp. 1264–1270.

Leonardis, A, H Bischof, and A Pinz (2006). *Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost*. Computer V. vol. 3954. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 359–372. ISBN: 978-3-540-33838-3.

Mason, CR, JE Gomez, and TJ Ebner (2001). "Hand synergies during reach-to-grasp." In: *Journal of Neurophysiology* 86.6, pp. 2896–2910.

Meier, F, E Theodorou, and S Schaal (2012). "Movement segmentation and recognition for imitation learning." In: *International Conference on Artificial Intelligence and Statistics, AISTATS*. vol. 2012.

Meier, F et al. (2011). "Movement segmentation using a primitive library." English. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3407–3412. ISBN: 978-1-61284-456-5.

Müller-Putz, GR and R Scherer (2005). "Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components." In: *Journal of neural ...* 2.4, p. 123.

Rittscher, J and A Blake (1999). "Classification of human body motion." In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision*.

Rodriguez-Serrano, J and F Perronnin (2012). "A model-based sequence similarity with application to handwritten word-spotting." In:

Roeschlein, RA and E Domholdt (1989). "Factors related to successful upper extremity prosthetic use." In: *Prosthetics and orthotics international* 13.1, pp. 14–18.

Santello, M, M Flanders, and JF Soechting (1998). "Postural hand synergies for tool use." In: *The Journal of Neuroscience* 18.23, pp. 10105–10115.

Sherrington, SCS (1908). "The integrative action of the nervous system." In: *Constable, London*.

Taylor, DM, SIH Tillery, and AB Schwartz (2002). "Direct cortical control of 3D neuroprosthetic devices." In: *Science* 296.5574, pp. 1829–1832.

Thomik, AAC and AA Faisal (2012). "Deriving Motion Primitives From Naturalistic Hand Movements for Neuroprosthetic Control." In: *BCCN 2012 Munich*, p. 201.

Thomik, AAC, D Haber, and AA Faisal (2013). "Real-Time Movement Prediction For Improved Control Of Neuroprosthetic Devices." In: *35th Annual International IEEE EMBS Conference - in review*.

Velliste, M, S Perel, and MC Spalding (2008). "Cortical control of a prosthetic arm for self-feeding." In: *Nature* 453.7198, pp. 1098–1101.

Williams, B, M Toussaint, and A Storkey (2006). "Extracting Motion Primitives from Natural Handwriting Data." In: *Artificial Neural Networks–ICANN 2006*, pp. 634–643.

Wolpaw, JR and DJ McFarland (1994). "Multichannel EEG-based brain-computer communication." In: *Electroencephalography and clinical Neurophysiology* 90.6, pp. 444–449.

Wolpaw, JR and DJ McFarland (2004). "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans." In: *Proceedings of the National Academy of Sciences of the United States of America* 101.51, pp. 17849–17854.

6 Appendix

As part of this project, we part of this work for publication and have attached both writings to this report. The first publication covers our work on segmentation of a large neurological data set which we presented in chapter 3 and has been submitted to the Neural Information Processing Systems (NIPS) conference (5-8 December 2013, Lake Tahoe, United States). The second publication presents our concept for an improved neuroprosthetic device which we devised in chapter 4 and has been submitted for consideration at the 6th International IEEE EMBS Neural Engineering Conference (6-8 November 2013, San Diego, California, United States). As of submission date of this report, both publications are currently in review.

Simple unsupervised time series segmentation for processing high-dimensional neurological data streams in linear time

David Haber
Department of Computing
Imperial College London
London SW7 2AZ, UK
dh909@imperial.ac.uk

Andreas Thomik
Department of Bioengineering
Imperial College London
London SW7 2AZ, UK
aat10@imperial.ac.uk

Aldo Faisal
Department of Bioengineering
Imperial College London
London SW7 2AZ, UK
a.faisal@imperial.ac.uk

Abstract

Many important challenges in Big Data require the analysis and segmentation of very long, high dimensional time series - which become often computationally intractable for state-of-the-art Bayesian approaches based on variational, non-parametric or MCMC derived methods. Here we present how the simple combination of PCA, approximate Bayesian segmentation and temporal correlation processing can achieve reliable time series segmentation outperforming high-concept state-of-the-art algorithms when applied to real world data streams. We applied our method, which relies on simple iterative covariance, correlation and maximum likelihood operations to perform complex neurological time series segmentation over millions of samples in 20 dimensions in linear time and space. Our method is suitable for even higher dimensional data streams as performance scales near constantly with the dimensionality of the time series samples. Our method enables Big Data driven approaches to neuroprosthetics and brain-machine-interfaces to become computationally tractable.

1 Introduction

Almost every natural or artificial phenomenon can be described as times series, such as the two dimensional coordinates a pen follows during writing. Crucially, many of these sequences can be decomposed into a finite set of sub-sequences which are concatenated in an order to generate the observed data, e.g. letters in writing. This is of particular interest if their arrangement is not purely random but follows a particular rule (such as spelling and grammar in writing) as one might be able to infer the rule from the alignment of sub-sequences, allowing for greater understanding of the generating process. A major interest of decomposing data into repeating sub-sequences comes from the field of motor control where it has been suggested that the brain may generate and control complex movement by combining a set of movement primitives [1, 2]. However, it is still problematic to (1) identify the correct number and shape of the sub-sequences and (2) locate them in a noisy data stream. These challenges are particularly important in the field of neurotechnology where the time-series studied are often of high-dimension (21 degrees of freedom (DoF) for the hand alone, 600 and more degrees of freedom for the muscular-skeletal system) and very long (often 1 million

and more samples). It is therefore crucial to have algorithms capable of handling both these cases efficiently and accurately with limited resources.

Current methods can be classified as either dictionary-based or dictionary-free. Library-based approaches typically build on a dictionary of known sequences and try to identify time-warped and/or noisy version thereof in a stream of data by first estimating potential segmentation points and then matching the corresponding data segments with those in the library either offline [3] or online [4]. Conversely, dictionary-free approaches try to infer both segment shape and starting point directly from the data using either simple Kalman filters [5], factorial Hidden Markov Models (HMM) [6] or inference on a probabilistic representation of both segmentation and segment shape [7]. The inference which needs to be performed in these methods makes them computationally expensive and unlikely to scale well with increasing number of data points and dimensionality of the data.

In this paper, we show how it is possible to outperform state-of-the-art techniques, in particular for high-dimensional and long data sequences, by using conceptually very simple and computationally inexpensive methods. We exploit the covariance structure of the data time series, Bayesian changepoint detection and temporal correlation. We evaluate the accuracy of our method by varying the dimensionality, noisiness and number of sub-sequences involved of an artificial data set. Subsequently, we apply the method to hand-movement data of people performing a series of everyday tasks and show how a wide range of movements can be described by a smaller subset of building blocks.

2 Proposed Method

Instead of modelling all aspects of the data set with a single model - which would make inference on it inherently complex - we combine four different, simple and specialised methods for pre-processing and segmentation of the data and extraction of unique segments. We use (1) Principal Component Analysis (PCA) for data compression, (2) changepoint analysis [8] to determine potential segmentation points in the data, (3) time-series compression [9] to determine the most relevant change points and (4) comparison of extracted segments to determine the subset of generating segments. We will give a brief overview of the methods as well as considerations on performance and eventual changes we made to their classic form.

2.1 Principal Component Analysis

Using PCA, we find a lower-dimensional representation of the data. This has the advantage of removing redundant information which is beneficial both in terms of memory and computational costs. Typically, this is chosen to be the number of principal components which account for a set amount of the variance observed, although this approach might behave poorly for very noisy data [10]. We find that for our database, using enough principal components to account for 90% of the variance observed produced satisfying results (but see Results for a more detailed analysis).

2.2 Change Point Analysis

We developed a Bayesian Change Point detection algorithm, suitable for very long, high-dimensional time series which is highly scalable for Big Data problems, as it operates with linear complexity in time and in memory (for the number of data points). Our method is derived from an elegant algorithm for one-dimensional time series by Adams & MacKay [8] which scales quadratically in computing time and memory. Briefly, the method assumes that each data point x_t $(t = 1, \ldots, T)$ in each segment S_i are i.i.d. variables sampled from a probability distribution $P(x_t|\Theta_i)$ with parameter set Θ_i. At every time step, we say that the most likely run-length *r* of the current segment is given by

$$r = argmax_k(P(x_t|\theta_k) \mid k = 1 \ldots t) \tag{1}$$

where Θ_k is the maximum likelihood estimate of the parameters given the last *k* data points. We assume that the joint distribution over run-length and observed data is given by [8]

$$P(r_t, x_{1:t}) = \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t}) \tag{2}$$

$$= \sum_{r_{t-1}} P(r_t, x_t | r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \tag{3}$$

$$= \sum_{r_{t-1}} P(r_t | r_{t-1}) P(x_t | r_{t-1}, x_t^{(r)}) P(r_{t-1}, x_{1:t-1}) \tag{4}$$

where r_t describes the current run-length and $x_{1:t}$ the data observed up to the time point t arises from a multivariate Gaussian distribution. We initialised the multivariate Gaussian parameters as follows:

$$\mu_0 = \mu_{prior} \tag{5}$$

$$\Sigma_0^{-1} = \Sigma_{prior}^{-1} \tag{6}$$

$$\kappa_0 = 1 \tag{7}$$

where D is the dimensionality of the input data. At each iteration the parameters are updated given the following equations:

$$\mu_{t+1}^{(0)} = \mu_{prior} \tag{8}$$

$$\mu_{t+1}^{(1:t+1)} = \frac{\kappa_t \times \mu_t^{(0:t)} + x_t}{\kappa_t + 1} \tag{9}$$

$$\Sigma_{t+1}^{-1(0)} = \Sigma_{prior}^{-1} \tag{10}$$

$$\Sigma_{t+1}^{-1(1:t+1)} = \frac{\kappa_t}{\kappa_t + 1}(x_t - \mu_0)^T(x_t - \mu_0) + \Sigma_t^{-1(0:t)} \tag{11}$$

$$\kappa_{t+1} = \kappa_t + 1 \tag{12}$$

where $\mu_{t+1}^{(1:t+1)}$ denotes elements 1 to $t+1$ of the mean vector at time $t+1$. We found that the elegant algorithm by Adams & MacKay [8] space becomes intractable for time series in excess of about 10,000 sample points as it keeps track of all possible run-lengths for each possible time-step, making memory cost quadratic in the number of time-steps. We found that this can be easily overcome by only considering the most likely run-length and dropping all other information, thus making the memory cost linear with the number of time steps. Another performance improvement is achieved, as we found that at time-step t_n, parameters do not have to be computed for all possible run-length $r = 1 \dots t$. Instead, given the final correlation lengths of movements, we can compute parameters r over a limited time window $t_w \ll t$ to obtain the same results (although it is important to choose t_w carefully to avoid over-fitting the parameters - in the case of a normal distribution this means ensuring $t_w > 3 \times D$). This can be explained by considering that the number of data points to avoid over-fitting the sufficient statistics of the underlying distribution is constant and only dependent on the number of free parameters it has. Adding additional data, whilst reducing the uncertainty of our estimates as a function of the square root of the number of added observations, will only marginally change the estimate of the maximum likelihood.This has the advantage of making the computational cost constant in time and altering the outcome of the computation only marginally.

2.3 Extraction of Change Points

We assume that a change point occurs when the time series switches between generative processes, i.e. between two different segments. The ideal behaviour of the changepoint algorithm, a linearly increasing run-length followed by a drop to zero, is, however, rarely observed on our behavioural data because noise and non-stationarity in the data induce uncertainties in the run-length estimate. A typical result produced by the algorithm as shown in Fig. 1 highlights the need for additional processing to extract the most relevant change points. Therefore we define the change points as where the maximum likelihood run-length drops significantly, where the height of the drop is computed according to the distance function [9]:

$$dist(r_{t-1}, r_t) = \frac{|r_{t-1} - r_t|}{|r_{t-1}| + |r_t|} \tag{13}$$

which is bounded between 0 and 1. We varied the threshold above which we considered a drop in run-length as significant and found it to be very flexible as long as it was not too close to the extremes (i.e. 0 or 1). Typically, values between 0.5 and 0.85 would produce almost identical results. Furthermore, we introduced a prior distribution over the duration of our movement segments, thus limiting the frequency of changepoint appearance. This distribution is may be estimated directly from the data generating process or from the time-series itself. Thus, a significant changepoint is defined both in terms of maximum run-length value and time since the last changepoint was detected.

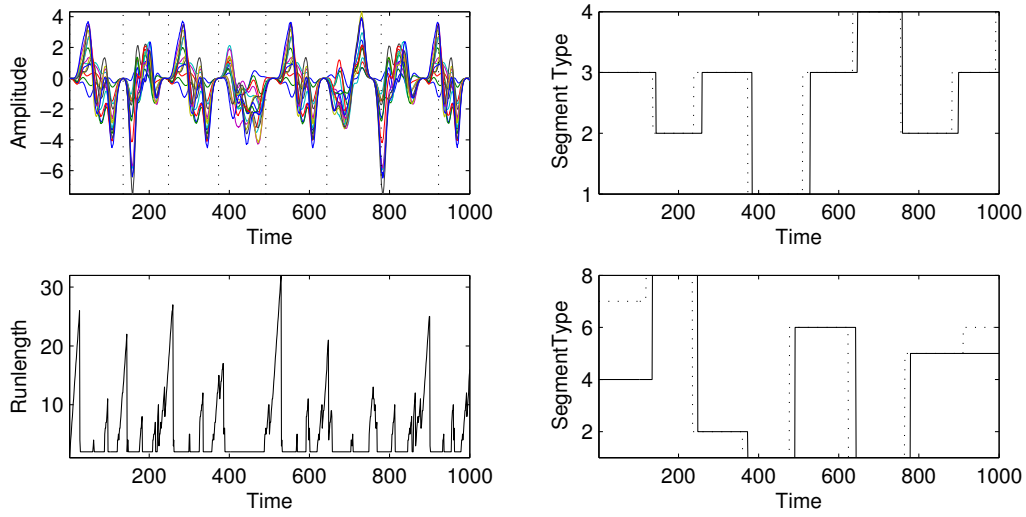


Figure 1: **Top left**: 15-dimensional data set, vertical lines indicate changepoints. **Bottom left**: maximum likelihood runlength as computed by changepoint algorithm. Note the irregularity due to noise in the data and non-stationarity. **Right panel**: Segmentation of the data, solid line: real segmentation, dotted line: computed segmentation.

2.4 Resampling and Comparison of Segments

We are interested in recognising similar segments that have been extracted in sequences of natural movement. It has to be taken into account that movements can be performed at different speeds, yet representing the same type of action. We therefore resample every segment to a default movement duration and use statistical measures to calculate the similarity between two segments with respect to their type. The statistical measures which we used include Euclidean distance, Pearson correlation coefficient and the comparison of principal components. To assess the accuracy of these measures we visualised the real hand data on a computer screen and compared our results based on the statistical measures with an expert's visual assessment of similarity between two segments. We found that Pearson correlation coefficient yields the best results on our data sets.

Table 1: Performance measures for proposed method on artificial data set. K is the number of different *types* of segments in the input data. Values indicated are mean (± SD).

K	K Extracted	Correlation Coefficient
3	4.60 (± 2.19)	0.81 (± 0.24)
5	6.40 (± 1.48)	0.86 (± 0.18)
7	8.53 (± 1.94)	0.87 (± 0.18)
9	10.60 (± 2.19)	0.85 (± 0.18)
10	11.71 (± 1.84)	0.87 (± 0.17)
11	12.77 (± 1.72)	0.87 (± 0.17)
13	13.83 (± 1.21)	0.88 (± 0.15)
15	14.77 (± 1.79)	0.89 (± 0.15)

2.5 Resampling and Comparison of Segments

We are interested in recognising similar segments that have been extracted in sequences of natural movement. It has to be taken into account that movements can be performed at different speeds, yet representing the same type of action. We therefore resample every segment to a default movement duration and use statistical measures to calculate the similarity between two segments with respect to their type. The statistical measures which we used include Euclidean distance, Pearson correlation coefficient and the comparison of principal components. To assess the accuracy of these measures we visualised the real hand data on a computer screen and compared our results based on the statistical measures with an expert's visual assessment of similarity between two segments. We found that Pearson correlation coefficient yields the best results on our data sets.

3 Results

We will now present the results that we obtained from two different types of data: an artificial data set and hand-movement data of various subjects performing everyday tasks.

3.1 Artificial Data Set

We generated an artificial data set by randomly generating a set of sequences from the family of functions defined as

$$f = \left(\sum_{i=1}^{N} w_i \phi(t - t_i) \right) + \epsilon \tag{14}$$

where $w_i \sim \mathcal{N}(0, I_D)$, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, Φ were chosen to be Gaussian basis functions, N is the number of basis functions, D the dimensionality of the data and t_i can be arbitrarily chosen to constrain the function to a given interval. The segments generated in this way were subsequently resampled to have a length which was sampled from $\mathcal{U}(100, 150)$ and aligned at random to generate a data set. We validated our method by generating data with varying its dimensionality, number of segments, and noise levels as well as changing all parameters of the method. During each test, all but the parameter varied where kept constant to D = 15, K = 10, N = 5000, noise = 0%, and PCA accuracy = 90%. We measured performance of our method by calculating the similarity between the original and extracted segment sequence (see figure 1, right panel). This is a very strict test as even slight errors in segment start point will negatively impact the computed accuracy, even if the segment identity was correctly identified.

Results are shown in Fig. 2 where the markers indicate mean values, whiskers standard deviation (SD) and grey areas values below chance level (where applicable). It should be noted that our method is extremely robust to most variations in the data and algorithm parameters, exhibiting very high accuracy (86.5%) and no significant decline in performance with increasing segment number

(Fig. 2/**B**) or noise (Fig. 2/**C**). The only significant drop in performance can be observed for low-dimensional data (Fig.2/**A**). We found this to be caused by the PCA step of the algorithm which would frequently collapse low-dimensional input data onto a single dimension which in turn makes it difficult for the changepoint algorithm to find differences in the data as its statistics become very similar across segments. A potential workaround would be to put a lower bound on the number of output dimensions given by the PCA step.

We further analysed how many different segments our algorithm extracts from a data set by comparing all segments extracted and clustering them depending on their similarity as computed from Pearson correlation coefficient (see also section 2.5). We defined two segments to be the same if their correlation coefficent was larger than 0.7 which we had found to be an optimal value based on cross validation. The results are shown in Table 1 where we show the number of unique segments estimated by our method compared to the real number of different segments in the data, as well as the average correlation coefficient. Clearly, the average number of estimated segments in the data is slightly higher than the real number of segments in the data. This is not particularly surprising as any "wrong" segment will form a cluster by itself and therefore push the estimate up. Considering this, the algorithm performs surprisingly well as, on average, it only finds 1.2 segments more than were present in the real data set. This has to be contrasted with the total number of segments in each data set which was, on average, forty segments.

3.1.1 Computational Performance

We tested how well our algorithm scales with increasing number of samples and dimensions of the input data (see Fig. 2/**E** and **F**). The figures show run times normalised to the run time of the smallest data set (i.e. 5000 data points or 5 dimensions). As predicted in section 2.2, our method scaled linearly in time with the number of samples in the data. The dependence on data dimensionality was found to be almost constant, mainly thanks to the PCA step which (although it was not specified) ensured that the data dimensionality for the changepoint algorithm was kept relatively low.

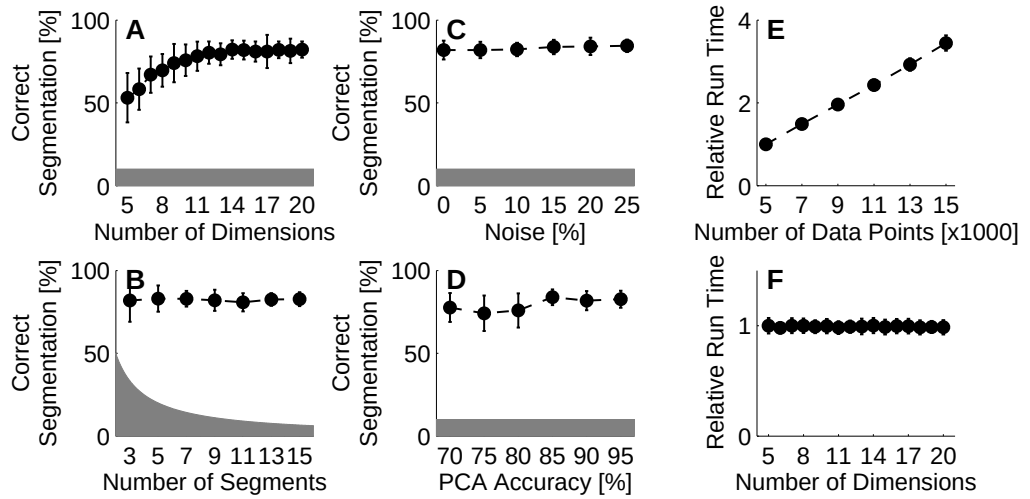


Figure 2: **(A)** - **(D)**: Influence of parameters on extraction accuracy. **(A)** dimensionality, **(B)** number of different segments, **(C)** noise levels, **(D)** PCA accuracy. **(E)** - **(F)**: Influence of number of data points **(E)** and number of dimensions **(F)** on run time. Markers indicate mean with standard deviation and the grey area represents levels of chance. If no errorbars are shown, they are smaller than the marker. Note that the run-lengths have been normalised to the shortest run-length. See text for details.

3.2 Neurotechnology for Kinematics in Daily Tasks

We next ran our algorithm on hand movement data collected during everyday tasks such as opening doors, using keys, eating, drinking, etc. (17 tasks in total, [citation withheld]). The data consists of

an 18-dimenional data stream corresponding to all joints of the wrist and hand with exception of the distal interphalangeal (DIP) joints and was collected using a CyberGlove I (CyberGlove Systems LLC, CA, USA). The data was recorded from seven subjects and its total length was just over six hours, or approximately 1.7M data points. Our interest was twofold: (1) are there any movement segments which get repeated across different tasks and (2) are these segments unique to the subject or a general pattern of human movement? We therefore separated the data by task and ran the data of each task individually through our algorithm. The distribution of movement length was estimated from a set of annotated data which had been collected previously (data not shown).

The segmentation of the data resulted in 6560 segments with a mean run-length of 256 data points, equivalent to a movement duration of 3.2 seconds. Similarly to the calculations performed on the artificial data set, we computed the similarity between extracted segments by warping them to the same length and calculating the Pearson correlation coefficient between them. We found that a large proportion the segments clustered around two different actions which by visual inspection we could classify as opening and closing of the hand. Whilst this is obviously a highly recurrent element of everyday tasks, we noted that by increasing the threshold above which two segments were considered the same it is possible to further differentiate between different openings and closings. However, the large amount of segments clustered as opening or closing suggest that maybe the Pearson correlation coefficient may not be the ideal metric for comparing real movement segments as much of the scale information is lost. Nevertheless, this is emphasises the importance of the grasp action in everyday life and singles this movement out as the most important to rehabilitate after injury or stroke.

4 Conclusion

The ease with which we are today capable of collecting data from a large range of sources in real time offers the prospect of exciting findings in a wide field of research topics. To make the most of this data, it is, however, important to have the right tools to analyse it. Typically, this means using algorithms which are as economical as possible both in computational and memory costs. Here, we showed how by smartly combining a few specialised and computationally efficient algorithms it is possible to accurately segment large data sets with a computational cost which scales linearly with the number of data points considered and is constant with the dimensionality of the input data.

Acknowledgements

References

[1] Emilio Bizzi, Simon F Giszter, Eric Loeb, Fernando A Mussa-Ivaldi, and Philippe Saltiel. Modular organization of motor behavior in the frog's spinal cord. *Trends in neurosciences*, 18(10):442–446, 1995.

[2] Kurt A Thoroughman and Reza Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407(6805):742–747, 2000.

[3] Franziska Meier, Evangelos Theodorou, and Stefan Schaal. Movement segmentation and recognition for imitation learning. In *AISTATS*, volume 2012, 2012.

[4] Franziska Meier, Evangelos Theodorou, Freek Stulp, and Stefan Schaal. Movement segmentation using a primitive library. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3407–3412. IEEE, 2011.

[5] Adam Coates, Pieter Abbeel, and Andrew Y Ng. Learning for control from multiple demonstrations. In *Proceedings of the 25th international conference on Machine learning*, pages 144–151. ACM, 2008.

[6] Ben H Williams, Marc Toussaint, and Amos J Storkey. *Extracting motion primitives from natural handwriting data*. Springer, 2006.

[7] Silvia Chiappa and Jan Peters. Movement extraction by detecting dynamics switches and repetitions. *Advances in Neural Information Processing Systems*, 23:388–396, 2010.

[8] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[9] Eugene Fink and Harith Suman Gandhi. Compression of time series by extracting major extrema. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(2):255–270, 2011.

[10] Ioannis Delis, Bastien Berret, Thierry Pozzo, and Stefano Panzeri. A methodology for assessing the effect of correlations among muscle synergy activations on task-discriminating information. *Frontiers in Computational Neuroscience*, 7:54.

REAL-TIME MOVEMENT PREDICTION FOR IMPROVED CONTROL OF NEUROPROSTHETIC DEVICES

Andreas A. C. Thomik[1], David Haber[2] and A. Aldo Faisal[1,2,3], *Member IEEE*

***Abstract*— Replacing lost hands with prosthetic devices that offer the same functionality as natural limbs is an open challenge, as current technology is often limited to the most basic grasps. In this work we develop a probabilistic inference-based method that allows for improved control of neuroprosthetic devices by observing the behaviour of the undamaged limb to predict the most likely actions of lost limbs. Our algorithm first automatically segments the streaming joint movement data of intact limbs into chunks of data that belong together (e.g. the limb motions involving a grasp) based on a database of daily living finger movements. It then estimates which action is currently happening to predict the movements for the missing joints. On test data, we can demonstrate that this two-stage approach yields statistically significantly higher prediction accuracy for missing limb movements than linear regression approaches that reconstruct limb movements from their straightforward correlation structure.**

I. INTRODUCTION

Neuroprosthetics offer the chance for people with a lost limb to regain naturalistic mobility and dexterity by controlling a prosthetic replacement (such as an artificial hand) in the same way they would control their own body. Whilst spectacular advances have been made by using direct cortical control from both invasive [1], [2], [3], [4] and non-invasive electrodes [5], [6], [7] as well as using other information sources such as EMG [8], [9], [10], the amount of information we can extract with current technology is still very low and not suited for real-time control of prosthetic devices with the same versatility as the natural limb [11]. This restricts the potential use of neuroprosthetics in everyday activities considerably and often leads to frustration and rejection of the prosthesis by the user [12], [13], [14].

Here, we suggest to enhance the control of neuroprosthetic devices by exploiting two aspects of natural movement: (1) modularity, by which complex movements are a combination of a finite set of simpler movements [15], [16]; and (2) stereotypical patterns of correlation between joints of the fingers during everyday tasks [17].

We base our method on a large data set of complex hand movements which represents more than 6 hours worth of data from different subjects performing a variety of tasks [18]. On this data we use an algorithm which we previously developed [19] to extract movement primitives and to generate clusters. These allow us to accurately complement partial observations (e.g. movement of three fingers) and predict the behaviour of the unobserved joints (e.g. remaining two fingers) in real-time by taking into account the probabilistic structures of the most representative clusters. We show that by considering movement primitives (or segments) instead of the entire, un-segmented time-series, we can significantly improve the prediction accuracy for the movement of non-observed joints with significantly less training data.

[1]Department of Bioengineering, Imperial College London, SW7 2AZ London, UK andreas.thomik10 at imperial.ac.uk

[2]Brain & Behaviour Lab, Dept. of Computing, Imperial College London, SW7 2AZ London, UK david.haber09 at imperial.ac.uk

[3]MRC Clinical Sciences Centre, Hammersmith Hospital Campus, W12 0NN London, UK a.faisal at imperial.ac.uk

II. METHODS

A. Data

We use hand movement data from subjects performing everyday tasks such as opening doors, using the phone, eating, etc. [18] The data consists of 18-dimensional time-series representing bending of all joints of the fingers with exception of the distal interphalangeal joint (DIP), as well as flexion/extension and adduction/abduction of the wrist as captured by a CyberGlove I (CyberGlove Systems LLC, CA, USA). Data was collected from 7 subjects and sampled at 80 Hz with a resolution of 8 bit per sensor. Before further analysis, the data is smoothed using a second-order Savitzky-Golay filter with a running window of 5 data points to remove discontinuities induced by the A/D converter. Finally, sensor values relating to wrist position are discarded to obtain a 16-dimensional data set of pure finger movements. The subsequent analysis is performed on the velocity values of the data.

B. Segmentation

We segment the data from all subjects and tasks using the method described in [19]. The algorithm first performs PCA to reduce the data dimensionality before segmenting the data using an efficient Bayesian changepoint detection algorithm and a prior over movement duration. Based on the changepoints that were computed we extract movement segments from the original data. The segmentation algorithm scales linearly with the number of data points and is near constant in dimensionality of the data which allows for the processing of very large data sets as collected from motion capture equipment over a prolonged period of time.

C. Clustering

A sequential process is used to measure similarity between two segments. The segments are resampled to a default movement duration to ensure consistency in the number of data points per segment. We then normalise all segments to have amplitudes in the interval $[-1, 1]$ by dividing each segment's amplitude by its absolute maximum. Finally, the

Preprint submitted to 6th International IEEE EMBS Conference on Neural Engineering. Received June 14, 2013.

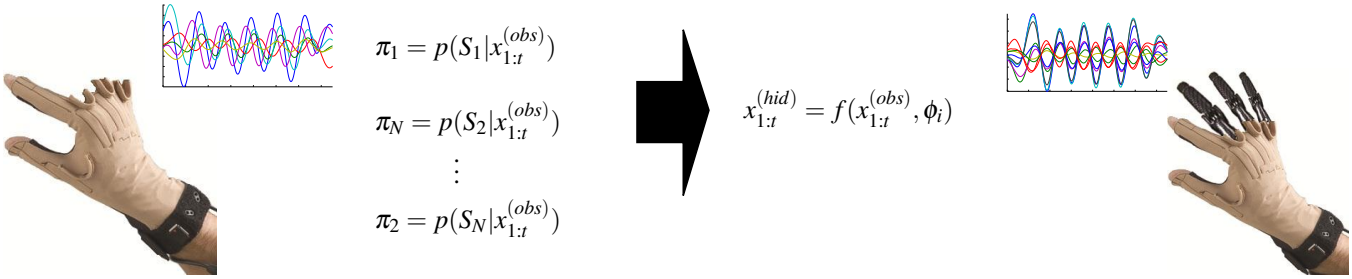


Fig. 1. Conceptual design of the prosthesis controller. **Left hand side:** A limited amount of information is observed (here, two fingers) and inference is performed to evaluate which movement primitive most likely generated the observed data. **Right hand side:** The parameters of the most likely movement primitive are used to estimate the missing dimensions and the prosthetic replacement is actuated accordingly.

similarity between clusters is obtained by computing the maximum cross-correlation (MXC) of all possible pairs of segments over a limited time-lag corresponding to 20% of the segment length. This corrects for slight misalignments of movement within the segments. We found that alternative methods such as the Pearson correlation coefficient were less accurate indicators of similarity as they are highly affected by even marginal shifts in time between segments that represent the same type of movement.

Clustering of segments is performed using an iterative method which, starting with an arbitrary segment, finds all other segments which are more similar than a given threshold t and repeats this with the found segments until the cluster converges to a stable state. Visual inspection of clustered elements showed clear resemblance in the movement executed. A slight drawback of this method is that it generates a number of clusters with a single segment as they are deemed too different from any other segment and therefore seen as truly unique by the algorithm. The number of these unique clusters can be regulated by t. These unique segments can be caused by either (1) unnatural manipulation of objects or (2) erroneous segmentation of the time-series by the algorithm in the pre-processing step.

D. Inference

To infer which cluster a given data point may belong to, we fit all segment clusters with a mixture of Gaussians in order to obtain a probabilistic representation of the data. We choose the optimal number of components for each mixture of Gaussians using the Bayesian Information Criterion. Consequently, the likelihood of the data at time t being generated by a particular cluster S_i is given by

$$\pi_i = p(S_t^i|x_{1:t}^{obs}) = \frac{p(x_{1:t}^{obs}|S_t^i) \times p(S_t^i)}{p(x_{1:t}^{obs})} \quad (1)$$

where $p(S_t^i)$ can be estimated directly from the training data and $x_{1:t}$ is the data observed so far. We use a weighted sum of probabilities for the data points observed in the past such that the influence of past data points decays exponentially over time.

E. Prediction

To predict the unobserved dimensions, we fit a regression with a linear combination of 50 Gaussian basis functions between the observed and unobserved dimensions for each cluster in the data set offline. Upon observing partial movement behaviour, we can compute the unobserved dimensions online and in real-time. We select the most likely cluster from the previously computed π_i distribution in a winner-takes-all fashion and estimate the hidden trajectory on the basis of the regression parameters fitted to that particular cluster. The time required to predict the unobserved movement behaviour at a given time-step is therefore only dependent on the number of clusters considered during the prediction step.

III. RESULTS

The following section presents the results of the segmentation, clustering and prediction steps of our method. We show that by exploiting the unique characteristics of each movement primitive cluster we are capable of predicting the unobserved data much more precisely than by taking into consideration the entirety of the movement data.

1) Movement Primitives: The segmentation of the data into movement primitives by the algorithm produced a total of 3710 primitives with an average movement duration of 3.28 seconds. Note that when a block of data is segmented, the first and last segments are discarded as they are potentially incomplete. Clustering of the movement primitives as described in the Methods section produced 2279 clusters when the similarity threshold was set to t = 0.8. We note that of these clusters, a vast majority (88.85%) were singleton clusters, i.e. segments for which the similarity with any other segment was less than t.

To simplify the procedure of estimating the regression parameters and to limit the influence of the singleton segments, we only consider movement primitive clusters which account for at least 0.2% of the total number of data points collected. This means that out of the 2279 clusters that are initially present, only 34 (1.5% of the clusters, 25.55% of the all data points) were used to predict the unobserved data. In Fig. 2 we show the similarity of the cluster core segments with

Preprint submitted to 6th International IEEE EMBS Conference
on Neural Engineering. Received June 14, 2013.

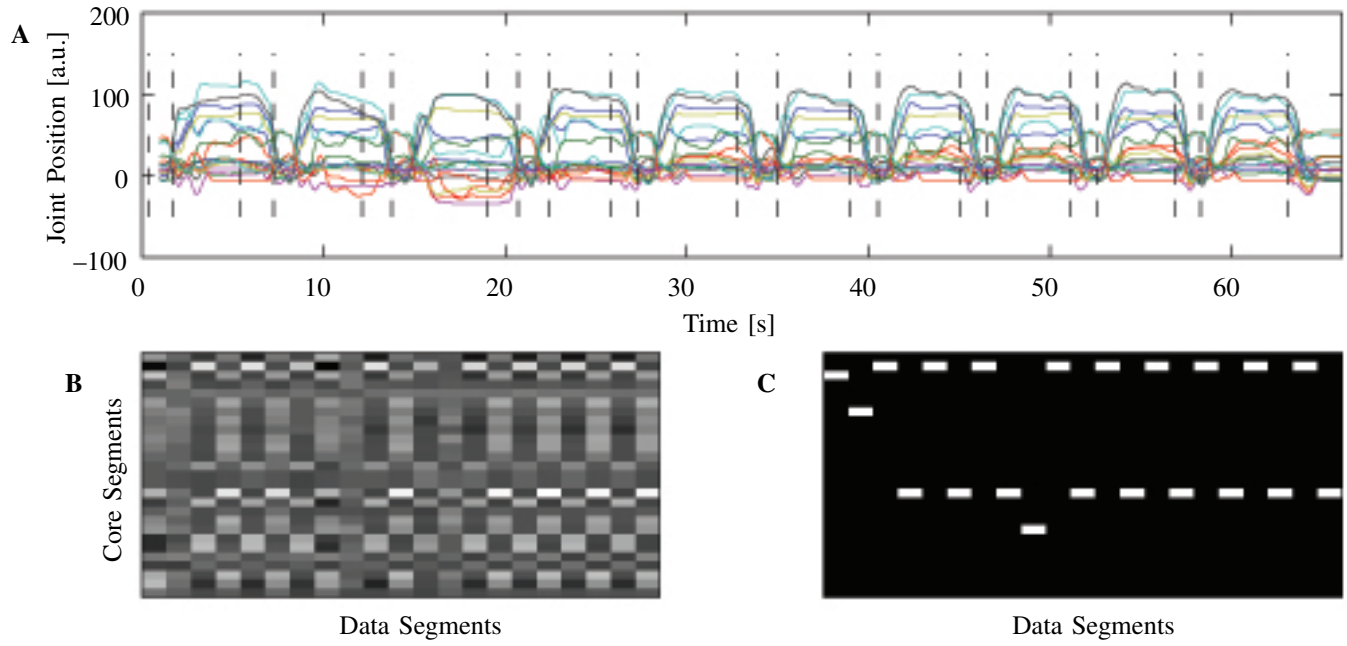


Fig. 2. (**A**) Raw data of a subject performing 10 repetition of the same action. Vertical lines indicate segmentation points as determined by the algorithm. (**B**) Similarity between the segments from (**A**) and the general movement primitives extracted from a larger data set. The brighter the case, the more similar the segments are. (**C**) Most likely movement primitive at each moment in time. See text for details.

the segments obtained from a longer recording in which the subject picked up and put down a bag 10 times. The raw data and segmentation are shown in Fig. 2**A**. Fig. 2**B** shows the similarity of the cluster centroids with the extracted segments as given by MXC. Note that the data shown here was not used to form the clusters against which it is being compared. Fig. 2(C) presents the segments with the highest similarity. The repetitive nature of the task becomes evident from the alternating order of underlying primitives.

2) Prediction: Estimation of the unobserved data was a two step process in which (1) the movement primitive cluster which has most likely generated the observed data is selected and (2) unobserved dimensions are estimated using a regression particularly fitted to the data of that movement primitive cluster (see Methods for details). We note that when the most likely generating movement primitive changed between time-step t and $t+1$, we sometimes observe a considerable change in the estimate of x_t^{hid} and x_{t+1}^{hid}. This can be explained by the statistical nature of primitive inference which sometimes leads to the wrong primitive being used for prediction. We safeguarded ourselves from such crude mistakes by fitting an exponential distribution to the absolute change in value between two time-steps of the unobserved data in our training set. If a jump of magnitude $|\hat{x}_t - \hat{x}_{t+1}|$ or larger had a likelihood less than 0.1%, we assumed that the inference had failed and re-iterated the movement primitive inference the current estimate discarded. This successfully prevented jumps in the estimate of unobserved dimensions whilst additionally refining the inference step for the underlying movement primitive.

Using this method, we tested two different scenarios (see Fig. 3): (1) only the thumb and index (fingers 1 and 2) were observed and the movement of middle, ring and little finger (fingers 3-5) estimated (Fig. 3**A**); and (2) thumb, index and middle finger were observed and the movement of ring and little finger reconstructed (Fig. 3**B**). Over all data sets at our disposal, we thus obtained an average $R^2 = 0.19$ (± 0.031) for the first and $R^2 = 0.25$ (± 0.023) for the latter case (see Fig. 3, white columns). Interestingly, the Pearson correlation coefficient (ρ) between the ground truth and the reconstructed data on the same data set was significantly higher than the R^2 values (Fig. 3**A**: $\rho = 0.47$ (± 0.013); 3**B**: $\rho = 0.49$ (± 0.014)) The discrepancy between R^2 and ρ suggests that although we correctly identify the shape of the unobserved time-series, the scaling is not perfect. This can be understood by considering that movements can be performed at varying speeds and the regression only takes into account the average movement velocity.

In a second test, we compared the results from our method with the simpler approach of estimating the movement of the missing fingers by calculating the regression parameters over the unsegmented data from 5 of our 7 subjects and estimating the missing dimensions on the remaining two subjects (Fig. 3, gray columns). We find that although our novel method only takes into account a fraction of the data available, it performs significantly better (Student t–test, $p < 0.05$) than the second approach which takes into account all the data – this becomes particularly evident when trying to estimate the movement of three fingers.

Preprint submitted to 6th International IEEE EMBS Conference
on Neural Engineering. Received June 14, 2013.

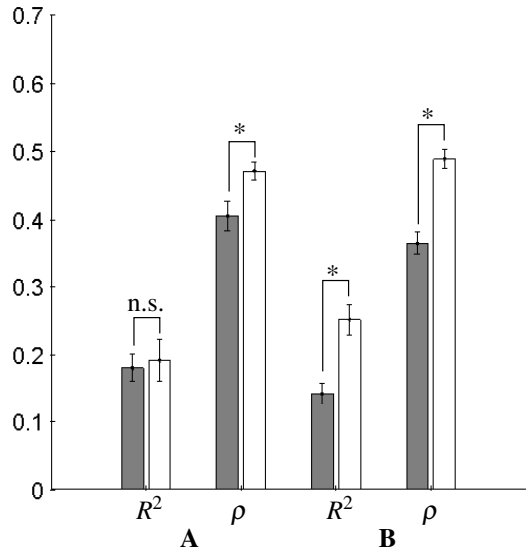


Fig. 3. Performance of our method (white) against regression over all the observed data (gray). (**A**) Estimate of the trajectory of fingers 3–5 from the observation of fingers 1 and 2 (**B**) Estimate of the trajectory of fingers 4 and 5 from the observation of fingers 1–3. Bars indicate standard error. *:$p < 0.05$. See text for details.

IV. CONCLUSIONS

In this paper we presented a method which is capable of accurately predicting the movement of missing finger joints from the remaining fingers. Instead of relying on direct information such as EMG of the muscles, we decompose training data into movement primitives and exploit the typical correlation patterns which are characteristic for those primitives. A probabilistic representation of a large data set containing everyday movement segments is generated during a pre-processing step and allows for movement prediction in real-time. We successfully tested our method by reconstructing the movement of missing fingers on a large data set of everyday tasks [18] and showed significantly better performance than the un-segmented approach.

We propose to integrate this method into the conceptual design of an improved neuroprosthetic device which on top of decoding intention from a variety of brain-machine interfaces (BMIs) such as EMG, intra-cortical electrodes or eye-tracking can make use of prior information about human movement statistics to improve the control. This would be of particular interest if we for example consider that it may well be possible to predict the movement of the fingers by only observing the movement of shoulder and elbow or the other hand, such as would be the case in a person suffering from hand amputation. Such a prior is especially of interest as current brain read-out mechanisms fail to provide the necessary information throughput to allow patients to accurately and naturalistically control a prosthetic device as complex as the human hand.

Based on the promising results laid out in this study, we suggest to extend the approach to different body limbs while testing different approaches to segmentation and regression to allow for truly naturalistic control of a prosthetic limb.

V. ACKNOWLEDGEMENTS

We thank Jovana Belic for collecting the data set. AACT is supported by the National Research Fund, Luxembourg (1229297).

REFERENCES

[1] DaM Taylor, SIH Tillery, and AB Schwartz. Direct cortical control of 3d neuroprosthetic devices. *Science*, 296(5574):1829–1832, 2002.

[2] M Velliste, S Perel, M C Spalding, AS Whitford, and AB Schwartz. Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(7198):1098–1101, 2008.

[3] LR Hochberg, MD Serruya, GM Friehs, JA Mukand, M Saleh, AH Caplan, A Branner, D Chen, RD Penn, and JP Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, 2006.

[4] LR Hochberg, D Bacher, B Jarosiewicz, NY Masse, JD Simeral, J Vogel, S Haddadin, J Liu, SS Cash, and P van der Smagt. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.

[5] JR Wolpaw and DJ McFarland. Multichannel eeg-based brain-computer communication. *Electroencephalogr. Clin. Neurophysiol.*, 90(6):444–449, 1994.

[6] JR Wolpaw and DJ McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.

[7] GR Müller-Putz, R Scherer, C Brauneis, and G Pfurtscheller. Steady-state visual evoked potential (ssvep)-based communication: impact of harmonic frequency components. *J. Neur. Eng.*, 2(4):123, 2005.

[8] S Bitzer and P van der Smagt. Learning emg control of a robotic hand: Towards active prostheses. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2819–2823. IEEE, 2006.

[9] TA Kuiken, LA Miller, RD Lipschutz, BA Lock, K Stubblefield, PD Marasco, P Zhou, and GA Dumanian. Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study. *The Lancet*, 369(9559):371–380, 2007.

[10] TA Kuiken, G Li, BA Lock, RD Lipschutz, LA Miller, KA Stubblefield, and KB Englehart. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *JAMA*, 301(6):619–628, 2009.

[11] WW Abbott and AA Faisal. Ultra-low-cost 3d gaze estimation: an intuitive high information throughput compliment to direct brain–machine interfaces. *Journal of Neural Engineering*, 9(4):046016, 2012.

[12] RA Roeschlein and E Domholdt. Factors related to successful upper extremity prosthetic use. *Prosthet. Orthot. Int.*, 13(1):14–18, 1989.

[13] J Davidson. A survey of the satisfaction of upper limb amputees with their prostheses, their lifestyles, and their abilities. *J. Hand Ther.*, 15(1):62–70, 2002.

[14] C Cipriani, F Zaccone, S Micera, and MC Carrozza. On the shared control of an emg-controlled prosthetic hand: analysis of user-prosthesis interaction. *Robotics, IEEE Transactions on*, 24(1):170–184, 2008.

[15] E Bizzi, SF Giszter, E Loeb, FA Mussa-Ivaldi, and P Saltiel. Modular organization of motor behavior in the frog's spinal cord. *Trends in Neurosci.*, 18(10):442–446, 1995.

[16] KA Thoroughman and R Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407(6805):742–747, 2000.

[17] AAC Thomik and AA Faisal. Deriving motion primitives from naturalistic hand movements for neuroprosthetic control. *BCCN 2012 Munich*, page 201, 2012.

[18] J Belic and AA Faisal. The structured variability of finger motor coordination in daily tasks. *BMC Neuroscience*, 2011.

[19] D Haber, AAC Thomik, and AA Faisal. Simple unsupervised time series segmentation for processing high-dimensional neurological data streams in linear time. *NIPS - in review*, 2013.