First Year PhD Report
Extracting Motion Primitives from Natural Handwriting Data

Ben H Williams
Institute of Adaptive and Neural Computation
School of Informatics
University of Edinburgh
5 Forrest Hill, Edinburgh UK
ben.williams@ed.ac.uk

August 10, 2005

Abstract

Movement selection and control is a very difficult inverse dynamics problem for robotic control that humans and animals accomplish easily. For the past 10 years there has been a popular theory that biological movement is made up of sub-routine type blocks, or 'motor primitives', with a central controller timing the activation of these blocks, creating synergies of muscle activation. Using machine learning techniques, this project addresses the possibility of extracting useful, and repeatable motor primitives from handwriting data.

1 Introduction

Animals and humans have evolved a complex motor system for the reason that our bodies have many joints and muscles that control posture and movement. Biological movement can be thought of in a high dimensional space, or 'Joint Space'. This high dimensional space can be represented in 3 dimensions using a framework of body dynamics. Therefore, any one point on the body framework can be represented in the high-dimensional joint space, or the 3 dimensional movement space. Movement tasks are specified generally in movement space, for example in the form of move end point of limb from position $[x, y, z]$ to position $[x', y', z']$, maybe avoiding some obstacle. This means that most movement tasks, however simple, can be accomplished in an infinite number of ways (Wolpert et al., 2001). However, humans and animals find solutions to movement tasks that are both repeatable to some extent across trials and circumstances, and subjects (Wolpert et al., 2001). Despite this repeatability, we are also very adaptable to new tasks, and can quickly adapt learned movements to cope with new environments (Davidson and Wolpert, 2003). Obviously, we do not have simple pre-programmed subroutines of movement, as this would make us very un-adaptable, but at the other extreme, it is unlikely that we solve the complete set of Newtonian force and motion equations to find an optimal movement solution for every situation, as this would involve a good deal of computation. In fact, as we do not have a complete knowledge of the world state at any one point, then the solutions to these equations would have to be statistical in nature, further increasing the computation required for an accurate result. Although some form of optimisation must be going on, we do not know the details of A) What is being optimised, and B) How it is being optimised.

(Bizzi et al., 1995) published a paper that suggested that the way motions are constructed and maybe even planned is by activating a series of overlapping primitives, or as they put it, superimposing force fields in which the limbs move. This theory has been supported by several studies into reaching and grabbing tasks since then (Wolpert and Kawato, 1998; Flanagan et al., 1999). Humans and animals have a very useful ability to observe another actor's movement and copy it. This can be done with partially observed, single trial learning, even when the observed actor does not have the same body configuration as the subject. This would be a very useful characteristic for robots to possess, as it would mean that their movements would not have to be pre-programmed (Alissandrakis et al., 2002; Rao et al., 2004; Matarić, 2000). There has been much work on recording the dynamics of all aspects of natural human movement. To categorise it and dissect it, people have tried to infer motor primitive type sub-blocks of movement from the sequences (Wolpert and Kawato, 1998; Matarić, 2004). Most of these attempts have pre-partitioned the sequences into movement sub-blocks, then extracted principal components of these blocks. However, this is a rather artificial way of extracting information from data. Although we are assuming that the movement is made up of motor primitive sub-blocks of motion, we are not placing any *a priori* constrictions on how long these sub-blocks can be, and when they are triggered, or even how many are combined to make up a typical movement. As these studies try to record complete or partial body configuration trajectories, there are obvious difficulties with data accuracy and realism, due to the limitations of the sensors, and the difficulties of creating a natural feeling environment for the subject. Although the data recorded in joint space is of a higher dimension, and therefore more complete, it is possible to record data from movement space, which alleviates some of the recording accuracy issues. This is difficult in reaching and grabbing type tasks, as there is no easy way of recording movement in 3 dimensions. However, handwriting is a particularly developed form of movement, surely rich in different motor primitives, and easily recordable using a digitisation tablet. The digitisation gives a 5 dimensional output characterising the pen position and tip pressure. The dynamics of this vector over time will reflect the hand motion and therefore contain projections of motor primitives. The task is therefore to create a learnable model that represents the activation of these primitives.

2 Literature Review

2.1 Background

Natural handwriting has been a much studied topic in the past 30 years both in academia and in industry, with most of the research focussing on recognition. For a review on handwriting recognition, see (Meulenbroek and Gemmert, 2003; Beigi, 1993). For the purpose of this project, it is possible to split the topic into 3 steps, with different papers focussing on each step. Firstly that of analysis of natural handwriting, for extraction of information from the data eg. (Marquardt et al., 1999; Kharraz-Tavakol et al., 2000). Secondly, there is generation of handwriting, and the generalisation to generation of humanoid movements, in the robotic movement imitation domain eg. (Wolpert and Kawato, 1998; Matarić, 2004; Wada et al.). Finally, the most researched area being that of handwriting recognition mainly because of Human Computer Interaction applications, an application area that naturally includes the related topic of speech recognition eg. (Cho and Kim, 2003; Omar et al., 2002).

2.2 Motor Primitives

The number of strategies in which humans and animals can accomplish any single motor task in the real world has an infinite number of solutions. Despite this, we show consistency both in varying situations, and across subjects (Todorov and Jordan, 2002; Matarić, 2004; Wolpert et al., 2001; Wolpert and Kawato, 1998). This suggests that we have some common low level movement behaviour set, or movement primitives, that are similar in broad terms from one subject to the next.

Motor primitives were first found in frogs (Bizzi et al., 1995) where stimulation of a single spinal motor afferent triggered a complete sweeping movement of the frog's leg. In fact the stimulation of specific sites on the spinal column induced force fields for the limb. These force fields were linearly superimposed when triggered concurrently. Thinking of these force fields as motor primitives implies that the motor control and hence behaviour of the frog may be built up of similar superimposed motor primitives.

There has been much work investigating these motor primitives, or muscle activation synergies. (d'Avella and Bizzi, 2005; d'Avella et al., 2003) used electromygraphic recording techniques to record the natural activation of frog leg muscles. Using component factorisation techniques, they showed evidence of modularisation of the motor control system. Extending this model to higher levels of motor control, (d'Avella and Bizzi, 1998) stimulated the vestibular nerve in several different places, and performed PCA on the resultant force fields. They showed that 94% of the total variation of the data could be explained using only 4 principal components, meaning that the movements were probably built up in a modular way.

For an overview of the inverse dynamics problem that must be solved for motor planning and control, see (Mussa-Ivaldi and Bizzi, 2000). They propose a possible solution through the use of motor primitives and an extension of the established spinal cord basis of muscle synergies to higher motor planning areas. For a review of the modularisation of motor control in the spine, see (Bizzi et al., 2002).

The breaking down of complex movements into superimposed sub-routine type motor primitives is desirable for robotic control, and the ability of the robot to mimic novel movements, as it places constraints on the action selection space. There is also a strong body of evidence to support a sharing of resources between motor control, and action perception. (Gallese et al., 1996) recorded from neurons in the pre-motor cortex, and found that they responded similarly when a specific action was performed by the subject, and also when the same action was observed. These neurons have become known as mirror neurons, and they are the strongest evidence that the brain contains structures that deal both with action and perception. The implications for robotic control are that to create a control system that replicates animal movements, it should also be capable of recognising the same movements.

There has been some work into extracting motor primitives from natural human whole body movement, and then replicating them with robots (Fod et al., 2002; Schaal et al., 2004; Ijspeert et al., 2003). So far, this has focussed on the activity of each primitive as a vector at each time step, therefore the primitives have no temporal dimension. This is most likely not the case in biological systems, with each primitive being a time extended block of specific movement, activated independently at different time points. Furthermore, each primitive would have non-exclusive control of a subset of muscles, with muscle commands summing probably non-linearly.

The hypothesis that complete movements are made up of superimposed, time extended blocks of movement implies that somewhere, there must be a timing circuit that 'fires' the appropriate primitive at the appropriate point. There is evidence that the cerebellum is involved with motor function, and more specifically with timing of motor function, and perception. (Meegan et al., 2000) showed that learning rhythms perceptually improved performance in motor tasks which involved the learned rhythm. (Dennis et al., 2004) showed a relationship between cerebellar volume in children and performance in motor timing tasks. (Penhume et al., 1998) used PET imaging to show that the cerebellum provides a supramodal contribution to motor timing tasks. They hypothesise that the cerebellum provides the necessary circuitry to extract timing information for the sensory and motor system.

2.3 Motor Planning

The reason for the interest into motor primitives is that they approach the problem of the transformation from kinematics to dynamics space. This has been looked at from a biologic perspective by (Padoa-Schioppa et al., 2002, 2004). They believe that there are forward internal model representations in the supplementary motor area, and that these models solve the kinematics-dynamics transformation.

Motor primitives go some way to solving the kinematics-dynamics transformation, but there must still be some form of controller to select which primitive should be active at which points in time. There must in effect be some internal world model to predict the outcome of a certain primitive. A model proposed by Wolpert involves multiple blocks that individually characterise a certain primitive, with each block providing both a forward (predictor) and inverse (controller) model.

There has been much work by Wolpert and colleagues (Wolpert et al., 2001; Davidson and Wolpert, 2003, 2004a; Hamilton et al., 2004; Körding et al., 2004; van Beers et al., 2004; Witney and Wolpert, 2003; Caithness et al., 2004) looking at how our internal forward models are structured and applied in motor adaptation tasks. In general, they show that there are multiple forward and inverse internal models, and that the movement strategies are selected so as to minimize muscle activation noise, and therefore end point error. For an overview of their research see (Wolpert and Kawato, 1998). Their model tries to explain both motor adaptation experimental results, and motor optimisation strategies. It has long been noted that learning different tasks in a similar environment is more difficult than learning two unrelated tasks. The tasks interfere with each other. However it is possible to switch motor behaviour in different environments, for instance motor skills when playing tennis, or driving a car. They believe that there is a modularisation of internal models, which can be independently activated for specific circumstances. This can go some way to explaining the phenomenon of motor learning interference. Furthermore, they hypothesise that movement strategies are evolved to minimise end-point error. This is a novel alternative to the common idea of minimising some path distance metric, such as absolute distance, or average muscle force needed, or jerk minimisation. They have obtained many experimental results to support this theory, which takes into account the fact that muscle noise is proportional to muscle activation.

2.4 Modelling Techniques

To extract information from natural data and to then model the system in a generative manner, there are a number of computational modelling techniques.

A straightforward, and simple approach to feature learning was presented in (Chiu et al., 2003). They look for re-occurring features in a time series. However, they do not allow for over lapping of several different features, as the Factorial Hidden Markov Model (fHMM) model does (see below). However, it may be a good way to approach the problem if other techniques prove to be too computationally demanding.

A short paper (Vinciarelli and Bengio, 2001) looks at how Hidden Markov Models (HMM) can be used for character recognition in handwritten digits. They use a separate HMM for each character, and pre-process the data using Independent Component Analysis (ICA) to help with an independence assumption between the HMMs. In effect, they were decorrelating the data, so that the output covariance matrix could be diagonal. Although they are taking a similar approach to us, they are addressing a different problem, without caring to break the characters down into primitives, they are simply trying to distinguish one character from another.

For a review of ICA methods, see (Hyvärinen, 1999). (Roweis and Alwan, 1997) attempt to use a HMM based approach to infer articulatory information in speech signals. This is similar to what we are trying to do, in that they try to represent speech in articulation space (configuration of mouth etc), whereas we are trying to represent movement in primitive space.

A relevant model is the Factorial Hidden Markov model. This is explored by Ghahramani (Ghahramani and Jordan, 1997). In this paper, they look at different methods of learning in fHMMs. To implement the Expectation Maximisation technique to infer the optimal parameters, they use different approximations to speed up the algorithm. Firstly, they implement the exact method, then they use using Gibbs Sampling to speed up the algorithm, then using completely factorised variational methods, and finally structured variational methods. They provide derivations for most of the necessary equations. Their approach uses the standard Baum-Welch forward backward algorithm to calculate the expected states. These methods are more thoroughly addressed in a paper by Bilmes (Bilmes 1998). The fHMM modelling done in this project is based on the Ghahramani paper, so for more details, see section 4.1.

3 Project Outline

The original aim of the project, or at least the founding motivations were to create a program that attempts to learn motor control of a virtual pen using visual feedback and some sort of reward type learning, in a similar way that children learn how to handwrite from examples of simple pen strokes that they are encouraged to copy.

To approach this problem, the system needs a repertoire of movements that it uses to control the pen. In fact this is a very important, and contemporary problem in robot control (Wolpert et al., 2001). Once the appropriate motor primitives have been identified, and they can be reliably extracted and separately analysed, then the problem of how children learn handwriting can be studied at a more detailed and relevant level.

There has been much work into motor primitives with a view towards better robotic movement control (Matarić, 2004; Ijspeert et al., 2002). However, there is not as yet any conclusive solution, possibly because the approaches have mainly been from a signal processing perspective rather than a biologically plausible model. This is probably because of the similarity of the problem with the Source Separation Problem, in speech recognition. However, the way that muscle synergies are activated and combined is not the same as the way that sound signals are linearly superimposed. To propose a learnable model, we must consider how the brain controls our bodies. For more detail on this, see section 2.2.

Much work has been done on primitive extraction in 3 dimensional movement, for instance reaching and grasping. The problem with this paradigm is that accurate position data is difficult to collect. An area that spans motor control, visual character recognition, human computer interaction and even language is that of handwriting, or more fundamentally, pen control. Using the medium of pen control to study motor primitives means that much data from many sources can easily be collected. It is also much easier to carry out adaptation experiments using a computer screen as the paper for visual feedback, and a digitising tablet to capture the motor output separately.

3.1 The Piano Model

There is much evidence to support the fact that once a particular movement has been commenced, it cannot simply be switched off. Rather, to modify a movement trajectory, the movement is superimposed with another movement (Davidson and Wolpert, 2004b). This evidence suggests that there is a sub-routine type of movement activation, where the sub-routines are not able to be quickly altered, but their individual activation can be a fast and globally relevant process.

To define terms within this project, the sub-routines of movement will be referred to as motor primitives. Therefore, to formalise the model in a generative way, the output of the system Y, is defined in (1),

$$Y(t) = \sum_{(i,j)} \alpha_{ij} z_i(t - \tau_{ij}) \tag{1}$$

where α_{ij} defines the activation strength of a particular primitive, $z_i(t)$ are the primitives, and τ_{ij} represents the time of activation of the primitive i.

In this definition, there are i primitives, with each primitive being triggered j times within the sample window, at time τ_{ij}. This model has been called the Piano Model because of its similarities to the operation of a piano being played, where the timing controller (the pianist) presses each key at the appropriate time in the piece of music. The keys on the piano produce a time extended clip of sound, which are superimposed to create the music that is heard by the listener. The crucial point is that the only dependence that the music has on the pianist is the timing and choice of keys he presses, with associated pressure[1], in the same way that in our motor primitive model, the precise movements are not dictated by a central controller, rather than their exact timings. Figure 1 shows a diagram to illustrate the piano model. Note that the primitives are of a higher dimensionality than shown.

The generative implementation of this model is not difficult, however, the learning of the primitives is more complicated, as the number, the length, and the position of the primitives is unknown (as well as the shape of the primitives). Therefore, to tackle the problem, simpler models were explored to try to extract

[1] This is debatable, as pianos may respond to the speed of key press, the duration of the key press, and what other keys are being pressed at the time. Pianos also have pedals to create different effects. The analogy is not intended to be exact.

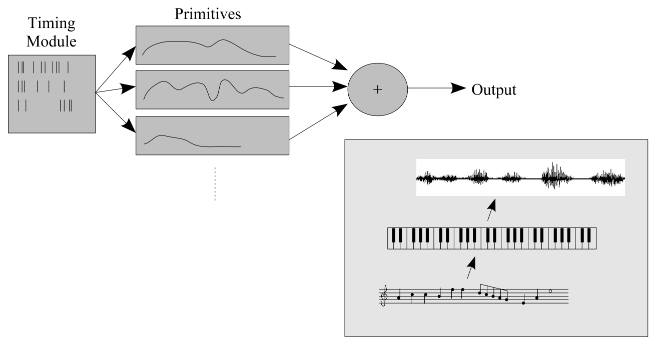

Figure 1: The Piano Model. The model is segmented into a timing part and a movement subroutine, or primitive part. The timing information is encoded in spike positions, possibly in a similar way to how biological neurons may encode the movement. The movements are encoded as multidimensional muscle activation synergies in biology. Here they have been simplified to one-dimensional signals. The parallel with written music being translated into auditory music is also shown.

the shape of the primitives from natural data. Using ICA to separate the data into independent components yields some interesting results (see section 4), however, it is difficult to conclude that in principal, ICA should theoretically be capable of extracting primitives due to a lack of observable dimensions. In effect, the dimensions are increased by taking multiple samples of the same primitive. However, to do this we require precise time indexing of the primitives. To further reduce the attraction of using ICA, the technique was unable to convincingly recover test primitives from artificially generated data. In effect, there needs to be some form of pre-processing to ensure that the datums for the ICA contain stationary primitives within the sample windows.

Assuming discrete time steps, an appropriate modelling framework for the Piano Model is that of Factorial Hidden Markov Models. These are the same as standard HMMs, but with multiple, parallel and independent hidden state chains, as seen in Figure 2.

The output mean is a linear sum of the factor means,

$$\mu = \sum_m W^{(m)} \cdot S^{(m)} \tag{2}$$

where μ is the output mean, W is the matrix of contributions towards the mean, and S is the hidden state.

Attributing a single primitive to each state chain, and taking the observable output to be a single Gaussian, gives a model that is the same as the Piano Model given some extra constraints: All the Markov chains should remain in state 0, contributing towards a zero mean output until a particular primitive is triggered, at which point the relevant Markov chain progresses monotonically through the states with high probability until the last state is reached, after which there is a high probability that the Markov chain returns to state 0. This

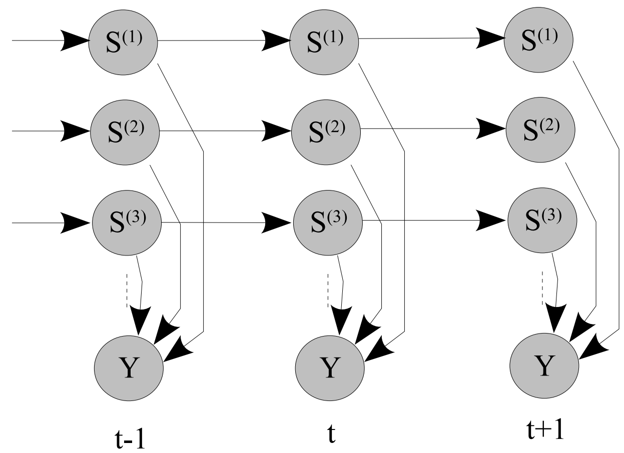

Figure 2: Graphical representation of an fHMM, showing the independence of the separate Markov chains. Although the observable output is dependent upon the state of the entire system, the internal states evolve with no interdependencies. With extra constraints, it is possible to equate each Markov chain, or factor to a separate primitive.

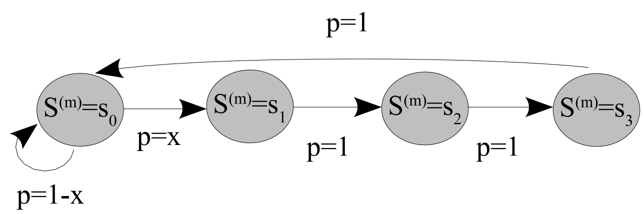

Figure 3: State change probabilities. This shows how the state change probabilities were constrained so that the individual Markov chains can correspond to a time extended primitive.

state change organisation can be seen graphically in Figure 3.

Factorial HMMs are guaranteed to converge to a local minima in the parameter search space. Therefore, if the initial state of the system is provided with enough *a priori* knowledge about how the primitives should operate, then performing EM on the model should give optimal primitives.

Using the primitives effectively as feature detectors, handwriting can then be represented in primitive space, thus allowing more robust recognition prior to any language or higher level models. The primitives could also be used as the output of a neural implementation of a handwriting model, to explore the learning, connectivity, and operational issues of a more biologically plausible implementation of such a model.

It would be interesting to see how, if at all, the motor primitives of a human adapt over time, as this would show how they relate to long-term adaptation, or motor learning. Particularly when new tasks are attempted, such as practising mirror writing. Using a digitisation tablet, handwriting data can be quickly and easily acquired, in a very natural writing environment for the subject.

In general, although all the children in the classroom will all learn handwriting from the same teacher, with the same teaching materials, they will all develop different styles of handwriting. This could be due to some difference in the structure of their arms and hands, or it could be due to a difference in the control policy for their hands. It is possible that the motor primitives are roughly constant throughout our lives, and built in to our brains from an early age. If this is the case it may explain way people converge towards different styles of handwriting given the same training procedures.

It would also be interesting to explore the mapping from movement space to joint space in terms of the primitives. Have the primitives evolved to minimise noise in muscle control, as is suggested by Wolpert (van Beers et al., 2004; Hamilton et al., 2004)? Using these techniques to extract primitives, can joint space data be analysed in a similar way, thus providing a repertoire of movements for a robot?

3.2 Character Primitives vs. Motor Primitives

Clearly, there will be recurring shapes in characters that are inherent in the character set rather than naturally occurring motor primitives. A number of ways to overcome this could be examined.

Firstly, not only Latin style handwriting could be collected, but other languages, and also scribbling. Analysing people's handwriting in more than one language (eg. Greek and English) would provide different character sets from the same person. There are many people in the University of Edinburgh who have two natural written languages, so finding subjects would not be a problem. Furthermore, a simple study in which people are taught an invented character set could be conducted. The analysis of their motor primitives should show mostly common motor primitives for both natural characters and the new set. Also, the learning transition phase would be very interesting to analyse. One hypothesis is that during learning, the motor primitives would be less well integrated, as the timing and intensity of their activation is the task being learned.

To help distinguish the primitives inherent in the character set from the primitives inherent in the subject's movement, it may be possible to take a baseline of the perfect character set as used for training the subject's handwriting originally, and examine the deviations from this.

An analysis of free-form scribbling may also provide reoccurring primitives, which, as the task is less constrained, would provide a better insight into the natural primitives used by the subject.

It should be noted that character sets may well have evolved to be directly based upon common motor primitives. However this is not currently provable, and a competing theory could be be that character sets have evolved to maximise visual differentiation between different characters/words. The shapes inherent in a character set are a function of the character set rather than some built-in set of motor primitives, and so should be considered separately, even if there is a developmental link. A way to visualise clearly the distinction between character primitives, and motor primitives is by considering the downward stroke in the letter I. This stroke is common to many characters, and is likely to be picked out by a motif extraction algorithm running on some handwritten text. Naïvely, we may assume that this is a motor primitive, as it is difficult to see how a simple line could be broken down into constituent parts. However, thinking about how the hand itself had to move to manipulate the pen in order to create the line, we can see that perhaps several different motor primitives were used to create this line. A rigourous statistical analysis of similar shapes within handwriting should be able to separate truly independent movement sub-blocks.

3.3 Robotics

On paper, the pen has 2 degrees of freedom (DOF), or 3 if one includes the raising of the pen, and the intensity of the line drawn. However, the arm and the hand have many more DOFs. It would be interesting from a robotic control perspective to create a higher DOF robotic 'hand' with control of a pen, that could recreate the primitives on paper, and then to examine the primitives that are given rise to in the separate motors of the hand.

3.4 Handwriting Style

One of the principle problems faced by handwriting recognition programs is that of different handwriting styles. It is also a problem that children face when trying to read a handwriting style that is vastly different from their own, although once tackled, humans can easily read many different handwriting styles, without explicit classification of the style. Although different cultures and countries generally have different handwriting styles, it is also interesting to note that within a country, and even within a group with a common training set (ie. Pupils from the same school), many different handwriting styles emerge. Moreover, although handwriting styles diverge around the time of acquisition, they do not appear to change much once acquired. This leads to the hypothesis that handwriting style is determined to a large extent by the motor primitives at the disposal of the subject. Furthermore, this suggests that if the set of motor primitives was subtly altered, it should affect the handwriting style.

4 Work accomplished / Underway

Using some handwriting trajectory data from the UNIPEN standard data set openly available on-line [2], I started by creating a framework to display and segment the raw data into characters. Figure 4 shows a screenshot of this framework. Most past studies have characterised handwriting data using the number of times the y-axis velocity crosses zero per character (Marquardt et al., 1999).

[2] http://unipen.nici.ru.nl/

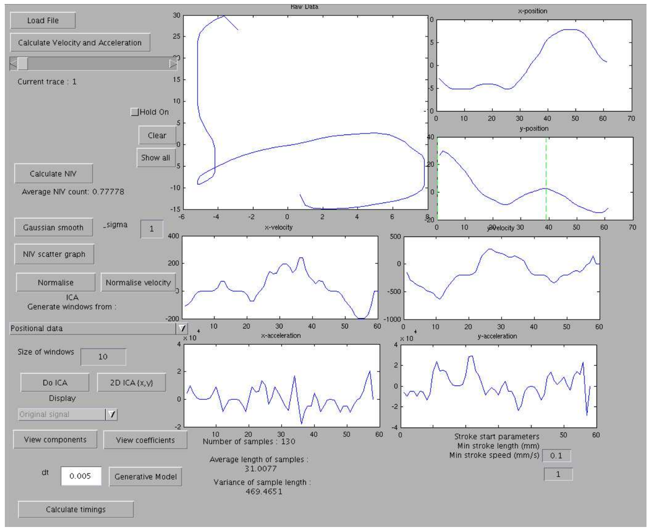

Figure 4: Screenshot of ICA program framework. This is a Gaussian smoothed sample of the UNIPEN data set, with the calculated velocity and acceleration profiles.

This analysis was incorporated into the program, to allow better comparison with previous studies. Also a small amount of pre-processing was incorporated, such as a low pass Gaussian smoothing filter. All the data available in fact was quite noisy, seemingly acquired at a lower resolution than stored. This is perhaps due to a lack of post-processing at the data acquisition end that would normally be used before recording.

Using this framework, it was easy to calculate velocity and acceleration information for the pen trajectories. The first attempt to extract motor primitive information from the data involved using ICA algorithms to separate statistically independent components of the data, under the hypothesis that these components would be related to the motor primitives. This is because in the proposed model, the Piano Model, (see section 3.1) the primitives are statistically independent from each other, with the statistical dependencies necessary to create a coherent movement arising from the timing of the triggering signals. Therefore, ICA should identify the primitives as the most statistically independent components. The FastICA algorithm developed by Jarmo Hurri, Hugo Gvert, Jaakko Srel, and Aapo Hyvrinen, available on-line[3] was used. Positional data, and velocity data was analysed. Datums were defined in several different ways. Firstly the simplest way, where each time slice represented one datum. This gave the result that the slant of the writing was picked out as one principal component, thus skewing the axes of the frame of reference (the paper).

Figure 5 shows a collection of some of the components generated by the FastICA algorithm, taking 60 time samples from each character as one datum.

[3]http://www.cis.hut.fi/projects/ica/fastica/code/dlcode.html

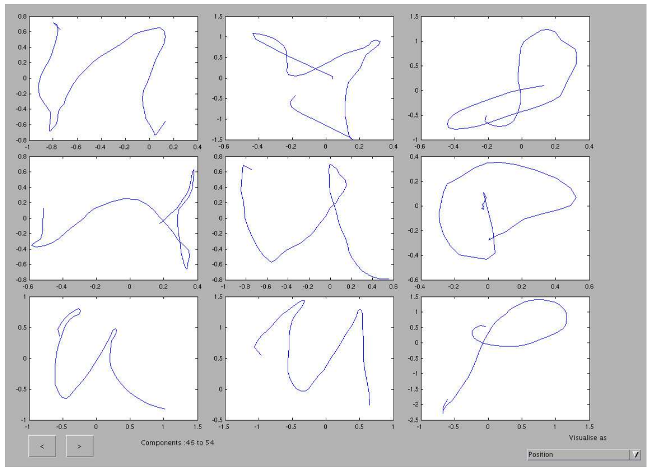

Figure 5: Example of ICA components extracted. Here, the datums were created by taking whole characters. As the data set was not very large, little generalisation occurred, resulting in ICA components largely representing individual characters. They are plotted here as x, y co-ordinates.

Clearly, the components are single characters, and show little generalisation. This is because the ICA algorithm is not able to partition the characters into segments that are more readily generalisable. In Figure 6, we see the coefficients used to perfectly reconstruct the sample seen in Figure 4, which suggest that the data has a very sparse representation in the coefficient matrix. To examine this further, a generative model was developed to explore how the components could generate data. This model adapts the linear mixture model of ICA, removing the coefficient matrix, and replacing it with an extended matrix of timing information, thus implementing a generative form of the Piano Model, with pre-defined primitives specified by the ICA component matrix.

The ability to control the timing information and individual primitive activation strength allowed us to select the largest component, and compare it with the original character, also to add noise to the temporary coefficients used for the character in question, and to select random coefficients. In Figure 7, we can see the effects of adding noise to the coefficients. This showed that there was virtually no generalisation across characters with this approach, with most components representative of a single character, with small alterations, such as the slant at the beginning being dependent on many, smaller components.

Sliding windows were then used to provide several datums per character, as seen in Figure 8. This method provided more hope, however the components generated were dependent really on the distribution of samples selected from the whole data set. There is no guarantee that they represent distinct motor primitives.

The general problem with these approaches was that the length of the primitives was not known. This meant that the length of the windows could not be

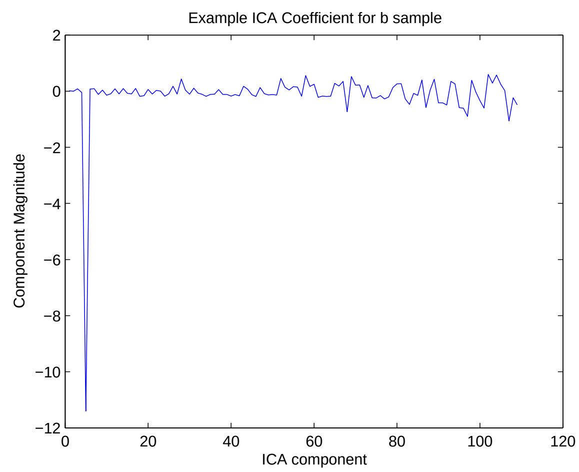

Figure 6: ICA coefficient. Single row of the mixing matrix from the ICA component decomposition, showing the coefficient values used to reconstruct a single sample perfectly. It shows the relatively sparse representation of the data.

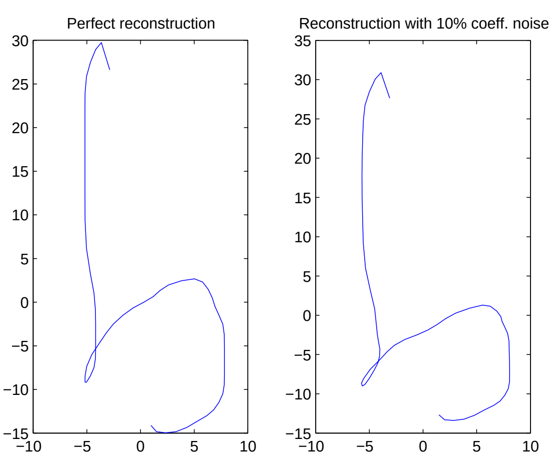

Figure 7: Reconstruction of data set. With 10% noise added to the coefficients, there is little change to the reconstruction of the data set. This shows that the components have not generalised much over the whole data set.

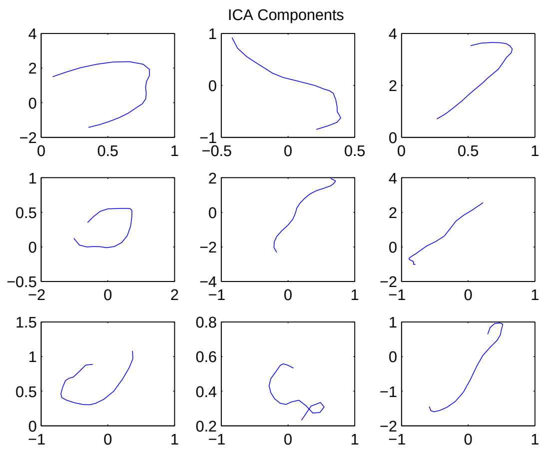

Figure 8: Example of components extracted using ICA with sliding windows to generate the datums. Although the components seem to have provided a more generalised representation of the data, it is difficult to see whether they correspond to motor primitives.

defined. Also there was the problem of the timing of the primitive. As a single delta time shift completely changes the datum for the ICA, it was difficult to know how to position the windows.

To attempt a simple best-fit type reconstruction using the primitives generated, an extension to the program allowed for several different fitting algorithms to be tested on the primitives generated. An example of one algorithm working with the primitives is shown in Figure 9.

Using a sensible algorithm to choose appropriate components to be activated at particular times makes a reasonable reconstruction possible. It should be noted that as the components are based on a data set that at best only partially encompasses the character in question, it could not be expected that the components would be able to perfectly reconstruct the data, as with the coefficients for the original, randomly sampled data set.

To test how well this system recovers artificially generated primitives from a trial data set, a one-dimensional example was used. There were two primitives present, one being a sine-wave, and one a ramp. There was also noise present. Figure 10 shows a sample of the components calculated.

Clearly, the components show aspects of the primitives, but because the windows were not guaranteed to be positioned in the same place with respect to the start of the primitives, the component separation algorithm sees them as completely different components (as they are represented along different dimensions).

ICA was investigated for its ability to extract components that are statistically independent from each other, thus if there is partitioning in some under-

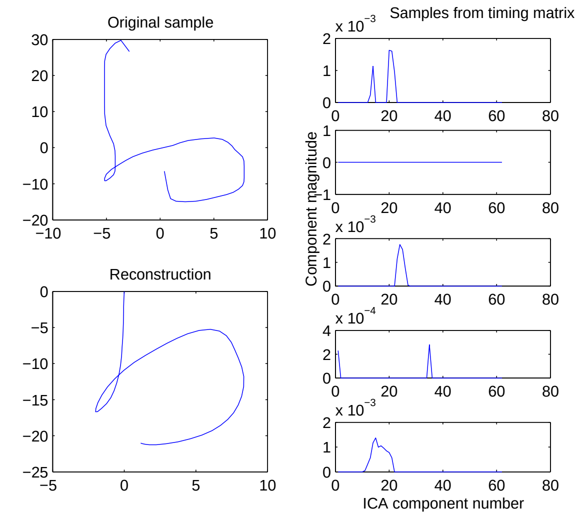

Figure 9: Example of a simple fitting algorithm trying to reconstruct the data. The components used were generated from ICA performed on datums taken from sliding windows in the velocity domain. The reconstruction looks like a reasonable approximation to the original sample.

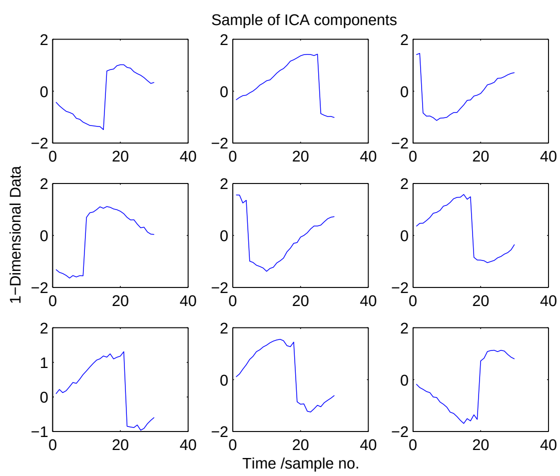

Figure 10: Components after performing ICA on a dummy data set. The generative components were a sine-wave and a ramp. ICA fails to cleanly find the original primitives, although the primitives seem to be partially represented.

lying generative model, then ICA should separate the constituent parts of the data pertaining to the different parts of the model. This works well in source separation problems, where there are many dimensions to the observable data (taken from microphones in different positions). However, our data is of a low dimension compared to the hypothesised generative model. To get around this, we tried to take each complete primitive as a single datum, by performing ICA in the time domain. This means that ICA would be comparing different examples of the same primitive, and so should be able to generalise across different trials. Unfortunately, without some clear, and precise indication of where the primitives occur, they are non-stationary with time, and if ICA is performed in the time domain, the primitives are therefore non-stationary with dimension. It was concluded that ICA could not yield useful results without a separate triggering facility to detect the position of the primitives.

4.1 Factorial HMMs

The reason that ICA could not extract primitives from the data was that the ICA model assumes nothing about the structure of the generative model that gives rise to the data, except that the components of the model are not Gaussian. The Piano Model specifies much more about the behaviour of the underlying model. Given the restrictions of using ICA, the problem of how to implement the Piano Model was reconsidered. As discussed in section 2.4, one of the commonly used models to represent systems with unobservable states, and regular statistical dependencies between the different variables, and with mainly short term time dependencies, is the Hidden Markov Model (HMM). When there are many independent hidden states that are combined to give the single observable output, then it is possible to use a HMM to represent them, however the number of hidden states explodes exponentially (Ghahramani and Jordan, 1997). A distributed representation of the hidden state is more versatile, and gives rise to a variation on the HMM model, a factorial HMM. For more details, and justification of this model over a simpler HMM, see (Ghahramani and Jordan, 1997). The fHMM model is arguably closer to real life, as it allows multiple independent processes to evolve over time, interacting only to produce the output. Also it is possible to perform more efficient (and approximate) inference with the fHMM, by using a variational approximation of the model structure for the E-step, allowing a traditional Baum-Welch approach, with minor changes.

Therefore, to implement the Piano Model, it was decided that the fHMM model should be used. (see section 3.1). The paper by (Ghahramani and Jordan, 1997) was used to provide the details of the model. For a more complete description of the model, please refer to this paper. Here is a summary of the model dynamics.

Each hidden Markov chain, or factor, models one primitive. The observables are one dimensional real-valued outputs. The graphical model of an fHMM was shown in Figure 2. There are M independent Markov chains, and K states possible within each chain. As you can see in the graphical model, the hidden state S depends solely on the previous time step. As the Markov chains are uncoupled, the state transition probabilities can be factorised. The observables depend only upon the hidden state at a particular time step. The hidden state transition probabilities are specified by M separate $K \times K$ matrices. The observable output is defined as a multivariate gaussian. Here are the probability distributions that define the model dynamics:

$$P(\{Y_t, S_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{3}$$

$$P(S_1) = \prod_{m=1}^{M} \pi^{(m)} \tag{4}$$

$$P(Y_t|S_t) = \mathcal{N}(\mu_t, C) \tag{5}$$

$$\mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \tag{6}$$

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P^{(m)} \tag{7}$$

$$\tag{8}$$

The parameters are specified by these matrices: The initial hidden state probabilities, $\pi^{(m)}$ is $K \times 1$, the hidden state transition probabilities, $P^{(m)}$ is $K \times K$, the output means, $W^{(m)}$ is $D \times K$, and the output covariance, C is $D \times D$.

The structured variational approximation was chosen for the E-step inference. This means that each Markov chain needs a responsibility factor, h_t associated with it, that takes the place of $P(Y|S)$, where Y is the observed vector, and S is the hidden state. Here is the factor,

$$h_t^{(m)new} = \exp\{W^{(m)'} C^{-1} \tilde{Y}_t^{(m)} - \frac{1}{2} \triangle^{(m)}\} \tag{9}$$

where

$$\triangle^{(m)} \equiv diag(W^{(m)'} C^{-1} W^{(m)}) \tag{10}$$

and

$$\tilde{Y}_t^{(m)} \equiv Y_t - \sum_{l \neq m}^{M} W^{(l)} \langle S_t^{(l)} \rangle \tag{11}$$

where $\tilde{Y}_t$ is the residual error.

In the M-step, the parameter update equations used were

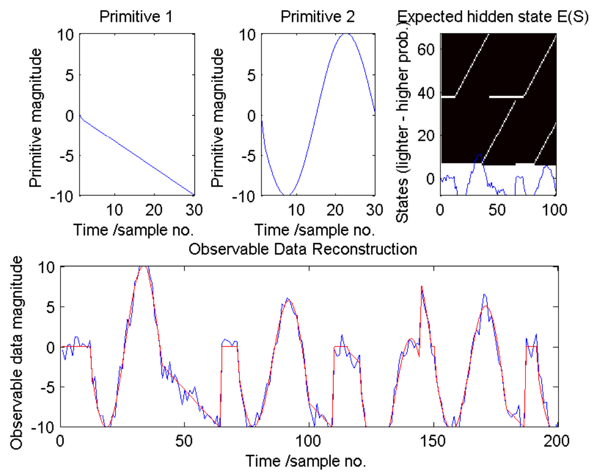

Figure 11: Sample data set generation. The top two figures show the primitives used, the bottom left figure shows the hidden states, and the bottom right figure shows the reconstructed data.

$$W^{new} = \sum_{t=1}^{T} Y_t \langle S'_t \rangle) (\sum_{t=1}^{T} \langle S_t S'_t \rangle)^{\dagger} \tag{12}$$

$$\pi^{(m)new} = \langle S_1^{(m)} \rangle \tag{13}$$

$$P_{i,j}^{(m)new} = \frac{\sum_{t=2}^{T} \langle S_{t,i}^{(m)} S_{t-1,j}^{(m)} \rangle}{\sum_{t=2}^{T} \langle S_{t-1,j}^{(m)} \rangle} \tag{14}$$

$$C^{new} = \frac{1}{T} \sum_{t=1}^{T} Y_t Y'_t - \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle Y'_t \tag{15}$$

where $W_i^{(m)}$ is the contribution of factor m to the output mean if in state i. W is a concatenated matrix of all these means, thus storing the primitives. Y_t is the observable output vector. $\langle S_t \rangle$ is the expected value of the hidden states at time t. π is a vector of the initial hidden state probabilities. $P_{i,j}^{(m)}$ is the hidden state transition probability from state j to state i in factor m. C is the output covariance matrix. (As we are dealing with one-dimensional outputs, it is simply the output variance.)

Firstly, to show that the Expectation Maximisation technique of learning the parameters and the hidden states is sufficient to recover the generating primitives, the model was tested on artificially constructed data.

In Figure 11, the first two plots show the primitives used, and the third plot shows the observable data in blue, and the hidden state sequence in white. The bottom plot shows the observable data with output variance in blue, and the

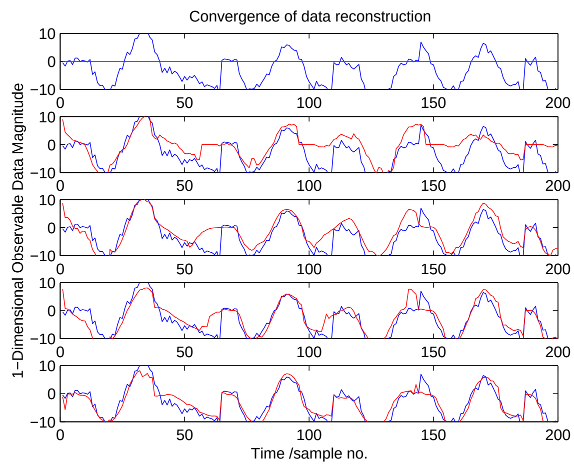

Figure 12: Five plots showing the progressively more accurate reconstruction of the observable data, after each iteration of the EM inference algorithm.

perfectly reconstructed data in red.

Figure 12 shows the observable state, reconstructed from the inferred hidden states after individual EM iterations, compared to the original observations. It can be seen that the EM algorithm does act to make the reconstruction of the data from the inferred hidden states converge towards the original observed data.

There were a few problems with convergence. These arose due to the heavy constraints placed on the state transitions. The states, representing the internal time sequences of the primitives, were constrained to progress monotonically. This provided expectations with very 'hard' probabilities, meaning that the E-step was very sure of the states. However, this meant that the M update became very unstable due to a very thin metric $(\sum_{t=1}^{T}\langle S_t S_t' \rangle)^{\dagger}$. To solve this problem, an identity matrix was added to the step, to increase the rank of the $\sum_{t=1}^{T}\langle S_t S_t' \rangle$ matrix. This slowed down the overall convergence of the algorithm, but prevented the W values from exploding.

The temporal dependence of the primitives is modelled in the hidden state transitions. However, the transition probability matrix does not take into account the linear, or rather monotonic state progression that is implied by such a model. Therefore, to add this constraint, the transition matrix was fixed so that the states could only progress in a monotonic fashion, and that the primitive could only be reset to the rest state after reaching the final state. In fact, the only probability that was learned was the probability of starting a primitive.

A further constraint was that each primitive should remain in a rest state until triggered. The output of this rest state should be zero. This was modelled first by simply setting the output of state 0 of each primitive to zero. This did not work, as the W update equation would take into account that the average

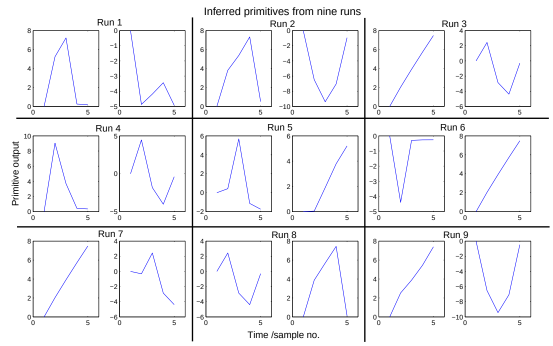

Figure 13: Reconstructed primitives. In these nine independent trial runs, it can be seen that partial fragments of primitives were identified by the fHMM algorithm.

observable output was zero, and would therefore try to make the combined average output of all the primitives to be zero, with the effect of shifting the primitives up or down. To resolve this, instead of setting the output of state 0 to zero, the whole primitive was shifted up or down by a fixed amount, so that the output of state 0 was zero. This means that the primitives kept their shape.

Now the problem was that the primitives tended to pick out parts of the original primitive, and as the transition matrix was constrained as detailed above, they could not find all the primitive. Figure 13 shows an example of the results produced, showing examples of 9 different initial conditions. Two primitives were used, a sine wave, and a ramp. We can see different parts of the primitives in the data, but none of the results perfectly represent the whole of both primitives.

A quick fix to this problem is to translate the primitives along the time axis, and re-run the algorithm. This was implemented by looking for zeros at the start of the primitive.

- If there are more than 2, then the primitive is shifted to the left so that there are only 2 zeros at the start.
- If there are less than 2 zeros, then the primitive is shifted right by one notch.

The EM algorithm is then re-run to optimise the model with these primitives as initial conditions. This procedure is re-run until all the primitives remain the same.

There are other solutions to this problem that may be implemented in the future, such as allowing a cyclic hidden state progression, with the rest state removed from the cycle.

Using the current model, it is possible to extract in a reasonably reliably way the original primitives from the data, as seen in the relatively successful results in Figure 14. Here we can see a reliable performance of the algorithm, successfully recovering both primitives in all samples.

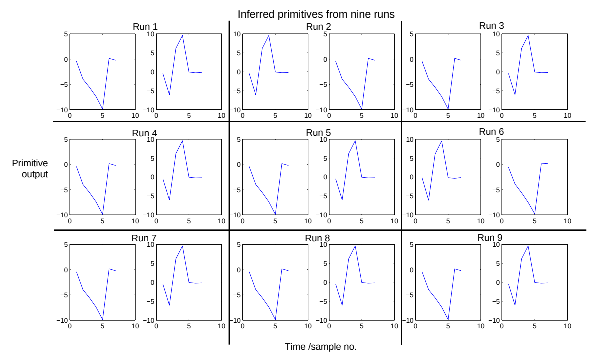

Figure 14: Reconstructed primitives. With the shifting procedure to identify the start of the primitive, these nine sample runs show that the fHMM algorithm is capable of finding the original primitives.

4.2 Summary of Problem History and Discussion

Once the computational issues with the algorithm were solved, the main problems have been associated with the adaptation of the generalised fHMM model to a more restricted version that corresponds to the Piano Model. The main difference is that the Piano Model has a 'rest state' in which none of the primitives are responsible for producing the observable output. In the fHMM model, this state is not present, and at any one point, a weighted sum of all the primitives is present at the output. However, as the weights corresponding to specific states are really the primitives themselves, it is necessary to specify that for all primitives, there is one state that corresponds to a rest state, giving a zero contribution to the output. To restrict the model in this way, the M-step was modified so that the $W^{(m)}$ values for the rest state were set to zero. The problem with this was that the other W values did not take into account this state when being updated. The rest of the primitive was effectively shifted so that its average was zero, as this was the overall average. To address this, the rest state value, instead of being simply set to zero, was subtracted from the whole of the primitive, thus maintaining the shape of the primitive, and ensuring that the rest state output value was set to zero.

The next problem was associated with the transition probabilities restriction. These were restricted so that the primitives must progress monotonically through the states, as seen in Figure 3. This meant that the transition probabilities were not free to adapt to add in extra states at the end of the developing primitive. In practise what was happening can be seen in Figure 13. Primitives were identified but only half way through the state sequence. The simplest way to approach this was to shift the primitive to the right or left, depending on the number of zeros at the front of the primitive, in the 'lead in' section. The problem with this approach, is that the lead in part can conflict with other occurrences of the primitive in the data just beforehand. One way to solve this may be to relax the transition probabilities so that the primitive can begin either at the start of the lead in, or just after it. Alternatively, a cyclic form of state transition restriction could replace the shifting routine. So far, the cyclic state transitions have not been successful, as it is too easy for very short primitives to develop, and be fitted trivially to the data.

Another problem has been that the primitives identified by the algorithm are not always unique. This is a standard clustering problem, with two sets being assigned to a single cluster, whilst neglecting a separate cluster. One way to approach this is to merge two sets together if they are very similar. To implement this, the primitives will be compared by taking the maximum of their convolutions, giving a similarity metric, and then if two are very similar, their average will replace one of the two, and the other will be randomised.

As the EM-algorithm is only guaranteed to converge to a local minimum, it is dependent sometimes strongly on the initialisations of the parameters. Thus far, they are randomly initialised, within a bound that is similar to the range of the data. It may be better to use the main ICA components as initialisations, as these do contain fragments of the primitives in question, and so may speed up the convergence of the EM algorithm.

5 Future Targets

One problem with the current implementation of the model is its speed of execution. It may be necessary to look at ways to speed up the model if it is to be used on large data sets. This may involve a slightly different optimisation technique, or simply writing the code in a more efficient manner.

The precise details of how the fHMM model is constrained to relate more to the Piano Model may change to make the model more reliable. A more generalised state transition structure may relieve the need to shift the primitives back and forth to get them to fit nicely.

In general, the current target is to develop a technique that can extract primitives from natural data (handwriting data), using the Piano Model as a basis. These primitives would be very useful both for modelling handwriting in a generative manner, and to study skill learning in humans, and whether these primitives adapt, or are constant across tasks. Discovery of these primitives would be very important for many different fields of research, from handwriting recognition to robotics.

Once a suitable modelling environment has been developed, then natural data can be used. This will be collected from a digitisation tablet, with a layer of carbon paper on top. This will enable natural visual and proprioceptive feedback for the subjects whilst writing. The data collected will be in 5 dimensions. 2 for position, 1 for pressure, 2 for angle or tilt of pen. Several different types of adaptation experiment could be performed in this manner. Here are some examples:

- Learning new characters: Does this affect the shape of the motor primitives, or just their timing?

- Adaptation in the young: When children are learning to handwrite, do the primitives adapt to a particular character set, or maybe style of handwriting?
- Character set vs. Handwriting style: For people who can write fluently in more than one character set (eg. Latin and Arabic), do the primitives remain constant over character sets, or are they influenced by the particular task at hand?

5.1 Further Points to Explore

Here are some sketches of ideas for further exploration.

There has been evidence that the size of the characters is independent to the style of the character, and adaptation studies have looked at micrographia in Parkinson's patients. Possibly linear scaling effects are not connected to motor primitives, rather than a scaling applied by output gating mechanism (with a maximum scaling factor 2.5, as demonstrated in adaptation studies).

Recognition based not only on visual input, but associated motor planning. Motor knowledge used as an 'informed padding' for gaps in input. Can be generalised to all recognition tasks? Easier to recognise what we 'understand'? Approach focusses on important, functional aspects of input, acting as a 'function filter'?

Internal forward models of motor control allow the subject to simulate motor commands and their consequences without physically performing them. This could be likened to an internal visualisation of a movement. Can internal simulated completion of partial characters improve on pixel-based recognition? Could internal visualisation of character + drawing strategy help to separate character from obscuring marks / missing sections of character?

Drawing from memory. Can a convincing 'sketch' be produced by such a program? Do handwriting styles influence sketching ability? Can a training set improve sketching ability?

Are scribbling, sketching, and handwriting linked? Can scribbles be regarded as motor / character primitives?

Does learning a new character set affect old primitives?

6 Aims and Possible Outcomes

The aim of this project is to help us understand how we learn to handwrite, and more generally, learn motor tasks. This understanding goes hand-in-hand with possible improvements to robotic control strategies, which would also be a valuable outcome.

A motor primitive theory which is more faithful to the biological evidence is aimed for. There are many studies looking at precise motor timing in the cerebellum. These ideas may well be complimentary to ours, where the precise trajectories are separate entities from the timing of their activations.

Because paper and pen trajectories are easily modelled in a virtual environment, it will be a logical step to incorporate these ideas into a virtual model of an agent with a virtual drawing pad, through which it may be possible to simulate the process of copying and internal modelling that presumably goes with natural handwriting acquisition. If successful, this would provide an extremely interesting environment to explore different methods of teaching handwriting, and how different feedback techniques may influence learning.

The method of inferring where independent, overlaid natural signals lie in a time series is clearly not limited to handwriting. Most human generated signals share this property of consisting of several, semi-independent, super-imposed motifs, or primitives. Particularly speech, where very complex, rapidly changing waveforms are created by a slowly changing underlying structure (the mouth shape). There will also be movement primitives associated with the mouth, which could help with understanding the speech waveform.

The robotics community have been trying to define what a useful motor primitive is for some years now. Clearly, this is because having primitives is as useful for robotic movement planning and execution, as it is for human movements. Using statistical techniques to find optimal primitives may well be a solution to this problem.

References

A. Alissandrakis, C.L. Hehaniv, and K. Dautenhahn. Do as i do: Correspondences across different robotic embodiments. In *Proc. Fifth German Workshop on Artificial Life (GWAL5)*, 2002.

H. S. M. Beigi. An overview of handwriting recognition. In *1st Annual conference on technological advancements in developing countries*, pages 30–46, Columbia University, New York, July 1993.

E. Bizzi, S.F. Giszter, E. Loeb, F.A. Mussa-Ivaldi, and P. Saltiel. Modular organization of motor behavior in the frog's spinal cord. *Trends in Neurosciences*, 18(10):442–446, 1995.

E. Bizzi, A. d'Avella, P. Saltiel, and M. Trensch. Modular organization of spinal motor systems. *The Neuroscientist*, 8(5):437–442, 2002.

G. Caithness, R. Osu, P. Bays, H. Chase, J. Klassen, M. Kawato, D. M. Wolpert, and J. R. Flanagan. Failure to consolidate the consolidation theory of learning for sensorimotor adaptation tasks. *Journal of Neuroscience*, 24(40):8662–8671, 2004.

B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs, 2003.

S.J. Cho and J.H. Kim. Bayesian network modelling of hangul characters for on-line handwriting recognition. In *ICDAR2003 IEEE*, 2003.

A. d'Avella and E. Bizzi. Shared and specific muscle synergies in natural motor behaviors. *PNAS*, 102(8):3076–3081, 2005.

A. d'Avella and E. Bizzi. Low dimensionality of supraspinally induced force fields. *Proc. Natl. Acad. Sci. USA*, 96:7711–7714, 1998.

A. d'Avella, P. Saltiel, and E. Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308, 2003.

P. R. Davidson and D. M. Wolpert. Motor learning and prediction in a variable environment. *Curr. Opinion in Neurobiology*, 13:1–6, 2003.

P. R. Davidson and D. M. Wolpert. Scaling down motor memories: de-adaptation after motor learning. *Neuroscience Letters*, 370:102–107, 2004a.

P.R. Davidson and D.M. Wolpert. Internal models undelying grasp can be additively combined. *Exp. Brain Res.*, 155:334–340, 2004b.

M. Dennis, K. Edelstein, R. Hetherington, K. Copeland, J. Frederick, S. E. Blaser, L.A. Kramer, J.M. Drake, M. Brandt, and J. M. Fletcher. Neurobiology of perceptual and motor timing in children with spina bifida in relation to cerebellar volume. *Brain*, 2004.

J.R. Flanagan, E. Nakano, H. Imamizu, R. Osu, T. Yoshioka, and M. Kawato. Composition and decomposition of internal models in motor learning under altered kinematic and dynamic environments. *The Journal of Neuroscience*, 19, 1999.

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

V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119:593–609, 1996.

Z. Ghahramani and M.I. Jordan. Factorial hidden markov models. *Machine Learning*, 1997.

A. Hamilton, K. Jones K, and D. M. Wolpert. The scaling of motor noise with muscle size and motor unit number. *Experimental Brain Research*, 157: 417–430, 2004.

A. Hyvärinen. Survey on independent component analysis, 1999.

A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor lanscapes for motor primitives. *MIT Press*, 15, 2003.

A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In S. Becker, Thrun, S., Obermayer, and K., editors, *Advances in Neural Information Processing Systems 15 (NIPS2002)*, pages 1547–1554, 2002.

O. D. Kharraz-Tavakol, T. Eggert, N. Mai, and A. Straube. Learning to write letters: transfer in automated movements indicates modularity of motor programs in human subjects. *Neurosci Lett.*, 17(1-2):33–36, 2000.

K. P. Körding, S. Ku, and D. M. Wolpert. Bayesian integration in force estimation. *Journal of Neurophysiology*, 92:3161–3165, 2004.

C. Marquardt, W. Gentz, and N. Mai. Visual control of automated handwriting movements. *Exp. Brain Research*, 128:224–228, 1999.

M.J. Matarić. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. Technical report, USC, 2000.

M.J. Matarić. Primitives-based humanoid control and imitation. Technical report, DARPA MARS project, 2004.

D.V. Meegan, R.N. Aslin, and R. A. Jacobs. Motor timing learned without motor training. *Nature Neuroscience*, 3(9):860–862, 2000.

R. G. J. Meulenbroek and A. W. A. Van Gemmert. Advances in the study of drawing and handwriting. *Human movement science*, 22:131–220, 2003.

F.A. Mussa-Ivaldi and E. Bizzi. Motor learning through the combination of primitives. *Phil. Trans. R. Soc. Lond.*, 355:1755–1769, 2000.

M.K. Omar, K. Chen, and M. Hasegawa-Johnson. An evaluation of using mutual information for selection of acoustic features represenation of phonemes for speech recognition. 2002.

C. Padoa-Schioppa, C. S. R. Li, and E. Bizzi. Neuronal correlates of kinematics-to-dynamics transformation in the supplementary motor area. *Neuron*, 36: 751–765, November 2002.

C. Padoa-Schioppa, C.S.R. Li, and E. Bizzi. Neuronal activity in the supplementary motor area of monkeys adapting to a new environment. *J. Neurophysiol*, 91:449–473, 2004.

V.B. Penhume, R.J. Zatorre, and A.C. Evans. Cerebellar contributions to motor timing: A pet study of auditory and visual rythm reproduction. *Journal of Cognative Neuroscience*, 10(6):752–765, 1998.

R.P.N. Rao, A.P. Shon, and A.N. Meltzoff. A bayesian model of imitation in infants and robots. 2004.

S. Roweis and A. Alwan. Towards articulatory speech recognition: Learning smooth maps to recover articulator information. In *Proc. Eurospeech97*, pages 1227–1230, 1997.

S.P. Schaal, J. Nakanishi, and A. Ijspeert. Learning movement primitives, 2004.

E. Todorov and M.I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

R. J. van Beers, P Haggard, and D. M. Wolpert. The role of exucution noise in movement variability. *J. Neurophysiol.*, 91:1050–1063, 2004.

A. Vinciarelli and S. Bengio. Offline cursive word recognition using continuous density hidden markov models trained with pca or ica features. 2001.

Y. Wada, Y. Koike, E. Vatikiotis-Bateson, and M. Kawato. A computational model for cursive handwriting based on the minimisation principle.

A. G. Witney and D. M. Wolpert. Spatial representation of predictive motor learning. *J. Neurophysiol.*, 89:1837–1843, 2003.

D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.

D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan. Perspectives and problems in motor learning. *TRENDS in Cog. Sci.*, 5(11):487–494, 2001.