Extracting Motion Primitives from Natural Handwriting Data

Ben H Williams, Marc Toussaint, and Amos J Storkey

Institute of Adaptive and Neural Computation
University of Edinburgh
School of Informatics
5 Forrest Hill, EH1 2QL Edinburgh, UK
ben.williams@ed.ac.uk

Abstract. For the past 10 years it has become clear 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. This paper shows that it is possible to use a factorial hidden Markov model to infer primitives in handwriting data. These primitives are not predefined in terms of location of occurrence within the handwriting, and they are not limited or defined by a particular character set. Also, the variation in the data can to a large extent be explained by timing variation in the triggering of the primitives. Once an appropriate set of primitives has been inferred, the characters can be represented as a set of timings of primitive activations, along with variances, giving a very compact representation of the character. Separating the motor system into a motor primitive part, and a timing control gives us a possible insight into how we might create scribbles on paper.

1 Introduction

As with all planning tasks, there is a debate as to the degree of pre-planning in movement control. Humans and animals find solutions to movement tasks that are both repeatable to some extent across trials and circumstances, and subjects [17, 16, 12]. Despite this repeatability, we are also very adaptable to new tasks, and can quickly adjust learnt movements to cope with new environments [5]. There must therefore be some compromise between preprogrammed movement, and instantaneous movement planning in biological organisms.

Evidence suggests that once a particular movement has commenced, it cannot be unexpectedly switched off. Rather, to quickly modify a movement trajectory, the movements are superimposed [6]. This suggests that there is a subroutine type of movement activation, where the subroutines are not quickly adaptable, but their individual activation can be a fast and globally relevant process. This modularisation of movement control could provide a good compromise between movement pre-planning and online error correction. These subroutines of motion will be referred to as motor primitives. Strong biological evidence exists to suggest that these primitives exist, with motor primitives first being conclusively found in frogs [2, 3, 4] where stimulation of a single spinal motor afferent triggered a complete sweeping movement of the frog's leg. For a review of modularisation of motor control in the spine, see [1].

There have been many studies on recording the dynamics of all aspects of natural human movement. People have tried to infer motor primitive type sub-blocks from the

sequences of movement [18, 12, 9, 15, 11]. Most of these attempts have pre-partitioned the sequences into movement sub-blocks, then extracted principal components. This means either considering the entire movement as a primitive, in the domain of handwriting, the entire character, or finding segmentation points, such as points of highest curvature, or maximum torque. The disadvantage of this method is that strong assumptions must be made about the partitioning of the data, and the duration of the primitives. Rather than pre-partitioning the data into sub-blocks, it would be better to allow this partitioning to be inferred.

We have used a probabilistic framework to define a generative model for primitive activation. The parameters for the generative model can be learnt using Expectation-Maximisation. This method provides a way to infer primitives without any pre-partitioning of the data. We hypothesise that motor primitives make up the movement commands being sent to the hand and arm muscles during handwriting. We record a vector describing the position, pressure, and tilt of the pen over time. The dynamics of this vector over time will reflect the hand motion and therefore contain projections of motor primitives.

In Section 2, we describe the model from motivation to modelling details. Sections 3 and 4 present the data, and the results obtained so far, showing typical primitives obtained, the separation of primitive from primitive timing, using primitives inferred from one dataset to model a different set of data, and some examples of the generative model running without specific timing information. Section 5 discusses the results, the partitioning of the model into primitive and timing parts, and the possible biological parallels with structures in the mammalian brain.

2 The Model

Unlike many previous studies which analysed motor primitives directly from given data, we base our approach on a generative model of motion. With probabilistic inference methods, we infer the primitives inherent in given data. We assume that the activation of motor primitives can be overlapping, and that they do not have uniform fixed length. The primitives are therefore the output vocabulary, which may either be inherited, or learnt over long time-scales. We assume that the primitives have little or no dependence on each other. The independence of one primitive from another, and post-activation persistence of the motor primitives, give rise to a model that is similar in nature to that of a piano. (See Figure 1)

2.1 A Simple Generative Model: The Piano Model

To formalise the model in a generative way, the output of the system Y at time t is defined as

$$Y(t) = \sum_{m,n} \alpha_{mn} W_m(t - \tau_{mn}) \,, \tag{1}$$

where $W_m(t)$ are the primitives, and τ_{mn} represents the time of the nth activation of primitive m, and α_{mn} defines the activation strengths of the primitives.

In this definition, m enumerates the primitives, whilst n enumerates the occurrence of the primitive within the sample window, at time τ_{mn}. We have called this model 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
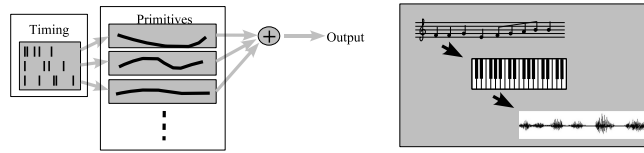
Fig. 1. The Piano Model. The model is segmented into a timing part and a movement subroutine, or primitive part. The timing information is encoded in spike positions, possibly in a similar way to how biological neurons may encode the movement. The movements are encoded as multidimensional muscle activation synergies in biology. Here they have been simplified to one-dimensional signals. The analogy with written music being translated into auditory music is also shown.
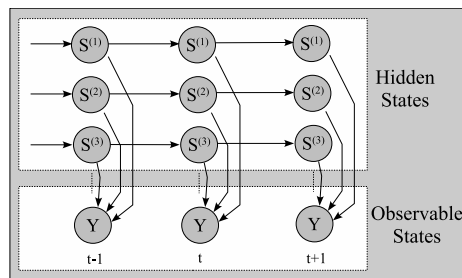

Fig. 2. Graphical representation of a Factorial Hidden Markov Model, showing the independence of the separate Markov chains. Although the observable output is dependent upon the state of the entire system, the internal states evolve with no interdependencies. S_t^m denotes the hidden state vector at time t, in factor m.

piece of music. The keys on the piano produce time extended clips 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, in our motor primitive model, a central controller does not control the precise movements but rather the timings of particular temporal movement sequences. Figure 1 shows a diagram to illustrate the Piano Model.

The Piano Model neglects noise effects and learning the parameters of the model is better realised in a probabilistic framework. Assuming discrete time steps, an appropriate modelling framework is that of Factorial Hidden Markov Models (fHMMs). These are the same as standard HMMs, but with multiple, parallel and independent hidden state chains, as seen in Figure 2.

2.2 The Factorial Hidden Markov Model

A graphical model of the fHMM can be seen in Figure 2. At each time step, the observable output Y_t, a vector of dimension D, is dependent on M hidden variables

---

[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.

$S_t^{(1)}, .., S_t^m$. The output is a multivariate Gaussian, such that

$$Y_t \sim \mathcal{N}(\mu_t, C) , \tag{2}$$

where C is a $D \times D$ parameter matrix of output covariance, and

$$\mu_t = \sum_{m=1}^{M} W^m S_t^m . \tag{3}$$

is the D-dimensional output mean at time t. W^m is a $D \times K$ parameter matrix giving the output means for each factor m, such that the output mean μ_t is a linear combination of its columns weighted with the hidden state activations.

Each of the M hidden variables can be in K different states. In equation (3) this is encoded in the K-dimensional state vector S_t^m using a 1-in-K code, i.e., $S_{t,i}^m = 1$ if the m-th factor is in state i and zero otherwise. This allows us to write expectations of the hidden states as $\langle S_t^m \rangle$. Each latent factor is a Markov chain defined by the state transition probabilities and the initial state distribution as

$$P(S_1^m\!=\!i) = \pi_i^m , \qquad P(S_t^m\!=\!i \,|\, S_{t-1}^m\!=\!j) = P_{i,j}^m , \tag{4}$$

where π^m is a K-dimensional parameter vector giving the initial hidden state distribution, and P^m is a $K \times K$ parameter matrix denoting the state transition probabilities. As can be seen in Figure 2, each factor is independent. This means that the joint probability distribution can be factorised as

$$P(\{Y_t, S_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{5}$$

$$= \prod_{m=1}^{M} \pi^m P(Y_1|S_1) \prod_{t=2}^{T} \prod_{m=1}^{M} P^m P(Y_t|S_t) . \tag{6}$$

The fHMM model was based upon that described in [10], which provides arguments for using a distributed state representation as is used here, and discusses the model in further detail.

To use the fHMM framework as a probabilistic implementation of the Piano Model, we must attribute each hidden Markov chain, or factor, to one primitive. The observables, being real-valued output vectors describing pen position derivatives are modelled by the multivariate output Gaussian distribution, dependent upon the hidden state values at time t. The model has M primitives, with each primitive having K states. This is similar to the Piano Model, given some extra constraints.

2.3 Constraints

The Piano Model differs from the fHMM model by allowing the primitives to be inactive, giving a zero output contribution. In fact, it is a tacit assumption that the primitive activation is fairly sparse, making the periods of inactivity important to the model. In the fHMM model, the output is always a linear combination of all the factors, thus a major constraint imposed was that state 0 for all Markov chains should contribute towards a zero output mean (i.e., to not contribute, as the data mean is zero).
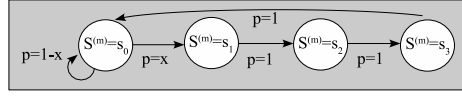
Fig. 3. State change probabilities. This shows how the state change probabilities were constrained so that the individual Markov chains can correspond to a time extended primitive.

In the Piano Model, each primitive should keep its shape. This means that the possible hidden state transitions in the fHMM model needed to be constrained. When a particular primitive was triggered, the hidden state vector for that factor changes from state 0 to state 1, then is constrained to progress monotonically through the states until the last state is reached,and returns to state 0. These state change restrictions can be seen graphically in Figure 3.

2.4 Learning the Model

Given the fully parameterised modelling framework, learning of the parameters can be done using an Expectation-Maximisation (EM) method. The structured variational approximation was chosen for the E-step inference. For more details on the various arguments for and against this choice, refer to [10]. With the structured variational approximation, the inference in the fHMM is split up into M separate Hidden Markov Models, with single hidden state chains, with each HMM contributing a learnt proportion towards the output. With separate HMMs, the normal Baum-Welch Forward-Backward algorithm can be used to infer the hidden state expectations [7]. The only addition necessary is a responsibility factor h_t^m, which models the amount that the mth HMM contributes towards the output. h_t^m takes the place of the observation likelihood in a standard HMM. See (7) for the details of calculating h_t. The M-step updates the parameters W^m, π^m, P^m, and C. The update equations are in Appendix A.

3 Implementation

Handwriting data were gathered using an INTUOS 3 WACOM digitisation tablet `http://www.wacom.com/productinfo/9x12.cfm`. This provided 5 dimensional data at 200Hz. The dimensions of the data were x-position, y-position, pen tip pressure, pen tilt angle, and pen orientation (0-360°). The normalised first differential of the data was used, so that the data mean was close to zero, providing the requirements for the zero state assumption in the model constraints (see section 2.1). The data collected were separated into samples, or characters, for processing purposes, and then the parameters were fitted to the data using our algorithm. Once the parameters were learnt, the hidden state expectations $\langle S \rangle$ were finalised, and the pen space reconstruction of the data could be calculated, along with the primitive timing statistics.

To clarify the operation of our algorithm, and the iterative nature of the EM inference, here are the pseudo-code and parameter settings.

Constants T=times of each sample, N=number of samples, K=max primitive length, M=number of primitives, D=dimension of data.

Initialisations The primitives W^m are initialized with a zero mean Gaussian distribution of the same variance as the data. The transition probabilities π^m and P^m are set as
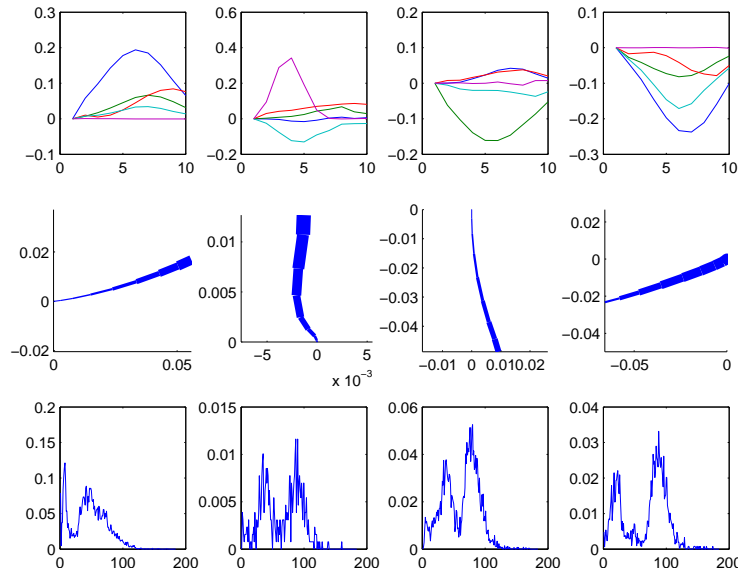
Fig. 4. A sample of 4 typical primitives taken from a set of 36, inferred from the *'g'* dataset. At the top, the 5 dimensional velocity space primitives are shown. The maximum length of these primitives was 10 samples. The centre row shows a pen-space reconstruction of each primitive, with the thickness of the line representing the pressure of the pen tip. The starting point of the reconstruction is at $(0, 0)$. The bottom row shows the distribution of the onset of the primitive over all the character samples.

defined in Section 2.3 with primitive onset probability N/T, giving a prior expectation of each primitive to be used once in each character. The output covariance C is taken directly from the covariance of the data.

loop (EM loop)

E-step: initialize $\langle S_t^m \rangle$ to $P(S_t^m\!=\!0) = 1$ for all t, m.

    loop

        compute h_t^m from equation (7).

        $\forall m$: forward-backward algorithm gives $\langle S_t^m \rangle$ using h_t^m as obs. likelihood.

    until expectations not changing significantly, or max 20 iterations

M-step: update the parameters as in (10)

until primitives don't changing significantly, or max 50 iterations.

A large dataset of over 1000 samples of the character *'g'* was created, and a smaller dataset of over 100 samples of the character *'m'* was also used, for speed issues. To examine whether primitives can be disassociated from any particular character set, a dataset of over 100 samples of scribbling was also created.

4 Results

Typical primitives. From the *'g'* samples, it was found that the primitives tend to model similar parts of the character across most, but not all samples. In Figure 4, we can see
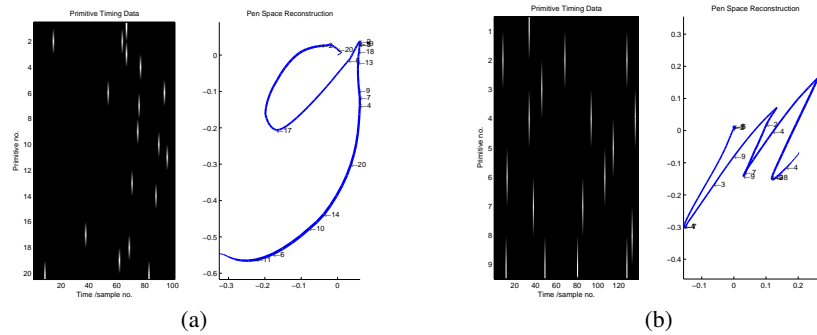
Fig. 5. Two examples of primitive timing codes. In both (a) and (b), the timing information is shown on the left, and the reproduction of the sample on the right, with the onset of each primitive marked with an arrow. In (a), 20 primitives of length 40 time steps were used to model the *'g'* dataset, of over 1000 characters. In (b), 9 primitives of length 30 were used to model the *'m'* dataset, of over 100 characters.
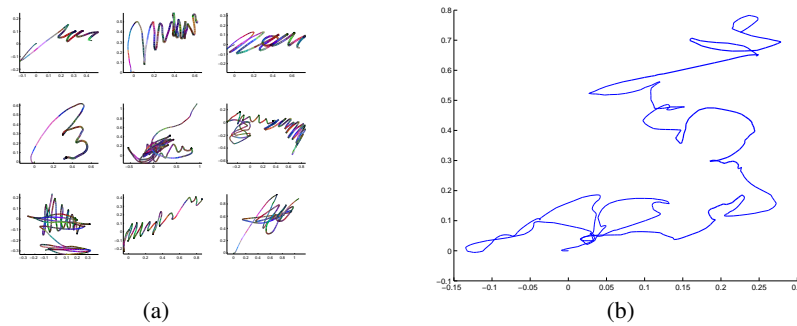

Fig. 6. A reconstruction of some samples from the scribble dataset is shown in (a). This was a set of unconstrained scribbles. The primitives are coloured differently, to show different activation areas. (b) shows a sample generated using primitives from this dataset. Starting point of reconstruction is $(0, 0)$.

a sample of primitives in velocity and their pen-space reconstructions along with their timing distributions, from a set used to model the *'g'* samples dataset.

Compact encoding with onset timing. To show the way different primitives are activated at different times, Figure 5 shows the timing information, which efficiently encodes the reproduction of the sample shown.

The distributions of activation of a primitive over the different samples tends to be bell shaped, either uni- or bimodal, meaning that the primitives have a preference for a particular part of the character, with the variation in timing accounting for the variations across characters, at least to a certain extent.

Primitives from scribbling to generate characters. To explore whether the algorithm was picking up features of a particular character set, or more generalised motor primitives, a dataset of scribbles were used. These consisted of unconstrained scribbling,
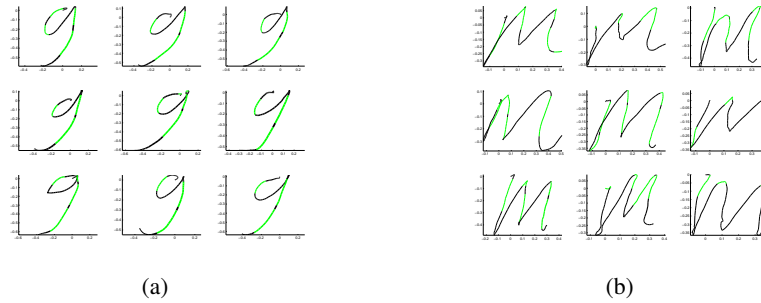
Fig. 7. A reconstruction of some samples, using primitives from the scribble set. One of the 36 primitives is highlighted. (a) shows characters from the *'g'* dataset, (b) shows characters from the *'m'* dataset.
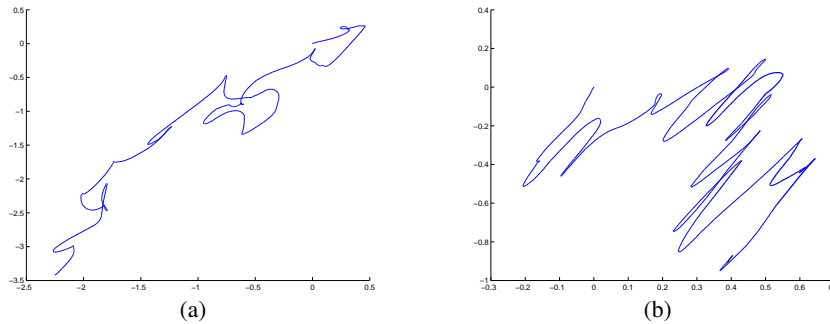

Fig. 8. Two samples generated using primitives without specific timing information. (a) was generated using primitives inferred from the *'g'* dataset, (b) was generated using primitives from the *'m'* dataset. Starting point of both reconstructions is $(0, 0)$.

without any character set goal objectives. Our algorithm was run on this dataset, and in Figure 6(a), we can see a reconstruction of scribble samples.

Using the primitives from this dataset, it was possible to represent the other dataset. In Figure 7, we can see examples of the character *'g'*, and *'m'* being drawn with scribble primitives.

Random expression of primitives with different timing statistics. To further explore what aspects of the character set are captured by the primitives, it is possible to simply run the model generatively, using the inferred parameters. In Figure 8 we see two samples generated using the sets of primitives and other associated parameters inferred from the *'g'* and *'m'* datasets. In both samples, we see aspects of the character, but no clear examples of a character drawn perfectly. This is partly because the primitives are assumed to be generatively independent, meaning the primitive that models the start of the character cannot convey information about its state to the primitive modelling the subsequent part. The primitives, although capturing an 'aspect' of the character, lack the precise timing information that dictates how the character is drawn. This timing information can be seen in Figure 5.

The generative scribbling could be likened to absent-minded doodles, where we control a pen, and produce writing output, but without any constraints dictating what character we should draw. Indeed, it is impossible to tell whether or not the sample shown in Figure 6(b) is from a scribbled dataset drawn by a human, or a generative sample 'drawn' by the algorithm.

5 Discussion and Conclusions

It is possible to model characters with primitives using a Factorial Hidden Markov Model that does not pre-specify the timing of the primitives. The primitives, although not constrained to be active for the whole character, or to be active between pre-defined segmentation points in the character, tend to model particular areas of the character. The variation of the character within the dataset is modelled partly by variations in primitive timing, and partly by different primitives being active.

The primitives do capture an aspect of the character set they were trained upon, although, they do not contain the precise timing information required for the reproduction of a character accurately. This timing information is a compact representation of the character, as it simply encodes the onset timing of each primitive. Without this timing information, the generative output of the model acquires the aspect of scribbling. It is possible that when humans doodle absent-mindedly, it is a lack of timing information that is causing the scribble type output from the motor system.

This model lends itself towards breaking up of the motor system into 2 main modules. Firstly, the Primitive Module, encompassing parameters such as likelihood of a particular motion, likely amplitude of a particular motion, with implementation capabilities (control of muscles) of segmented, independent motions. Secondly, the Timing Module, encoding the overall motor 'strategy', and allowing compensation through feedback, and online error correction to the timing of the motions, rather than the actual motions themselves. In the brain, this segmentation could be paralleled by the motor cortex and lower motor systems encoding the actual motor commands, while the cerebellum encodes the motor command timing. There is strong evidence to suggest that the cerebellum is involved not only with motor timing, but perception of timed events [13, 8, 14]. As the lack of a cerebellum does not completely impair movement, this implies that muscle control is located elsewhere, such as in the motor cortex.

A Appendix

In the E-step, the responsibility factor, h_t was calculated using a residual error,

$$h_t^{(m)new} = \exp\{W^{(m)T}C^{-1}\tilde{Y}_t^{(m)} - \frac{1}{2}\triangle^{(m)}\} \tag{7}$$

$$\triangle^{(m)} \equiv \text{diag}(W^{(m)T}C^{-1}W^{(m)}) \tag{8}$$

$$\tilde{Y}_t^{(m)} \equiv Y_t - \sum_{l\neq m}^{M} W^{(l)}\langle S_t^{(l)}\rangle \tag{9}$$

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

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

$$W \leftarrow (\sum_{t=1}^{T} Y_t \langle S_t^T \rangle)(\sum_{t=1}^{T} \langle S_t S_t^T \rangle)^{\dagger} \tag{10}$$

$$\pi^{(m)} \leftarrow \langle S_1^{(m)} \rangle \qquad P_{i,j}^{(m)} \leftarrow \frac{\sum_{t=2}^{T} \langle S_{t,i}^{(m)} S_{t-1,j}^{(m)} \rangle}{\sum_{t=2}^{T} \langle S_{t-1,j}^{(m)} \rangle} \tag{11}$$

$$C \leftarrow \frac{1}{T} \sum_{t=1}^{T} Y_t Y_t^T - \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle Y_t^T \tag{12}$$

$\langle S_t \rangle$ is the expected value of the hidden states at time t. $\dagger$ denotes *pseudo-inverse*.

References

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

[2] 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.

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

[4] 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.

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

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

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *J. R. Statist. Soc. B*, 39:1–38, 1977.

[8] 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.

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

[10] Z. Ghahramani and M.I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.

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

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

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

[14] 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.

[15] S.P. Schaal, J. Nakanishi, and A. Ijspeert. Learning movement primitives. In *ISRR2003*, 2004.

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

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

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