A Dual-Pathway Neural Network Model of Control Relinquishment in Motor Skill Learning

Ashish Gupta & David C. Noelle
Department of Electrical Engineering and Computer Science
Vanderbilt University
{ashish.gupta, david.noelle}@vanderbilt.edu

Abstract

Cognitive psychologists have long recognized that the acquisition of a motor skill involves a transition from attention-demanding controlled processing to more fluent automatic processing. Neuroscientific studies suggest that controlled and automatic processing rely on two largely distinct neural pathways. The controlled pathway, which includes the prefrontal cortex, is seen as acquiring declarative representations of skills. In comparison, the automatic pathway is thought to develop procedural representations. Automaticity in motor skill learning involves a reduction in dependence on frontal systems and an increased reliance on the automatic pathway. In this paper, we propose a biologically plausible computational model of motor skill automaticity. This model offers a dual-pathway neurocomputational account of the translation of declarative knowledge into procedural knowledge during motor learning. In support of the model, we review some previously reported human experimental results involving the learning of a sequential key pressing task, and we demonstrate, through simulation, how the model provides a parsimonious explanation for these results.

1 Introduction

Learned motor skills are central to most human activities. Learning plays a critical role in our abilities to walk, talk, cook, type, and play games like ping-pong, for example. The importance of learning in human motor performance has led many robotics researchers to examine machine learning approaches to motor control. We currently lack a clear and complete computational account of how humans acquire motor skills, however. In hopes of addressing this deficit, this paper offers some insights into the neurocomputational structure of motor skill learning.

One of the central findings of cognitive research into skill learning involves the process of *automaticity*, through which fluency at a skill is improved by gradually shifting from a declarative representation of the task to a more procedural representation [Anderson, 1981]. A growing body of neuroscientific evidence suggests that declarative and procedural processes are implemented by two largely separate neural networks in the brain [Bapi *et al.*, 2000; Hikosaka *et al.*, 2002; Wolpert *et al.*, 2001]. Both networks have the capability to individually acquire motor skills, but they typically coordinate with each other during learning. The *controlled pathway* includes dorsolateral prefrontal cortex (DLPFC), anterior cingulate cortex (ACC), anterior parts of the cerebellum, anterior parts of the basal ganglia, and the pre-supplementary motor area (preSMA). This pathway is seen as acquiring representations of motor skills that are primarily declarative. Declarative representations are formed very quickly, and they guide skill execution during the early stages of learning. The second network, which we call the *automatic pathway*, includes areas like the supplementary motor area (SMA), primary motor cortex, lateral parts of the cerebellum, and lateral parts of the basal ganglia. As a skill becomes well practiced, this network slowly encodes a procedural representation of the skill. With practice, the involvement of the frontal controlled pathway decreases, and the skill comes to be primarily executed by the automatic pathway. The modulation of frontal involvement is thought to be governed by a separate coordination mechanism, perhaps embodied in the preSMA and the ACC [Hikosaka *et al.*, 2002].

This paper addresses a key question concerning this process - are the information processing properties of these brain regions, as they are currently understood, sufficient to account for the behavioral shift in skill learning? To address this question, we have explored a neurocomputational model of motor skill learning that is based on the dual-pathway hypothesis. We report the results of simulation experiments involving a sequential key pressing task. In these simulations, keys are pressed using a two joint planar arm. The arm learns to trace out a sequence of trajectories such that the end effector successively moves from one key to the next in a trained sequential order.

In our model, the controlled pathway learns a declarative representation of the task: the key sequence. When executing the task, the prefrontal cortex (PFC) in the controlled pathway actively maintains an abstract representation of the next key to be pressed. This representation of the desired key, along with the current state of the arm, is then transformed by the network into an appropriate reaching trajectory toward that key. Once the current target key has been pressed, the PFC rapidly updates to encode the next key in the sequence, and the next reach is produced. Thus, the controlled pathway needs only to learn the sequence of keys during task learning, depending on a pre-trained *motor area* to translate the PFC representation of each target key into an appropriate reaching motion.

In contrast, the automatic pathway of our model learns the entire motor skill from scratch. It acquires a procedural representation of the skill by learning the entire motion trajectory needed for the complete sequence of key presses. This pathway learns more slowly than the controlled pathway, because much more detailed knowledge must be learned.

As the automatic pathway becomes proficient in executing the sequence, the involvement of the controlled pathway is withdrawn. In our model, this shift is driven by a *cognitive control* mechanism. This mechanism monitors performance error and modulates the weight given to the controlled pathway appropriately. When error is high, the contribution of the fast-learning controlled pathway is strengthened. As error falls, the contribution of the automatic pathway is allowed to dominate.

2 Background

2.1 Leabra

Our model uses the Leabra modeling framework [OReilly and Munakata, 2001]. Leabra offers a collection of integrated cognitive modeling formalisms that are grounded in known properties of cortical circuits but are sufficiently abstract to support the simulation of behaviors arising from large neural systems. It includes dendritic integration using a point neuron approximation, a firing rate model of neural coding, bidirectional excitation between cortical regions, fast feedforward and feedback inhibition, and a mechanism for synaptic plasticity. Of particular relevance to the current model are Leabra's synaptic learning rules and its lateral inhibition mechanism.

Leabra modifies the strength of synaptic connections in two ways. An error correction learning algorithm changes synaptic weights so as to improve network task performance. Unlike the backpropagation of error algorithm, however, Leabra's error correction scheme does not require the biologically implausible communication of error information backward across synapses. In addition to this error-correction mechanism, Leabra also incorporates a Hebbian correlational learning rule. This means that synaptic weights will continue to change even when task performance is essentially perfect.

The effects of inhibitory interneurons tend to be strong and fast in cortex. This allows inhibition to act in a regulatory role, mediating the positive feedback of bidirectional excitatory connections between brain regions. Simulation studies have shown that a combination of fast feedforward and feedback inhibition can produce a kind of "set-point dynamics", where the mean firing rate of cells in a given region remains relatively constant in the face of moderate changes to the mean strength of inputs. Leabra implements this dynamic using a k-Winners-Take-All (kWTA) inhibition function that quickly modulates the amount of pooled inhibition presented to a layer of simulated cortical neural units based on the layer's level of input activity. This results in a roughly constant number of units surpassing their firing threshold. In our model, this mechanism both encourages learning through the development of sparse neural representations and it helps to make neural network outputs interpretable.

2.2 Cognitive Control

Cognitive control involves the ability to behave according to rules or goals instead of responding in a reflexive manner. Prefrontal cortex (PFC) is critical for robust cognitive control [OReilly and Munakata, 2001]. Dense recurrent connectivity in PFC allows it to actively maintain information in firing rate patterns, and it is thought that this active maintenance of task relevant information is central to task directed behavior. Also, interactions with the midbrain allow the PFC to rapidly update its state, based on task requirements [Rougier *et al.*, 2005]. We have incorporated both active maintenance and rapid updating in the PFC component of our model.

The strength of cognitive control can be modulated based on the task requirements. Botvinick et al. [2001] proposed that the anterior cingulate cortex (ACC) monitors the amount of conflict between parallel neural pathways and strengthens cognitive control when conflict is high. In our model, performance error is seen as a sign of conflict. Thus, we modulate the strength of cognitive control between trials, in proportion to the amount of error experienced on previous trials.

2.3 Motor Skill Learning

A wide variety of sequential key pressing tasks have been used to investigate human motor skill learning [Bapi *et al.*, 2000; Hikosaka *et al.*, 2002; Rand *et al.*, 2001], and a number of interesting findings have resulted. There is a period of rapid improvement in performance during the early stages of training. During this stage, learning is effector independent (e.g., switching hands does not substantially degrade performance). Further, interfering with the frontal systems involved in the controlled pathway during this period seriously disrupts performance. Interfering with the automatic pathway, however, does not affect performance during this early period.

After extensive training, the execution of the skill becomes more automatized. The skill becomes relatively effector dependent. Also, performance remains robust if the controlled pathway is disrupted.

Additionally, studies of *choking under pressure* have suggested that errors in performance in the face of stress may have different causes early and late in learning [Beilock *et al.*, 2004]. Early in learning, when the controlled pathway dominates, errors may arise due to a failure to engage cognitive control systems. With a well practiced skill, however, degraded performance may be due to the excessive exertion of unnecessary cognitive control. These results are also reflected in our model.

2.4 Previous Computational Models

Ours is certainly not the first computational model of motor skill learning. For example, Wolpert et al [1998; 2001] proposed MOSAIC, a model for sensorimotor control. MOSAIC consists of multiple modules, where each module consists of a pair of forward and inverse models. While MOSAIC has many strengths, it does not address the issue of representational change in skill learning. There is no mechanism for early declarative representations, making MOSAIC somewhat analogous to our automatic pathway.

The automatic pathway in our model is largely based on the previous Leabra networks of Gupta and Noelle [2005b; 2005a]. These networks were used to explore the neurocomputational principles underlying skill savings and the transfer of knowledge from one skill to another.

Nakahara et al. [2001] proposed a skill learning model that is very similar in general architecture to our own. There are a number of important differences between our models, however. First, their model does not focus on the question of differences in declarative and procedural represntations of the skill. Instead, the early dominance of the controlled pathway is driven by differential learning rates, with the controlled pathway made to learn faster. In contrast, our model is novel in showing that declarative and procedural representations can naturally emerge from neural encodings, and this difference in encoding easily explains the difference in the speeds of learning. Also, unlike our model, the Nakahara model does not include a mechanism for dynamically adjusting cognitive control — the relative contribution of the controlled pathway. While the previous model does include a "coordinator", the function of this mechanism is not the same. Hence, in their model, once a skill has been automatized, its controlled execution based on task demands is not possible. This is inconsistent with the behavioral observations. Lastly, our model critically depends on the active maintenance of target key representations in the PFC and the rapid updating of these representations as keys are pressed, while the Nakahara model incorporates no such mechanism.

The biggest point of difference between our model and the previous models of skill learning is that our model grounds the previous more-abstract theories in well established neurocomputational mechanisms. In other words, our model demonstrates that the declarative/procedural translation theory fits naturally with Leabra neurocomputational primitives, and our instantiation takes a step toward the generation of quantitative predictions for the model.

3 Tasks

We have used our model to simulate the learning of key pressing motor sequences. Our model controls a simulated 2-joint planar arm which moves over a 9-key keyboard, as show in in Figure 1. The state of the arm at any point in time is represented by the vector (q1, q2), where q1 and q2 are the two joint angles. The joint angles range between 0° and 120°. Movements are to be generated in such a way that the end effector follows a straight line trajectory from the position of the previous key to the position of the next key in the sequence. The arm starts over of the bottom-left key. The motion trajectory is discretized at equidistant time intervals, and hence, any trajectory is represented as a sequence of arm states over the successive time steps. During training, the arm is essentially guided along the desired trajectory, with differences between the motor output of the arm controller and the configuration of the arm, as specified by the guide, acting as a measure of error to drive synaptic weight change.

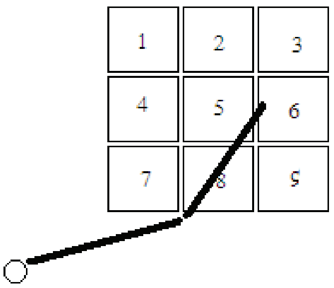

Figure 1: A two joint planar arm and a keyboard. The state of the arm at any point in time is given by the vector of joint angles (q1, q2). The arm produces motion trajectories such that its end effector moves from one key to the next, in sequence.

4 The Network

Figure 2 shows the Leabra network used for our simulations. The Sensory_Input layer provides the current state of the arm as input to the network and the Motor_Output layer is to produce the desired arm state for the next time step. Each joint angle is encoded over a pool of 15 neural units. Each of the 15 units has a preferred angle, ranging from -10° to 130°, in 10° increments. To encode a given joint angle, the closest unit with regard to preference, as well as its two neighbors, are set to their maximal firing rates. Similarly, patterns of activity over each row of 15 units in the Motor_Output are decoded by identifying the preferred angle of the unit in the middle of the three adjacent units that are all active. Other patterns of activity in the Motor_Output layer are considered to be ill-formed. With each joint angle encoded over 15 units in this way, the complete arm configuration is encoded over 30 units.

The network is composed of two pathways: the controlled pathway on the left and the automatic pathway on the right. In the automatic pathway, the Sensory_Input layer influences the Motor_Output layer via the Automatic_Path layer. This is similar to the network used by Gupta and Noelle [2005a; 2005b], with one addition. A contextual hidden layer has been added to this pathway, which provides a copy of the Automatic_Path layer activity at the previous time step as input to the Automatic_Path layer during the next time step. Connection weights from the Automatic_Path layer to the Motor_Output are not allowed to exceed 50% of the maximum weight value allowed by Leabra (implemented by setting the relative weight scaling parameter to 0.5). This restriction allows the controlled pathway to strongly dominate over the automatic pathway by strengthening the controlled pathway's influence on the Motor_Output layer beyond what is possible for the automatic pathway. This dominance occurs when cognitive control is strong. When cognitive control is weak, however, the automatic pathway weights can still be strong enough to drive appropriate outputs.

In the controlled pathway, the Sensory_Input layer provides input to the PFC layer. This layer generates a declarative representation of the key sequence, by sequentially activating a single unit in the PFC_Output layer corresponding to the cur-

Table 1: Network Performance (SSE) During Early Stages of Learning.

Sequence	Both Pathways (High Control)	Controlled Pathway Alone	Automatic Pathway Alone
Sequence 1	12.4 (±5.1)	22.2 (±6.4)	296.4 (±25.6)
Sequence 2	19.2 (±2.2)	21.6 (±2.9)	214.6 (±9.4)
Sequence 3	16.8 (±2.5)	18.2 (±3.7)	234.0 (±22.7)

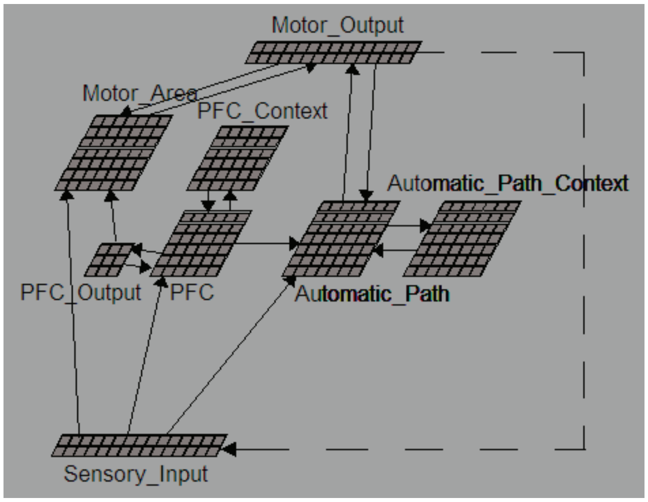

Figure 2: The Leabra network. Each gray box corresponds to a neural processing unit. Each arrow represents complete interconnectivity between the units in two layers. The dashed line from the Motor_Output layer to the Sensory_Input layer signifies that, when the arm is unguided, the output at the previous time step is the input for the next time step.

rent target key. The PFC_Context layer feeds the PFC layer activity from the previous time step. The PFC_Output layer, as well as the Sensory_Input layer, provide input to the Motor_Area layer. The Motor_Area layer translates the current key target, in PFC_Output, and the current Sensory_Input into an appropriate action at the Motor_Output. It is important to note that, during training, the PFC_Output layer receives an explicit error signal (as does the Motor_Output layer), driving the PFC to learn to produce the correct sequence of target keys. Finally, the PFC layer also provides input to the Automatic_Path layer. This input helps guide learning for the automatic execution of the sequence.

Our model includes a cognitive control modulation mechanism. This mechanism modulates the strength of the controlled pathway's contribution to the final motor output as well as the strength of the input going from the controlled pathway to the automatic pathway. Cognitive control is modulated as follows:

$$Control_{new} = \lambda \, Control_{old} + (1 - \lambda)\,(\alpha \, Conflict + \beta)$$

$Control_{new}$ specifies the value of control for the current trial. This value, which is between 0 and 1, is used to scale the weights from the controlled pathway (using Leabra's relative weight scaling parameter). $Control_{old}$ specifies the value of control for the previous trial. α, β and λ are constants, with values of 1, 0 and 0.6 respectively, determined by an ad hoc search. $Conflict$ is a normalized measure of performance error, and it is computed as follows:

$$Conflict = \frac{Error - \theta}{\gamma}$$

where $Error$ is the sum squared error (SSE) produced in the Motor_Output layer during the previous trial. θ and γ are constants with values of 10 and 80, determined by an ad hoc parameter search. If the value of $Control$ is less than 0.15, it is thresholded to 0. If the value of $Control$ is greater than 1, it is set to 1. Hence, the magnitude of control is approximately proportional to a running average of output error over previous trials. When error has been high, control will be high, and the influence of the controlled pathway will be strong.

The focus of this work is on the learning of specific motor skills, rather than on the development of basic motor competence. Thus, it was assumed that the system included the means to generate a reaching motion to a single target key, with the identity of that key being actively maintained in PFC. This PFC-controlled reaching process was implemented in the pathway from the PFC_Output, through the Motor_Area, to the Motor_Output. This portion of the network was pre-trained to capture fundamental motor competence. During pre-training, the network experienced every possible arm configuration along with every possible target key, and it was trained, using Leabra's standard synaptic modification rules, to produce the next time step of an appropriate reaching response. Once this pre-training was complete, learning was disabled for all projections going into or out of the Motor_Area layer.

In order to examine the learning properties of our model, we trained it to produce three randomly generated 10-key sequences. Each simulation involved the learning of one of these sequences. Random generation of the key sequences resulted, at the finer level of arm motion time steps, in sequences of 57, 42 and 51 arm states for the three sequences. For each sequence, we examined the learning profile of each of the two pathways when isolated, as well as the performance of the model as a whole. Each simulation was repeated five times with different random initial synaptic weights in the network, and we report the mean and standard error of the mean over these five repetitions for each measurement taken.

5 Results

Initially, the automatic pathway was disabled, and only the controlled pathway was trained. On each trial, the initial arm state was presented at the network's input, and this triggered

Table 2: Network Performance (SSE) After Extensive Training.

Sequence	Both Pathways (High Control)	Controlled Pathway Alone	Automatic Pathway Alone
Sequence 1	66.4 (±4.8)	30.2 (±4.0)	29.0 (±4.5)
Sequence 2	36.3 (±2.2)	24.6 (±3.0)	19.3 (±2.2)
Sequence 3	37.4 (±3.2)	18.8 (±3.7)	11.4 (±2.5)

the selection of a target key at the PFC_Output layer. A training signal was then provided to this layer, specifying the correct target key. The correct target was then actively maintained in the PFC while the Motor_Area layer generated the corresponding reaching motion. Once each reach was complete, the PFC was allowed to rapidly update, based on the activity in the PFC_Context layer and the Sensory_Input layer, selecting a new target key at PFC_Output. A training signal was then provided, once again, to PFC_Output, and this process continued until the end of the sequence, and the end of the trial, was reached. Through this training process, the controlled pathway learned quickly. An average of 11.8 (±1.8), 18.6 (±2.0), and 13.2 (±3.1) trials were required for the controlled pathway to learn the three sequences to criterion, respectively.

Next, the controlled pathway was disabled in order to examine the learning performance of the automatic pathway. Once again, each trial began with the initial arm position being provided as input. Synaptic weight changes were made in response to training signals provided at the Motor_Output layer, with performance error driving learning in the standard Leabra manner. The arm was guided from key to key in the sequence, forcing the Sensory_Input to always fall along the correct trajectory. This process continued until the motor sequence was complete. An average of 83.0 (±10.4), 76.4 (±5.8) and 70.2 (±7.0) trials were required for this pathway to master the three sequences, respectively. Hence, the time required to train the automatic pathway was found to be substantially greater than the time required to train the controlled pathway for all three sequences. Clearly, learning the declarative sequence of key identities was easier than learning the nuanced motor trajectory needed to visit every key in order. This provides an explanation for why learning in the controlled pathway is generally faster, allowing it to dominate performance early in skill acquisition.

Finally, the complete model was trained on each key sequence, with the control modulation mechanism determining the strength of the controlled pathway on any given trial. Initially, performance error was high. This quickly resulted in a high level of control (i.e., a control value of 1), maximally increasing the influence of the controlled pathway. Because the controlled pathway can learn rapidly, error then dropped rapidly. This drop occurred after 15 (±1.3), 19.8 (±4.9) and 15.2 (±2.3) training trials for the three sequences, respectively. Network performance at this point in training is shown in Table 1, alongside the performance that arose when each pathway was temporarily disabled at this point in training. Note that correct motor sequences were produced by the intact model and by the controlled pathway alone but not by the automatic pathway in isolation. This is consistent with the observation that human performance suffers early in skill learning when the controlled pathway is disrupted (e.g., under PFC-based working memory load).

Another interesting observation is that the full model was able to generate the correct sequence despite the automatic pathway's tendency to generate incorrect responses. It appears as if the controlled pathway, which was the primary contributor to the correct output, learned to compensate for some of the erroneous activity from the automatic pathway. This may be the reason why the error for the isolated controlled pathway is slightly greater than the error for the full model. The isolated controlled pathway might have been overcompensating for an automatic pathway that was no longer present.

Training was continued past this point. When the control was up-modulated to a high value, the network produced correct motor sequence due to the corresponding frontal involvement. However, as correct outputs were generated, the running average of error decreased and the strength of control dropped. When this happened, the controlled pathway's contribution to the motor output decreased, bringing error back up and strengthening control. Thus, control oscillated close to its maximum level. During this entire process, the automatic pathway continued to learn. When the strength of control was high, the network generated correct output. Since the amount of error was negligible on these trials, Leabra's error-correction learning rule played only a small role, and the automatic pathway learned primarily through the Hebbian component of the learning algorithm. When control dipped and significant error appeared at the output, the automatic pathway benefited from the error driven learning component of Leabra. For the three sequences, the automatic pathway needed an average of 334.4 (±42.5), 76.8 (±4.6) and 233.2 (±21.4) training trials, respectively, to master the task. Once the automatic pathway learned the sequence, the strength of control dropped to 0. This signified that no control was being employed and the task had been automatized.

Network performance at this late stage of learning is shown in Table 2. At this point, each pathway produced reasonable performance when isolated from the other. Interestingly, error increased when both pathways were incorporated and control was set to its maximum level. Thus, our model suffers when excessive control is employed during the execution of an automatized motor skill, just as is observed in humans who are performing a well-practiced skill under pressure [Beilock *et al.*, 2004]. Late in training, as control reached its minimum value of 0, the automatic pathway learned to generate the correct motor sequence without any input from the controlled pathway. Hence, the introduction of control resulted in unwanted frontal input, degrading performance.

A final curious observation is that the time required by the automatic pathway to learn the sequence in the full model is substantially greater than the time needed when the automatic pathway is trained alone. This occurred because the controlled pathway kept the network error low, limiting the utility of Leabra's error driven learning mechanism and causing connection weights to change more slowly.

6 Discussion

We have reported some initial explorations of a computational cognitive neuroscience model of automaticity in motor skill learning. The use of this computational framework now gives us the capability to produce both qualitative and quantitative predictions concerning human behavior. In our model, a declarative representation of the skill is quickly acquired in the frontal controlled pathway. With additional practice, a procedural representation of the skill is also acquired in the automatic pathway. As the automatic pathway becomes more and more proficient, the contribution of the controlled pathway is gradually retracted by a control modulation mechanism.

For the simple sequence learning task that was explored in these simulations, the controlled pathway was faced with the relatively simple task of learning a sequence of key identities. The actual motor output was initially produced as a succession of reaching motions that were generated by a pre-trained component of the model. Not all motor tasks lend themselves to such a simple declarative representation, however. While skilled motions like a golf put or a ping pong smash can definitely be broken down into discrete declarative steps, the actual execution of each of those steps is not as simple as a previously-mastered reaching behavior. When learning such skills, it might be necessary for the motor areas participating in the controlled pathway to learn to execute each component step, limiting the utility of the controlled pathway.

Some theories of automaticity suggest that the declarative component can assist in the training of the procedural component. This happens, in a small way, in our model. Early in training, the controlled pathway produces correct output activation levels, and this allows Hebbian learning in the automatic pathway to improve performance in that pathway. Hebbian learning in Leabra is fairly weak, however, particularly in comparison to the error driven learning mechanisms used in this framework. We intend to explore ways in which this this interaction can be strengthened, allowing the controlled pathway to "teach" the automatic pathway.

The main limitation of our model is that it does not yet capture execution-time differences between controlled processing and automatic processing. It is well established that controlled execution of an skill is slower than automatic execution. This is our most pressing matter for future research.

Acknowledgements

This work was supported, in part, by the USA National Science Foundation under grant EIA-0325641. The authors also extend their thanks to the members of the Vanderbilt Computational Cognitive Neuroscience Laboratory (CCNL) and to three anonymous reviewers.

References

[Anderson, 1981] John R. Anderson, editor. *Cognitive Skills and Their Acquisition*. Lawrence Erlbaum, Hillsdale, New Jersey, 1981.

[Bapi *et al.*, 2000] R. S. Bapi, K. Doya, and A. M. Harner. Evidence for effector independent representations and their differential time course of acquisition during motor sequence learning. *Experimental Brain Research*, 132(4):149–162, 2000.

[Beilock *et al.*, 2004] S. L. Beilock, B. L. Bertenthal, A. M. McCoy, and T. H. Carr. Haste does not always make waste: Expertise, direction of attention, and speed versus accuracy in performing sensorimotor skills. *Psychonomics Bulletin and Review*, 11(2):373–379, 2004.

[Botvinick *et al.*, 2001] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen. Conflict monitoring and cognitive control. *Psychological Review*, 108(3):624–652, 2001.

[Gupta and Noelle, 2005a] A. Gupta and D. C. Noelle. Neurocomputational mechanisms for generalization during the sequential learning of multiple tasks. In *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, 2005.

[Gupta and Noelle, 2005b] A. Gupta and D. C. Noelle. The role of neurocomputational principles in skill savings. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, pages 863–868, 2005.

[Hikosaka *et al.*, 2002] O. Hikosaka, K. Nakamura, K. Sakai, and H. Nakahara. Central mechanisms of motor skill learning. *Current Opinion in Neurobiology*, 12:217–222, 2002.

[Nakahara *et al.*, 2001] H. Nakahara, K. Doya, and O. Hikosaka. Parallel cortico-basal ganglia mechanisms for acquisition and execution of visuomotor sequences — a computational approach. *Journal of Cognitive Neuroscience*, 13(5):626–647, 2001.

[OReilly and Munakata, 2001] R. C. OReilly and Y. Munakata. *Computational Explorations in Cognitive Neuroscience*. MIT Press, Cambridge, Massachusetts, 2001.

[Rand *et al.*, 2001] M. K. Rand, O. Hikosaka, S. Miyachi, X. Lu, K. Nakamura, K. Kitaguchi, and Y. Shimo. Characteristics of sequential movements during early learning period in monkeys. *Experimental Brain Research*, 131:283–304, 2001.

[Rougier *et al.*, 2005] N. P. Rougier, D. C. Noelle, T. S. Braver, J. D. Cohen, and R. C. O'Reilly. Prefrontal cortex and the flexibility of cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences*, 102(20):7338–7343, 2005.

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

[Wolpert *et al.*, 2001] D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan. Perspectives and problems in motor learning. *Trends in Cognitive Sciences*, 5(11):487–494, 2001.