

Available at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/bica

Research Article

A neurocomputational approach to automaticity in motor skill learning

Ashish Gupta ^a, Lovekesh Vig ^b, David C. Noelle ^{c,*}

^a Google India Pvt. Ltd., Bangalore, India

^b School of Information Technology, Jawaharlal Nehru University, New Delhi, India

^c Cognitive and Information Sciences, University of California, Merced, CA, USA

Received 27 June 2012; received in revised form 9 July 2012; accepted 9 July 2012

KEYWORDS

Motor learning;
Skill learning;
Automaticity;
Cognitive control;
Computational cognitive neuroscience

Abstract

Cognitive agents physically interacting with the world can best adapt to their task environments if they are able to learn motor skills from experience. Many cognitive architectures have focused on a single learning mechanism to accomplish such adaptation. Behavioral studies with humans, however, have shown that the acquisition of a motor skill generally occurs in two stages. In the initial stage, acquisition is performed via attention-demanding neural processes, producing a high cognitive load. This is followed by more fluent automatic processing, requiring less deliberation. Neuroscientific studies have since identified two relevant interacting neural systems, suggesting that the acquisition of a motor skill involves a transition from heavy dependence on a system involving cognitive control to only weak dependence on such a system. This cognitive control system, which includes the prefrontal cortex, is thought to be responsible for acquiring and manipulating declarative representations of skills. This frontal system is seen as modulating processing in a more automatic neural pathway, which develops procedural representations over time. In this paper, we propose a biologically plausible computational model of motor skill automaticity. This model offers a neurocomputational account of the translation of declarative into procedural knowledge during learning. In support of the model, we review some previously reported experimental results, and we demonstrate, through simulation, how the model provides a parsimonious explanation for these results. The model is seen as exemplifying a novel approach to motor skill learning in artificial agents.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Motor skills are integral to an agent's interaction with its environment. They enable the agent to move,

* Corresponding author.

E-mail addresses: ashish2240@gmail.com (A. Gupta), lovekeshvigin@gmail.com (L. Vig), dnoelle@ucmerced.edu (D.C. Noelle).

explore, learn, and accomplish critical goals. In complex environments, the full range of important motor skills may not be foreseen, making the dynamic acquisition of motor skills important for adaptation. Standard machine learning approaches to motor skill learning often fail to meet the real time requirements of online adaptation, however, and these methods have general difficulties with scaling to the high-dimensional sensors and manipulators common in domains such as humanoid robotics. There have been many attempts to create a learning framework that enables robots to autonomously learn complex skills (Schaal, 1999; Schaal & Atkeson, 2010; Wolpert & Kawato, 1998). While some of these efforts have been profoundly inspired by human learning capabilities, a clear and complete computational account of how humans acquire motor skills remains elusive. This paper aims at addressing this deficit by offering insights into the neurocomputational structure of human 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 distinguishable neural networks in the brain (Bapi, Doya, & Harner, 2000; Hikosaka, Nakamura, Sakai, & Nakahara, 2002; Wolpert, Ghahramani, & Flanagan, 2001). While motor skill learning can proceed, to some degree, in each of these neural networks separately, the two networks typically coordinate with each other during learning. In broad strokes, the *controlled pathway* includes the dorsolateral prefrontal cortex (DLPFC), the 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 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 from controlled to more automatic processing? 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 previously developed *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. In a sense, this pathway learns, over time, to encode a kind of “muscle memory” for the key sequence to be produced, without any sort of declarative representation of the sequence.

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.

The remainder of this paper reports on computational simulations of this model of automaticity, demonstrating its ability to account for human skill learning. First, Section 2 briefly summarizes relevant past work on automaticity from the cognitive science literature, including work on computational models of skill learning. Section 3 describes in detail the learning task being investigated and the neurocomputational model used in the reported simulations. The simulation results are provided in Section 4, followed by a discussion of the results.

2. Previous work on skill learning

2.1. Automaticity

Early psychologists noticed the fact that certain human behaviors can become automatic through habitual training (Huey, 1908; James, 1983). Guthrie (1935) studied the incremental improvement in task performance with practice and made the distinction between “Acts” and “Movements”. Acts are complex responses that are comprised of a number of different movements (Houston, 1991). To master a complete complex act, many different responses must be connected to many different stimulus configurations. According to Guthrie, movements are learned in a binary (all or nothing) fashion, and as an individual practices, he/she learns more of these movements which leads to gradual improvement in the overall act performance.

Schneider and Schrifin (1977) have experimentally distinguished between controlled and automatic processing. Controlled processes are required for novel tasks or those tasks requiring undivided attention. Controlled processes are sequential and are carried out in a stepwise fashion. Repeated practice allows some tasks to be performed without the need for devoted attentional resources and are performed to completion in the presence of the initiating stimulus. Automatic processes are characterized by this decrease in the attentional resources allocated for these tasks, often accompanied by a speeding of task performance. Most behaviors involve multiple processes and components, some of which are automatic and some that will be controlled.

Schmidt (1975) proposed a theory of motor skill learning which accounts for the improvement and increased speed of motor skills with practice as being due to the development of schemas for the movements. According to this theory, generalized motor programs control actions. As a class of movements is practiced, the performer learns the appropriate parameter values to supply to the program and the movements become faster and more accurate.

Neumann (1984) reviewed the notion that automatic processing occurs in a passive, bottom-up fashion, independent of intentions and free of interference. According to Neuman, automaticity is not an intrinsic property of the process, but an emergent property depending on both the processing system and situational context. This suggests that there are two different models for automaticity, one for perceptual tasks and another for motor tasks. Automatic components of perceptual tasks may arise from the hierarchical nature of many perception tasks, with automaticity at higher levels of processing building on improvements at lower levels. Motor tasks, on the other hand, involve some type of behavioral performance, and automatic components of these tasks may require the parameters of the task to be learned. In our own work, we have remained sensitive to the presence of tight coupling between perceptual and motor processes in most behaviors, causing us to attend to work on perceptual automaticity, but the model that we present in this paper focuses on motor aspects of automaticity.

Masters (1992) argued that if the internal representation of a movement was entirely procedural (perhaps learned through an implicit training process), free of any declarative components (which might arise from explicit instructions during training), then, on occasions in which the performer comes under pressure or stress, these movements will be subject to less interference than explicitly learned, declaratively represented, movements. In other words, many forms of pressure or stress appear to interfere with declarative skill encodings in a way that does not arise for more procedural skill encodings. This finding was replicated by Hardy, Mullen, and Jones (2004), and it is consistent with the two pathways in our model, with the automatic pathway in our network being procedural and the controlled pathway being declarative.

Scientists are only beginning to uncover the changes in neural activity in different brain regions as a task becomes automatized. Wu, Kansaku, and Hallett (2004) conducted a functional MRI study to investigate the physiology of how movements become automatic. Their study involved subjects performing a sequential task along with a

secondary task. The fMRI results before and after automaticity revealed no additional activity in the automatic condition but activity was less in the bilateral cerebellum, presupplementary motor area, cingulate cortex, left caudate nucleus, premotor cortex, parietal cortex, and prefrontal cortex during the automatic stage. These findings suggest that most of the motor network participates in executing automatic movements and that it becomes more efficient as movements become more automatic. Wu further investigated changes in the effective connectivity of the brain motor networks when movements become automatic (Wu, Chan, & Hallett, 2008). They found that the cerebellum, cingulate motor area, supplementary motor area, and putamen had significantly greater connectivity, whereas the precuneus had less connectivity in the motor networks at the automatic stage. These findings are reflected in our model in the way in which activity in neural networks associated with top-down attention and cognitive control decreases as movements become automatic.

Despite the considerable volume of work by cognitive scientists, and more recent work by neuroscientists, on the subject of automaticity, to the best of our knowledge there have not been any neurocomputational models of automaticity explored with a view towards robotic learning. Our work aims to bridge this gap. The automaticity of motor skills, as exhibited by humans, has several benefits that may transfer to robotic applications. Some benefits include conservation of attentional resources, which may be spent on other processes, multi-task functioning, and greater retention of learned material (Holt & Rainey, 2002). Automaticity does have the drawback, however, of typically requiring extensive practice, though such practice might be completed relatively quickly in the case of some robotic tasks.

2.2. Cognitive control

Cognitive control refers to the capability of suppressing and overcoming more prepotent or reflexive responding in favor of a more adaptive behavior in the current context. The prefrontal cortex (PFC) plays a critical role in supporting robust cognitive control (Cohen, Braver, & O'Reilly, 1996; O'Reilly & Munakata, 2000). Dense recurrent connectivity in PFC allows it to actively maintain information in firing rate patterns, acting as a kind of *working memory* for control state. Through broad neural projections from PFC, it is thought that these patterns modulate processing elsewhere in the brain in a task-appropriate manner. Leveraging reinforcement learning mechanisms in the midbrain, involving the dopamine (DA) system, connections between PFC and the basal ganglia (BG) support the learning of when the pattern of PFC neural firing should be updated in order to achieve task success (Braver & Cohen, 2000; O'Reilly, Noelle, Braver, & Cohen, 2002; Rougier, Noelle, S., Cohen, & O'Reilly, 2005). We have incorporated both active maintenance and adaptive updating mechanisms of this kind in the PFC component of our model.

The cognitive control signals actively maintained in the PFC modulate activity in other brain areas through extensive projections (Cohen et al., 1996). This modulating activity can be relatively strong, when PFC activity is high and sent to more posterior brain areas, or relatively weak, when

PFC activity is low or projections are inhibited (Kriete & Noelle, 2011). The strength of cognitive control can be modulated based on the agent's task performance. Botvinick, Braver, Barch, Carter, and Cohen (2001) proposed that the anterior cingulate cortex (ACC) monitors the amount of conflict between parallel neural pathways and strengthens cognitive control when conflict between these pathways 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 substantially 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 generally remains robust if the controlled pathway is disrupted.

The shifting dependence on cognitive control systems in skill learning is also seen in studies of *choking under pressure*. These studies have suggested that performance errors in the face of stress may have different causes early and late in learning (Beilock, Bertenthal, McCoy, & Carr, 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 on an otherwise robust automatic pathway.

2.4. Previous computational models

Ours is certainly not the first computational model of motor skill learning. A wide variety of approaches have been proposed for the modeling of human cognitive skill learning in general and motor skill learning in particular.

Multiple symbolic computational cognitive architectures, utilizing production-rule-like knowledge representation schemes, have incorporated mechanisms for skill learning and automaticity. For example, modeling the acquisition of complex sequential cognitive skills was central to the development of the Soar cognitive architecture (Newell, 1990). The core mechanism of learning in the Soar framework involves a process of *chunking*, reducing a chain of production-like rule applications to individual, compact, situation-specific rules (Laird, Newell, & Rosenbloom, 1987). This chunking mechanism captures a broad range of human learning phenomena, including aspects of the rate of learning with practice, but it does not inherently embody a representational shift from declarative to procedural

representations. Thus, chunking in Soar is roughly analogous to learning in the automatic pathway of our model. More recent accounts of skill acquisition in Soar have included declarative representations, building upon the foundational chunking mechanism, moving the framework toward a kind of dual-pathway account (Laird, 2012).

A similar theoretical progression can be seen in accounts of skill acquisition in the ACT-R cognitive architecture (Anderson & Lebiere, 1998). While, like Soar, the ACT-R framework makes use of symbolic production rules to encode procedural knowledge, the foundational mechanisms for learning in ACT-R are sub-symbolic. Continuous parameters that control the rate of memory retrieval are adapted by experience, speeding the retrieval of relevant knowledge and the subsequent application of productions dependent on that knowledge. As in Soar, this foundational learning mechanism reflects aspects of our automatic pathway, but lacks a transformation from declarative to procedural representations with practice. Such a representational transformation has been more recently explored in the ACT-R framework through the introduction of a mechanism of *production compilation* (Taatgen & Lee, 2003). This mechanism incrementally translates declarative encodings of a skill into new production rules. While this compilation approach moves the ACT-R account of skill learning toward a more dual-pathway structure, the translation process is still largely autonomous, making it unclear how this account can capture how more declarative processes can reassert control over behavior after extensive learning. Finally, it is worth noting that much of the work on skill learning in symbolic cognitive architectures has focused on cognitive skills, though some efforts have been made to address aspects of motor skill learning, as well (Byrne & Anderson, 1997).

Outside of the literature on symbolic cognitive architectures, detailed computational models have been developed with a specific focus on motor skill acquisition. For example, Wolpert and Kawato (1998) and Wolpert et al. (2001) have 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.

Traditionally, complex motor skills have been considered to be broken down into a schema hierarchy of skills. Botvinick and Plaut (2004) have proposed an alternative explanation using a simple recurrent network model of sequential motor skill learning. Their model is able to explain many of the behavioral phenomena related to routine sequential learning tasks. Like MOSAIC, this work has focused on routine skills, capturing the abilities of the automatic pathway in our model. Our model can be seen as extending this previous work by incorporating early learning guided by a cognitive control mechanism. More specifically, the automatic pathway in our model is grounded in previous computational cognitive neuroscience models of skill learning developed in Gupta and Noelle (2005a, 2005b), making use of the Leabra modeling framework (O'Reilly & Munakata, 2000). These previous neural network models were used to explore the neurocomputational principles underlying skill savings and the transfer of knowledge from one skill to another.

Nakahara, Doya, and Hikosaka (2001) proposed a skill learning model that is similar in general architecture to our own. This model does not focus on the question of differences in declarative and procedural representations of a skill, but it does include separated controlled and automatic pathways. The early dominance of the controlled pathway, in this model, is driven by differential learning rates, with the controlled pathway forced to learn faster. Also, the Nakahara model does not include a mechanism for dynamically adjusting cognitive control—the relative contribution of the controlled pathway.¹ Hence, in this model, once a skill has been automatized, its controlled execution, based on task demands, is not possible. This is inconsistent with the behavioral observations. While our own approach also involves controlled and automatic pathways, the learning mechanisms of our model are substantially different than those used by Nakahara. Rather than biasing learning rates in order to explain the shift from controlled to automatic behavior, 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, our model dynamically adjusts control as a function of task performance. 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.

Contributions to the theory of automaticity have also been derived from results in category learning theory. In particular Ashby et al. developed the COVIS model of automaticity (Ashby, Alfonso-Reese, Turken, & Waldron, 1998; Ashby & Valentin, 2005), which is closely related to our model. COVIS postulates two systems that compete throughout learning: a frontal-based explicit system that uses logical reasoning and depends on working memory and executive attention and a BG-mediated implicit system that uses procedural learning. COVIS, however, fails to model data relating to highly overlearned behaviors and was later extended to the SPEED model (Ashby, Ennis, & Spiering, 2007) which accounted for experimental data pointing to a more limited role of the basal ganglia in the automatic pathway, positing a separate pathway for automatic task learning. Our model does not currently capture overlearning but does not rely on the BG for automatic learning, either. In this way, our model is consistent with SPEED but diverges from other computational accounts of motor learning that focus on automaticity arising in the BG (Shah & Barto, 2009), following more recent research that suggests that the role of the BG in the storage and execution of movement sequences may have been previously overstated (Desmurget & Turner, 2010; Turner & Desmurget, 2010).

3. A motor skill learning model

3.1. The Leabra modeling framework

Perhaps one of the most important differences between our model and previous models of skill learning is that our

¹ While this model does include a “coordinator”, the function of this mechanism is not one of modulating declarative control.

model grounds the more abstract theories in well established neurocomputational mechanisms. In other words, our model demonstrates that the declarative/procedural translation theory fits naturally with neurocomputational primitives. We have built our model using the Leabra computational cognitive neuroscience framework (which is reviewed in more detail in the appendix of this report) (O’Reilly & Munakata, 2000). Thus, our model involves grounding the proposed learning processes in simulated neurons and synapses, rather than more abstract processes. In contrast, past models have tended to be mathematically abstract in nature, often leaving it unclear how these are to be translated into neural accounts of skill learning.

The Leabra framework offers a collection of integrated cognitive modeling formalisms that are grounded in known properties of cortical circuits while being sufficiently abstract to support the simulation of behaviors arising from large neural systems. This framework 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 (O’Reilly & Munakata, 2000). Leabra models have successfully illuminated cognitive function in a wide variety of domains, including perception, object recognition, attention, semantic memory, episodic memory, working memory, skill learning, reinforcement learning, implicit learning, cognitive control, and various aspects of language learning and use (O’Reilly & Munakata, 2000). Of particular relevance to skill learning are Leabra’s lateral inhibition formalism and its synaptic learning mechanism.

In the neocortex, two general patterns of connectivity have been observed involving inhibitory neurons and their interactions with excitatory neurons, namely *feedforward* and *feedback* inhibition (O’Reilly & Munakata, 2000). Feedforward inhibition occurs when the inhibitory interneurons in a cortical region are driven directly by the inputs to that region, producing rapid inhibition of the excitatory neurons in that area. Feedback inhibition occurs when the same neurons that excite nearby inhibitory interneurons are, in turn, inhibited by the cells they excite, producing a kind of negative feedback loop.

The effects of inhibitory interneurons tend to be strong and fast in the 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 (O’Reilly & Munakata, 2000). As inputs become stronger, they drive inhibitory interneurons as well as excitatory pyramidal cells, producing a dynamic balance between excitation and inhibition. 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. The amount of lateral inhibition within a layer can be parameterized in a number of ways, with the most common being the percentage of the units in the layer

that are expected, on average, to surpass threshold. A layer of neural units with a small value of this k parameter (e.g., 10–25%) will produce sparse representations, with only a small fraction of the units being active at once.

With regard to learning, Leabra modifies the strength of synaptic connections in two primary ways. An error-correction learning algorithm changes synaptic weights so as to improve network task performance. Unlike the backpropagation of error algorithm, 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. This form of correlational learning allows Leabra to capture certain effects of overlearning.

More details concerning the Leabra computational modeling framework may be found in the appendix of this report.

3.2. A sequential key pressing task

We used our model to simulate the learning of key pressing motor sequences. This task was largely selected for its relatively simple formal structure, supporting formal analysis of our model's performance. The learning of motor sequences of key depressions also has the advantage of having been widely studied in humans, focusing on the learning of repetitive office activities, such as 10-key operation and keyboard typing skills (Adams, 1987). Indeed, skilled performance on key pressing tasks is often considered a prototypical example of the kinds of automatized motor behavior characterized as "muscle memory".

In our simulations of a sequential key pressing task, our model controls a simulated 2-joint planar arm which moves over a 9-key keyboard, as shown in Fig. 1. The state of the arm at any point in time is represented by the vector (q_1, q_2) , where q_1 and q_2 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.

3.3. The neural network model

Fig. 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 unit preference, as well

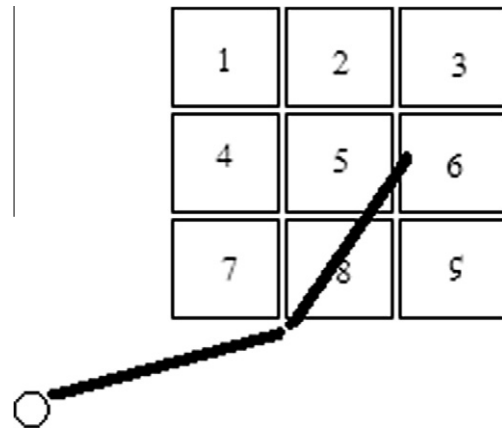


Fig. 1 A two joint planar arm and a keyboard. The state of the arm at any point in time is given by the vector of joint angles (q_1, q_2) . The arm produces motion trajectories such that its end effector moves from one key to the next, in sequence. The joint angles are not constrained to those that place the end effector over a key, allowing for smooth trajectories between keys, as shown in the diagram.

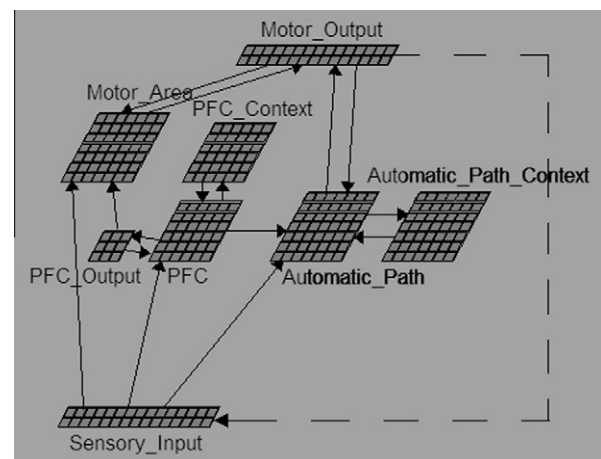


Fig. 2 The Leabra network. Each gray box corresponds to a neural processing unit. Each arrow represents complete interconnectivity between the units in two layers. The dashed line from the Motor_Output layer to the Sensory_Input layer signifies that, when the arm is unguided, the output at the previous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

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 of the two joint angles 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 networks used in Gupta and Noelle, 2005b, 2005a, 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 (Gupta & Noelle, 2007). 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 current 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.² $Conflict$ 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.³ If the value of $Control$ is less than

² Model performance was largely insensitive to small changes in these parameter values, but substantial variations in λ strongly influenced the rate of automaticity.

³ These parameters are essentially scaling constants, transforming an SSE value to a range roughly between zero and one. Thus, different parameter values would be needed for different tasks, as the possible error range is task-dependent.

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 experienced a period of "pre-training" (i.e., training prior to efforts to learn a specific key sequence) which was intended to capture the development of fundamental motor competence. During this pre-training process, 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. At this point, the network possessed the basic motor capability of reaching to a target key.

In order to examine the learning properties of our model, we trained it to produce ten 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 different arm states for the ten sequences as shown in Tables 1 and 2. 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 (a value limited only by the availability of computational resources) 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.

4. 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 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 relatively quickly. The average number of trials that were required for the controlled pathway to learn each of the ten sequences are displayed in Table 1.

Table 1 Number of trials required for training (early stages).

Sequence	Arm states	controlled pathway alone	automatic pathway alone	both (high control) pathways
1	57	21.8 (± 1.8)	83.0 (± 10.04)	15.0 (± 1.3)
2	42	18.6 (± 2.0)	76.4 (± 5.9)	19.8 (± 4.9)
3	51	13.2 (± 3.1)	70.2 (± 7.0)	15.2 (± 2.3)
4	43	14.5 (± 2.6)	76.3 (± 8.1)	20.4 (± 3.9)
5	50	12.7 (± 2.9)	81.6 (± 11.4)	22.5 (± 2.7)
6	49	15.3 (± 2.8)	73.5 (± 8.5)	11.1 (± 2.5)
7	54	18.9 (± 2.1)	75.1 (± 9.7)	18.4 (± 3.1)
8	44	12.3 (± 1.6)	66.9 (± 7.7)	10.6 (± 2.0)
9	59	20.2 (± 3.7)	81.8 (± 10.3)	25.3 (± 4.9)
10	47	16.8 (± 1.8)	75.9 (± 8.0)	10.9 (± 2.2)

Table 2 Network performance (SSE) during early stages of learning.

Sequence	Arm states	Both pathways (high control)	Controlled pathway alone	automatic pathway alone
1	57	12.4 (± 5.1)	22.2 (± 6.4)	296.4 (± 25.6)
2	42	19.2 (± 2.2)	21.6 (± 2.9)	214.6 (± 9.4)
3	51	16.8 (± 2.5)	18.2 (± 3.7)	234.0 (± 22.7)
4	43	11.8 (± 2.1)	19.4 (± 4.2)	256.3 (± 18.3)
5	50	15.3 (± 3.3)	20.2 (± 4.5)	206.7 (± 28.2)
6	49	14.5 (± 2.9)	19.6 (± 3.2)	223.6 (± 16.6)
7	54	15.9 (± 3.0)	19.1 (± 2.7)	241.2 (± 25.5)
8	44	14.2 (± 2.6)	20.7 (± 4.9)	233.3 (± 24.9)
9	59	17.9 (± 3.3)	18.2 (± 5.7)	263.9 (± 12.7)
10	47	17.1 (± 2.5)	19.8 (± 5.3)	245.6 (± 20.7)

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. The average trials required for this pathway to master the ten sequences are given in Table 1. 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 ten 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. The trials required for this drop to

occur are recorded in Table 1 for the ten sequences. Network performance at this point in training (when error first reached a consistently low level) is shown in Table 2, 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) (Holt & Rainey, 2002; Hikosaka et al., 2002).

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 when the controlled pathway was removed. 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.

Indeed, there is further evidence of complex interactions between the pathways during early learning in the complete model. Specifically, some key sequences were learned more quickly by the complete model than by the controlled

pathway, alone, indicating that the presence of the automatic pathway actually sped early learning in the complete model. This suggests that, while the automatic pathway may have mostly provided erroneous activity to be overcome by the controlled pathway during these early stages of learning, some small positive contribution to correct performance was made by the automatic pathway, allowing the complete model to reach low levels of error more quickly in some cases.

Training of the full model was continued past this point. When the level of cognitive control had a high value (i.e., close to one), the network produced the 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 outputs. 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 correlational (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 ten sequences, the average training trials required for the automatic pathway to master the task are recorded in Table 3. Once the automatic pathway learned the sequence, the strength of control dropped to zero. This signified that no control was being employed and the task had been automatized.

Network performance at this late stage of learning (determined by the point at which cognitive control persisted at a consistently low level), is shown in Table 4. 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 zero, 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.

5. Discussion

We have reported some initial explorations of a neurocomputational model of automaticity in motor skill learning. The use of this computational framework now gives us the capability to produce 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 (Anderson, 1981; Schneider & Schrifflin, 1977). This happens, in a non-obvious way, in our model. Early in training, the controlled pathway produces correct output activation levels, and this allows correlational (Hebbian) learning in the automatic pathway to improve performance in that pathway. Correlational 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 interaction can be strengthened, allowing the controlled pathway to "teach" the automatic pathway.

Recent research suggests that the role of the basal ganglia in the storage and execution of movement sequences may have been previously overstated (Desmurget & Turner, 2010; Turner & Desmurget, 2010). This may imply separate pathways for procedural and automatic processing. Our model would have to be suitably modified to account for these results and to capture overlearning. Another

Table 3 Number of trials required for training (later stages).

Sequence	Arm states	Automatic pathway alone
1	57	334.4 (± 42.5)
2	42	76.8 (± 4.6)
3	51	233.2 (± 21.4)
4	43	200.3 (± 3.9)
5	50	226.1 (± 2.2)
6	49	119.4 (± 2.5)
7	54	188.3 (± 2.1)
8	44	100.6 (± 1.8)
9	59	255.3 (± 3.6)
10	47	144.6 (± 2.4)

Table 4 Network performance (SSE) after extensive training.

Sequence	Arm states	Both pathways (high control)	Controlled pathway alone	Automatic pathway alone
1	57	66.4 (± 4.8)	30.2 (± 4.0)	29.0 (± 4.5)
2	42	36.3 (± 2.2)	24.6 (± 3.0)	19.3 (± 2.2)
3	51	37.4 (± 3.2)	18.8 (± 3.4)	11.4 (± 2.5)
4	43	38.5 (± 2.6)	22.3 (± 3.1)	20.4 (± 2.3)
5	50	55.7 (± 2.9)	29.6 (± 3.4)	22.5 (± 2.7)
6	49	35.3 (± 3.2)	18.5 (± 3.7)	11.1 (± 3.0)
7	54	48.9 (± 3.5)	25.1 (± 3.9)	18.4 (± 3.4)
8	44	32.3 (± 2.7)	16.9 (± 2.7)	10.6 (± 2.3)
9	59	60.2 (± 4.7)	31.8 (± 4.1)	25.3 (± 4.7)
10	47	36.8 (± 3.3)	17.9 (± 3.8)	10.9 (± 2.5)

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.

Acknowledgments

The authors extend their thanks to the members of the Computational Cognitive Neuroscience Laboratory at the University of California, Merced and to three anonymous reviewers.

Appendix A. The Leabra framework

This appendix provides a compact overview of the primary mathematical components of the Leabra framework for computational cognitive neuroscience modeling (O'Reilly & Munakata, 2000).

A.1. Dendritic integration

A fundamental function of neurons is the transformation of incoming synaptic information into specific patterns of action potential output. An important component of this transformation is synaptic integration — the combination of voltage deflections produced by a myriad of synaptic inputs into a singular change in membrane potential. Leabra simulates this integration at the dendrite of the neuron via a weighted summation of all the input activations followed by functional transformation (normally sigmoidal) of the sum.

A.2. Point neuron approximation

Leabra uses a point neuron activation function that models the electrophysiological properties of real neurons, while simplifying their geometry to a single point. This function is nearly as simple computationally as the standard sigmoidal activation function, but the more biologically-based

implementation makes it considerably easier to model inhibitory competition, as described below. Further, using this function enables cognitive models to be more easily related to more physiologically detailed simulations, thereby facilitating bridge-building between biology and cognition.

A.3. Lateral inhibition

The processes involved in lateral inhibition are particularly relevant to the model presented in this paper. Lateral inhibition allows for competition between neurons involved in the encoding of stimuli. Along with the mechanisms of synaptic learning, this competition separates the neurons that associate the stimulus with responding, or acquisition neurons, from those which associate the stimulus with non-responding, called extinction neurons. The class of inhibitory functions that Leabra adopts are known as *k*-winners-take-all (*kWTA*) functions. A *kWTA* function ensures that no more than *k* units out of a total of *n* in a layer are active at any given point in time. This is attractive from a biological perspective because it captures the *set point* property of inhibitory interneurons, where the activity level is maintained through negative feedback at a roughly constant level (O'Reilly & Munakata, 2000).

A.4. *kWTA* function implementation

The *k* active units in a *kWTA* function are the ones receiving the most excitatory input (g_e). Each unit in the layer computes a layer-wide level of inhibitory conductance (g_i) while updating its membrane potential such that the top *k* units will have above threshold equilibrium membrane potentials with that value of g_i , while the rest will remain below firing threshold. The function computes the amount of inhibitory current g_i^θ that would put a unit just at threshold given its present level of excitatory input, where θ is the threshold membrane potential value. Computing inhibitory conductance at the threshold (g_i^θ), yields:

$$g_i^\theta = \frac{g_e^* g_e^- (E_e - \theta) + g_l g_l^- (E_l - \theta)}{\theta - E_i} \quad (1)$$

where g_e^* represents the excitatory input minus the contribution from the bias weight and $g_l g_l^-$, $g_e g_e^-$ are the total con-

ductances from the Potassium and Sodium channels respectively. The E_i and E_e variables refer to the equilibrium potentials for the potassium and sodium channels, respectively (O'Reilly & Munakata, 2000). The value of g_i is computed as an intermediate value between the g_i^0 values for the k th and $k + 1$ th units as sorted by level of excitatory conductance (g_e). This ensures that the $k + 1$ th unit remains below threshold, while the k th unit is above it. Expressed as a formula this is given by:

$$g_i = g_{k+1}^0 + q(g_i^0(k) - g_i^0(k+1)) \quad (2)$$

where $0 < q < 1$ determines where the inhibition lies between the k and $k + 1$ th units.

A.5. Leabra learning algorithms

Leabra provides for a balance between Hebbian and error-driven learning. Hebbian learning is performed using a conditional principal components analysis (CPCA) algorithm (O'Reilly & Munakata, 2000). Error-driven learning is performed using GeneRec (O'Reilly, 1996), which is a generalization of the Recirculation algorithm (Hinton & McClelland, 1988), and approximates Almeida-Pineda recurrent backpropagation (Almeida, 1987; Pineda, 1989).

A.6. Correlational (Hebbian) learning

The objective of the CPCA learning rule is to modify the weights for a given input unit (x_i) to represent the conditional probability that the input unit (x_i) is active when the corresponding receiving unit (y_j) is also active. This is expressed as:

$$w_{ij} = P(x_i = 1 | y_j = 1) = P(x_i | y_j) \quad (3)$$

In Eq. (3) the weights reflect the frequency with which a given input is active across the subset of input patterns represented by the receiving unit. If an input pattern occurs frequently with such inputs, then the resulting weights from it will be relatively large. On the other hand if the input pattern occurs rarely across such input patterns then the resulting weights will be small. The following weight update rule achieves the CPCA conditional probability objective represented by Eq. (3).

$$\Delta w_{ij} = \epsilon [y_j x_i - y_j w_{ij}] = \epsilon y_j (x_i - w_{ij}) \quad (4)$$

where ϵ is the learning rate parameter. The weights are adjusted to match the value of the sending unit activation x_i , weighted in proportion to the activation of the receiving unit (y_j). Thus inactivity of the receiving unit implies that no weight modification will occur. Conversely, if the receiving unit is very active (near 1), the update rule modifies the weight to match the input unit's activation. The weight will eventually come to approximate the expected value of the sending unit when the receiver is active (consistent with Eq. (3)).

A.7. Error driven learning

GeneRec implements error backpropagation using locally available activation variables thereby making such a learning rule biologically plausible. The algorithm incorporates

the notion of plus and minus activation phases. In the *minus phase*, the outputs of the network represent the expectation or response of the network, as a function of the standard activation settling process in response to a given input pattern. Then, in the *plus phase*, the environment is responsible for providing the outcome or target output activations.

The learning rule for all units in the network is given by Eq. (5):

$$\Delta w_{ij} = \epsilon (y_j^+ - y_j^-) x_i^- \quad (5)$$

for a receiving unit with activation y_j and sending unit with activation x_i . The rule for adjusting the bias weights is just the same as for the regular weights, but with the sending unit activation set to 1:

$$\Delta \beta_{ij} = \epsilon (y_j^+ - y_j^-) \quad (6)$$

The difference between the two phases of activation is an indication of the units' contribution to the overall error signal. Bidirectional connectivity allows output error to be communicated to a hidden unit in terms of the difference in its activation states during the plus and minus states ($y_j^+ - y_j^-$).

References

- Adams, J. A. (1987). Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. *Psychological Bulletin*, 101(1), 41–74.
- Almeida, L. B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In M. Caudil & C. Butler (Eds.), *Proceedings of the IEEE First International Conference on Neural Networks* (Vol. 2, pp. 609–618). New York: IEEE.
- Anderson, J. R. (Ed.). (1981). *Cognitive skills and their acquisition*. Hillsdale, New Jersey: Lawrence Erlbaum.
- Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahway, NJ: Erlbaum.
- Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U., & Waldron, E. M. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review*, 105, 442–481.
- Ashby, F. G., Ennis, J. M., & Spiering, B. J. (2007). A neurobiological theory of automaticity in perceptual categorization. *Psychological Review*, 114(3), 632–656.
- Ashby, F. G., & Valentin, V. V. (2005). *Multiple systems of perceptual category learning: Theory and cognitive tests*. Elsevier.
- Bapi, R. S., Doya, K., & Harner, A. M. (2000). Evidence for effector independent representations and their differential time course of acquisition during motor sequence learning. *Experimental Brain Research*, 132(4), 149–162.
- Beilock, S. L., Bertenthal, B. L., McCoy, A. M., & Carr, T. H. (2004). 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.
- Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, 108(3), 624–652.
- Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, 111(2), 395–429.

- Braver, T. S., & Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In S. Monsell & J. Driver (Eds.), *Attention and performance XVII* (pp. 713–736). Cambridge, MA: MIT Press.
- Byrne, M. D., & Anderson, J. A. (1997). Enhancing ACT-R's perceptual-motor abilities. In M. G. Shafto, & P. Langley (Eds.), *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (p. 880). Stanford, CA: Erlbaum.
- Cohen, J. D., Braver, T. S., & O'Reilly, R. C. (1996). A computational approach to prefrontal cortex, cognitive control and schizophrenia: Recent developments and current challenges. *Philosophical Transactions of the Royal Society of London B*, *351*, 1515–1527.
- Desmurget, M., & Turner, R. S. (2010). Motor sequences and the basal ganglia: Kinematics, not habits. *Journal of Neuroscience*, *30*(22), 7685–7690.
- Gupta, A., & Noelle, D. C. (2005a). Neurocomputational mechanisms for generalization during the sequential learning of multiple tasks. In B. Prasad (Ed.), *Proceedings of the 2nd Indian international conference on artificial intelligence* (pp. 2699–2711).
- Gupta, A., & Noelle, D. C. (2005b). The role of neurocomputational principles in skill savings. In *Proceedings of the 27th annual conference of the cognitive science society* (pp. 863–868).
- Gupta, A., & Noelle, D. C. (2007). A dual-pathway neural network model of control relinquishment in motor skill learning. In M. M. Velosos (Ed.), *Proceedings of the 2007 International Joint Conference on Artificial Intelligence (IJCAI-2007)*, AAAI Press, Menlo Park, California. (pp. 405–410). Hyderabad, India.
- Guthrie, E.R., 1935. *The psychology of learning*. New York: Harper.
- Hardy, L., Mullen, R., & Jones, G. (2004). Knowledge and conscious control of motor action under stress. *British Journal of Psychology*, *87*, 621–636.
- Hikosaka, O., Nakamura, K., Sakai, K., & Nakahara, H. (2002). Central mechanisms of motor skill learning. *Current Opinion in Neurobiology*, *12*, 217–222.
- Hinton, G. E., & McClelland, J. L. (1988). Learning representations by recirculation. In D. Z. Anderson (Ed.), *Neural information processing systems 1987* (pp. 358–366). New York: American Institute of Physics.
- Holt, B. J., & Rainey, S. J. (2002). *An overview of automaticity and implications for training and thinking process*. Tech. Rep. 1790, U.S. Army Research Institute for the Behavioral and Social Sciences.
- Houston, J. P. (1991). *Fundamentals of learning and memory*. New York: Harcourt.
- Huey, E. G. (1908). *The psychology and pedagogy of reading*. Cambridge, MA: MIT Press.
- James, W. (1983). *The principles of psychology*. Cambridge, MA: Harvard University Press.
- Kriete, T., & Noelle, D. C. (2011). Generalisation benefits of output gating in a model of prefrontal cortex. *Connection Science*, *23*(2), 119–129.
- Laird, J. E. (2012). *The soar cognitive architecture*. Cambridge, MA: MIT Press.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, *33*(1), 1–64.
- Masters, R. S. W. (1992). Knowledge, nerves and know how. *British Journal of Psychology*, *83*, 343–358.
- Nakahara, H., Doya, K., & Hikosaka, O. (2001). Parallel cortico-basal ganglia mechanisms for acquisition and execution of visuomotor sequences — A computational approach. *Journal of Cognitive Neuroscience*, *13*(5), 626–647.
- Neumann, O. (1984). *Automatic processing: A review of results and a plea for an old theory*. Verlag: Springer.
- Newell, A. (1990). *Unified theories of cognition*. Cambridge: Harvard University Press.
- O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, *8*, 895–938.
- O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience*. Cambridge, Massachusetts: MIT Press.
- O'Reilly, R. C., Noelle, D. C., Braver, T. S., & Cohen, J. D. (2002). Prefrontal cortex and dynamic categorization tasks: Representational organization and neuromodulatory control. *Cerebral Cortex*, *12*(3), 246–257.
- Pineda, F. J. (1989). Recurrent backpropagation and the dynamical approach to adaptive neural computation. *Neural Computation*, *1*(2), 161–172.
- Rand, M. K., Hikosaka, O., Miyachi, S., Lu, X., Nakamura, K., Kitaguchi, K., et al (2001). Characteristics of sequential movements during early learning period in monkeys. *Experimental Brain Research*, *131*, 283–304.
- Rougier, N. P., Noelle, D. C., S., B. T., Cohen, J. D., & O'Reilly, R. C. (2005). Prefrontal cortex and the flexibility of cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences*, *102*(20), 7338–7343.
- Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, *11*(4), 1317–1329.
- Schaal, S., & Atkeson, C. G. (2010). Learning control in robotics: Trajectory-based optimal control techniques. *IEEE Robotics and Automation Magazine*.
- Schmidt, R. A. (1975). A schema theory of discrete motor-skill learning. *Psychological Review*, *82*, 225–260.
- Schneider, W., & Schrifflin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search and attention. *Psychological Review*, *84*(1), 1–66.
- Shah, A., & Barto, A. G. (2009). Effect on movement selection of an evolving sensory representation: A multiple controller model of skill acquisition. *Brain Research*, *1299*, 55–73.
- Taatgen, N. A., & Lee, F. J. (2003). Production compilation: A simple mechanism to model complex skill acquisition. *Human Factors*, *45*(1), 61–76.
- Turner, R. S., & Desmurget, M. (2010). Basal ganglia contributions to motor control: A vigorous tutor. *Current Opinion in Neurobiology*, *20*(6), 704–716.
- Wolpert, D. M., Ghahramani, Z., & Flanagan, J. R. (2001). Perspectives and problems in motor learning. *Trends in Cognitive Sciences*, *5*(11), 487–494.
- Wolpert, D. M., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.
- Wu, T., Chan, P., & Hallett, M. (2008). Modifications of the interactions in the motor networks when a movement becomes automatic. *Journal of Physiology*, *22*, 4295–4304.
- Wu, T., Kansaku, K., & Hallett, M. (2004). Modifications of the interactions in the motor networks when a movement becomes automatic. *Journal of Neurophysiology*, *91*, 1690–1698.