Adaptive neural models of queuing and timing in fluent action

Daniel Bullock

Cognitive & Neural Systems Department, Boston University, 677 Beacon Street, Boston, MA 02215, USA

In biological cognition, specialized representations and associated control processes solve the temporal problems inherent in skilled action. Recent data and neural circuit models highlight three distinct levels of temporal structure: sequence preparation, velocity scaling, and state-sensitive timing. Short sequences of actions are prepared collectively in prefrontal cortex, then queued for performance by a cyclic competitive process that operates on a parallel analog representation. Successful acts like ball-catch depend on coordinated scaling of effector velocities, and velocity scaling, mediated by the basal ganglia, may be coupled to perceived time-to-contact. Making acts accurate at high speeds requires state-sensitive and precisely timed activations of muscle forces in patterns that accelerate and decelerate the effectors. The cerebellum may provide a maximally efficient representational basis for learning to generate such timed activation patterns.

An important strategy in cognitive science is to choose the representation and associated algorithms/heuristics that are best suited to rapid solution of a problem type. A cognitive system that solves many types of problems needs many kinds of representations. In cognitive neuroscience, there is a consensus that the brain uses many types of representations and associated heuristics. Brain theorists seek to characterize these types and explain how they cooperate to solve animals' cognitive problems.

Three levels of temporal structure in skilled performance

A surprisingly demanding problem is the genesis of skilled behavior using complex effectors like a human’s arm or speech articulators. Skilled behavior emerges in temporally structured episodes, and brain areas that use distinct representations contribute to this temporal structuring. This review examines computational models of neural circuits contributing to three levels of temporal structure in behavior. Level one is the fluent succession of acts prepared collectively as a sequence. This feature is noticeable during breakdowns, for example, stuttering, in which familiar sequences fail to emerge fluently. Level two is coordination of rates across parallel processes, such as planned joint rotations contributing to a reach. To catch a ball, reach dynamics are coordinated with event dynamics: the global rate, and thus the duration, of an interceptive reach is scaled to the ball’s approach time.

Such scaling is compromised in basal ganglia disease. Level three is timed anticipatory responses, such as the braking contractions that our muscles generate to preempt movement ‘overshoots’. Loss of such responses in cerebellar patients severely degrades movements. What neural representations and processes enable fluent succession of acts, act-event coordination, and timed anticipatory responses? And what is the proper analysis of particular cases? In reaching to grasp, the hand opens to an aperture larger than the object and then closes, just as the reach completes. Does this require explicit preparation of a sequence of apertures in prefrontal cortex (level one), or may the sequence emerge at lower levels?

Fluent succession of acts via competitive queuing

Fifty years ago, Lashley [1] used data on sequencing errors – in which early and later elements of a sequence mistakenly exchange positions – to infer that neural representations for all elements of a planned sequence are simultaneously active before sequence production. The proposal that sequences are represented by simultaneous parallel activation of representations of their elements differs from many classical and contemporary proposals. In most recurrent-state network models [2–4], representations of all the elements of a well-learned sequence are never simultaneously active. Instead, the generating system traverses a series of context-states, each of which activates a representation of just the current element. This transiently active representation guides that element’s performance and updates the state representation to create the distinctive context needed to recall the next element. The sequence and the sequence representation are emergent and serial, not parallel as Lashley proposed.

In many cases examined by Lashley, for example, typing familiar words, the elements of a sequence are few, routine, and knowable in advance. Thus, parallel activation is feasible. Some neural models that take this approach exemplify Lashley’s inference that links between successive elements are unnecessary in a sequence representation. Grossberg [5] constructed the first of this class of neural network models (see Fig. 1), now often called ‘competitive queuing’ (CQ) models [6,7].

Neurophysiological evidence for the CQ model

Until 2002 there was no compelling electrophysiological evidence that the brain used the parallel sequence code and iterative choice cycle postulated by CQ theorists. New
Figure 1. A competitive queuing (CQ) network and a comparison of simulated CQ dynamics with cellular data from area 46 of prefrontal cortex. (a) All CQ models have at least two layers, a parallel planning layer and a competitive choice layer. The planning layer contains nodes representing possible sequence elements, such as letters of the alphabet A-Z. To plan a sequence, a desired subset of these nodes is activated in parallel (e.g. nodes representing letters that spell the Australian greeting ‘GIDAY’) and the relative degree of activation (signaled by the relative heights of bars placed above the nodes) is used to specify the relative priority of performance. At the onset of a gating signal, the active representations begin to compete for output via the choice layer. If the competition is fair, the most active plan-layer node will win the competition and thereby generate a corresponding output from the choice layer, which initiates the action. A second effect of this output, mediated by an inhibitory pathway from each output node to its corresponding plan-layer node, is deletion of activity at whatever plan-layer node has just won. Iteration of this choose–perform–delete cycle assures that an element’s initial relative activation level in the planning layer implicitly codes its relative priority in the forthcoming sequence, and that after the iteration, the plan layer will be empty, and ready for preparation of further sequences. If nothing interrupts the feedback and iterated choice processes, then production of a planned sequence is very fluid. (b) A simulation of cellular dynamics in the plan layer of a normalized CQ model [10] during production of a 5-letter sequence, such as ‘GIDAY’, or a 5-stroke drawing [8]. Each simulation trace depicts the activation history of one of the sequence element representations (1–5) during the interval from just before initiation of sequence performance to just after production of element 5. These simulation traces correspond remarkably well with empirical observations made a decade later [8] and shown in (c). [However, the data-processing steps [8] used to generate (c) preclude using it directly as evidence for neural normalization.] (c) Each colored data trace represents the relative activation level in area 46 (of a monkey’s prefrontal cortex) of a small neural ensemble that represents one element of a 3-, 4-, or 5-element sequence used to draw a geometric form. (a) Adapted with permission from [7]; (c) Reproduced with permission from [8].
cell recordings by Averbeck et al. [8] plugged that evidential gap. They trained monkeys to draw a copy of a static geometric form using a routine, prescribed stroke sequence. Thus a form cue encoded sequence-representing information from long term memory. Recordings from area 46 of prefrontal cortex showed that before the monkey began the stroke sequence, there existed a parallel representation, as proposed in CQ theory (see Fig. 1b,c). As strokes were produced, deletions occurred as expected from this representation, with the most active representation being deleted first, and so on, until the final stroke was made and its representation deleted. These data buttress the hypothesis that the brain uses parallel activation patterns to represent, plan and control the execution of short sequences. The same results disconfirm the conjecture [9] that monkeys don’t use a ‘collective’ planning strategy.

In some CQ models [5,10–12], the total activity allocated to plans is normalized and redistributed to the remaining representations on each iteration. By the time the last element is chosen, its activity will have grown to a much larger value than it had initially (Fig. 1b). Such normalization (which abets the explanatory successes of a recent production system model [13]) predicts that the average activation of each representation varies inversely with sequence length. Experiments that varied the number of alternative response options in deferred choice tasks [14–16] confirmed this neurophysiological prediction.

**Progress of CQ models in explaining sequencing and timing**

To motivate normalized CQ models, Grossberg [5] stressed that neurons exhibit finite activation ranges and noise. Both constrain the ability of neurons to use relative activations to reliably code the relative priority of a large number of sequence elements. Brains using this analog code should exhibit a small upper bound on the number of elements that can be reliably recalled in correct sequential order without secondary strategies, such as reloading chunks from long-term memory [5,6,12]. Cowan [17] showed that such a small upper bound has been found for working memory capacity in tasks requiring immediate recall of novel sequences in correct order. Page and Norris [18] showed that a CQ model with noisy choice predicts additional data from immediate serial recall tasks, such as the overwhelming tendency for exchange errors to be transpositions of adjacent elements in the planned sequence.

In all CQ models, the latency to produce a sequence element depends on the time needed for the activation level of the corresponding plan to win the competition. Because more simultaneously active plans imply a lower activation level for each (normalization), the latency to initiate the first element should increase with sequence length. Such effects, measured by Sternberg and colleagues [19], were successfully simulated [10] in 1991. However, more recent data [20,21] showed that for well-practiced sequences, some practice-dependent process overcomes the latency penalty initially associated with preparing sequences. Neurophysiological data [22] implicate the lateral cerebellum in recall of well-learned sequences, and recent neuroanatomical data [23] show projections to prefrontal cortical area 46 from the cerebellum’s dentate nucleus. Consistent with these data, Rhodes and Bullock [12] constructed an adaptive neural model to explain how practice-dependent cerebellar outputs could mediate ‘parallel analog loading’ of sequence element activations into the plan layer of a working memory that obeys CQ principles (Fig. 1a). This model shows how a cortico-cerebello-cortical circuit can learn and recall long-term (procedural) sequence memories. After practice, strong cerebellar output forces the choice layer of the CQ system to pre-commit to the first sequence element. Thereafter, initiation latency is independent of the number of elements in the practiced sequence.

The past decade has seen many extensions of CQ-compatible sequence learning and control models to cognitive phenomena. These include sequences with repeating elements [6,11,24], sequences with overlapping performance of successive elements, as in speech coarticulation [6] and cursive handwriting [25,26], pitch-duration sequences in melody learning [27], and language production [28,29]. Notably, Ward’s connectionist language generator illustrates how to link CQ theory to ‘construction grammar’ [30]. If recent neurophysiological evidence [8,14–16] for CQ assumptions and predictions survives scrutiny, many further CQ model elaborations can be expected.

**Coordination of rates and completion times in voluntary action**

Many movement models, such as Equilibrium Point (EP) models ([31], Box 1), treat the temporal structure of actions from a biomechanical perspective. By contrast, some central pattern generation models, such as Vector Integration To Endpoint (VITE) models ([32], Box 1), treat timing from a cognitive dynamics perspective, with a focus on voluntary gating of plan execution and voluntary control of movement rates. VITE models have successfully simulated both the discharge patterns of diverse motor cortical cell types [32,39] and numerous movement timing phenomena [12,25,26,39,40]. Recently, VITE-consistent models have explained timing properties of interceptive reaching and reach-grasp coordination. Peper and colleagues [41] noted that if an interceptive reach is to succeed at getting the hand to an approaching ball before it passes by, then global scaling of the reaching rate should be coupled to a relational percept. In their Required Velocity (RV) model [41], an evolving perceptual variable, the ball’s declining time-to-contact (TC) with the catcher, continuously adjusts the reaching rate to that velocity required for a successful interception. Because TC increases during the reach, it can function in the same way as the increasing GO signal in VITE models [32]. Comparing predictions of the VITE and RV models with data revealed a common weakness [42], which was corrected by allowing ball velocity to affect both rate (via TC) and hand direction. The resulting ‘prospective’ VITE model of Dessing and colleagues [42] explains many interceptive reaching phenomena (Fig. 2a). Evidence for
Some neurobiologists once entertained the hypothesis that actions might not require any internal trajectory planning, because movements might be treated as mere transitions between postures. If the balance of muscle forces needed to hold a goal posture B were abruptly instated, then the body’s existing posture, A, would suddenly be in a biomechanical disequilibrium created by muscles acting as spring-like force generators. Movement would ensue, with the body’s configuration attracted towards goal posture B as a mechanical equilibrium point in the space of possible body configurations. So emerged ‘equilibrium point’ (EP) models. Simple versions of EP theory fall to logical counterexamples. Suppose a kneeling quadripped hoping to stand were to abruptly activate its muscles to the levels needed to generate the pattern of forces typically used to hold standing posture. This fails because standing quadrupeds support most of their weight by columns of bone – not by muscle forces. Elaborated EP models have failed empirical tests [31], and none explains overwhelming evidence (e.g. [15,16,32]) for continuous movement vector computations in motor cortex. By contrast, such evidence inspired central pattern generator (CPG) models such as the ‘Vector Integration To Endpoint’ (VITE) model [32]. A CPG model is also preferable to sensory feedback control models, because simple voluntary movements in primates survive removal of all sensory feedback regarding the

TC neurons is abundant [42–44], and TC is prominent in other timing models, including models of viapoint movements [44] and legato articulation [45,46] by pianists.

Another focus of timing research is reach-grasp coordination. The Hoff–Arbib model [47] showed how internal duration computations could ensure that hand opening and closing were adjusted to both the planned duration of a reach and the maximum expected hand aperture (the thumb to finger distance), which was solely a function of object size. Yet reach duration often emerges from a dynamic coupling between actor and object motions [41,42], and maximum hand aperture depends on reach rate [48]. Thus neither component durations nor maximum apertures are known in advance. The VITE-based model (Fig. 2b) of Ulloa and Bullock [49] generates reach-grasp coordination without such advance knowledge. Key timing-data trends [50,51] emerge dynamically. Thus, although there is a single pre-planned hand aperture goal (object size), a cross-coupling between the reach and grasp circuits allows the aperture to be transiently incremented during the reach by an amount proportional to reach velocity, which in VITE trajectories is maximal midway through the duration of a typical (unperturbed) reach. Because the transient increment to aperture fades after mid-time of the reach, the model generates an aperture sequence without explicit sequence preparation. Under typical (and perturbed, see Fig. 2b) conditions, the reach and hand closure finish synchronously. This exemplifies the temporal equifinality property (Box 1 and Fig. 2b) afforded by using a single rate-scaling signal to coordinate parallel processes.

To students of control theory, VITE circuits (Fig. 2) invite comparisons with engineering approaches such as PID (Proportional, Integral, Derivative) control, in which motor command components are proportional to the feedback error (between the desired and current sensory states), its integral, and its time derivative. However, whereas PID control requires sensory feedback, a central pattern generator (CPG) like VITE can operate without sensory feedback. Our analysis [39] concluded that primates use flexible CPGs of desired kinematics, which operate in cascade with a spinal PID controller and a cerebellar feedforward controller. The VITE circuit models only some properties of the inferred CPG. Equally important as voluntary rate scaling to the success of coordinated movements is flexible recruitment and de-recruitment of the multiple effectors that contribute motion components in different directions and amplitudes as an act unfolds. Guenther and colleagues [52,53] have developed adaptive neural models called DIRECT and DIVA to explain effector recruitment in arm and speech movements, respectively. Fiala [54] and Barreca and Guenther [53] each demonstrated how to preserve principles of the VITE model within complex, DIRECT-type, models. In DIRECT, learning creates inverse kinematic mappings and predictive forward kinematic mappings that transform between sensory and motor coordinates. Such networks enable ‘motor-equivalence’, i.e. equivalent results from variable means, because as the movement unfolds, the learned internal mappings recruit novel combinations of joint rotations adequate to produce desired visual motions of a hand, or a tool held in the hand. The DIRECT model clarifies how adaptive circuits can achieve performances similar to abstract, but behaviorally predictive, models like the ‘optimal feedback’ model [55] of Todorov and Jordan. In particular, DIRECT and DIVA respectively enable on-line corrections to unexpected perturbations of movements by arms and speech articulators.

**Timed anticipatory responses**

In a successful ball catch, the arm flicks out and ‘stops on a dime’ at whatever degree of arm extension enables the hand to catch the ball. Newtonian mechanics implies that an arm set in motion by extensor muscles would (disastrously) continue ‘past the mark’ unless braked by precisely timed, anticipatory action of opposing muscles. When driving a car, stomping the accelerator and hitting
the brake are separate voluntary actions. When ‘driving’ our bodies, the braking contractions are automatic and of subcortical origin. The timed anticipatory components of the motor cortex signals that shape braking reactions disappear if part of the cerebellum’s dentate nucleus is cooled [56]. Many such observations implicate the cerebellum as an engine for the learning and performance of anticipatory responses [56–59] that fit the following formula: In context C generate signals in output channel A after waiting time interval T. Furthermore, Perrett, Ruiz and Mauk [60] showed that a restricted cerebellar cortical lesion, which spares the cerebellum’s deep nuclei, produces an animal that makes its anticipatory responses too early. Such experiments have created a consensus that some mechanism in the cerebellar cortex learns to withhold an anticipatory response until the optimal time T after onset of the conditional stimulus (CS) that indicates the state/context within which the response should be generated.

The network and cells in the cerebellar cortex are complex, and several models have been proposed to explain cerebellar adaptive timing. These models fell into two broad classes: network-delay models and synaptic-delay models (see Box 2). Two similar cerebellar network-delay models [61,62] were proposed independently in 1994 and both borrowed ideas from earlier adaptive timing models (e.g. [63]). Both postulated that a CS (conditional stimulus) carried to the cerebellum by mossy-fibers induces local interactions between Golgi cells and large populations of granule cells. Such interactions might enable any CS to generate a spectrum of transient, time-lagged granule cell activations that was specific to that CS. Given such a temporal basis, whichever granule cell activation has the appropriate time delay to coincide with a climbing fiber signal can, over the course of repeated experiential trials, become able to control cerebellar output, provided that an associative learning process operates to change the synaptic weight between that granule cell’s parallel fiber and those Purkinje (output) cells excited by the climbing fiber signal. (Signal propagation along parallel fibers does not provide a behaviorally significant delay.) Even the latest network-delay models [64,65] use whole cells and network interactions to create the temporal basis. By contrast, synaptic-delay models [66] postulate that the spectrum of delays needed for adaptive timing emerge in synapse-specific elements, namely the tiny spines [66,67] that populate the branchlets of Purkinje cells’ dendritic trees. It is with these spines that parallel fibers actually synapse.

Relative to synaptic-delay models, network-delay models are very inefficient. They are metabolically inefficient because they use a cellular network to do what may be done within tiny dendritic spines; and they are computationally inefficient because each granule cell population dedicated to creating a spectrum of time-lagged responses to one CS cannot readily serve other roles attributed to cerebellar cortex. In particular, no network-delay model has shown how purported temporal-basis granule cells could simultaneously fulfill the ‘spatial pattern separation’ role (Box 2) attributed to the granule cell stage in Marr–Albus models [68–70]. By contrast, all

---

The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). The inverse, DVV by computing a relative velocity vector (RVV). The inverse, TC–1(t), an ‘urgency to complete’ signal, refines the effective GO signal. These interactions adjust movement velocities to values needed to intercept in the time available. This VITE model is called ‘prospective’ rather than ‘predictive’ because it anticipates without ever predicting the locus of interception. If a model were to predict and move by the shortest path to the locus of interception, it could not explain the systematic reversals of movement direction exhibited by human catchers. (b) A neural model [49] for temporal coordination of three component processes – hand orientation control, control of hand transport by the arm, and hand aperture (distance between thumb and index finger) control – that operate in parallel during reaches that culminate in a precision grasp of an object at rest. The symbols T, D, P, and V stand for internally computed target, difference, and present position vectors, respectively; the Vt stand for desired velocity vectors. The values of the subscript i indicate either transport (T), aperture (A), or orientation (O), for example, T is the target object’s location, Ta is targeted hand aperture, and To is targeted object’s location, TA is targeted hand aperture, and TO is targeted object (e.g. a ball). In all VITE models, DV is the difference vector computed between representations of the target position vector (TPV) and the hand’s present position vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continuously specifies desired hand position. The desired velocity vector (DVV) for the hand arises as a product of the DV and an output vector (PPV). Output from the PPV continues...
variants [4,71] of the synaptic-delay model introduced by Fiala and colleagues [66] are compatible with granule-cell axons (as shown) is a striking connectional asymmetry that has figured prominently in most theories of cerebellar learning, including adaptive timing. (A) and (B) reprinted with permission from [4].

Although further computational challenges to some network-delay models exist – for example, poor signal-processing repeatability – recent empirical results cast strong doubt on the sufficiency of any cerebellar network-delay model. Three separate laboratories [57,72,73] have discovered that cerebellar adaptive timing occurs under conditions in which it should be impossible if adaptive timing requires the network-delay mechanism. By contrast, the synaptic-delay model correctly predicts successful adaptive timing in all three cases. In the earliest experiment, Shinkman, Swain and Thompson [57] substituted prolonged direct stimulation of parallel fibers (granule cell axons) for CS presentation, and demonstrated normal delayed response learning and performance. For this protocol, the network-delay model instead predicts a learned response that begins much too early and persists until CS offset. The second experiment, by Raymond and Lisberger [72], was explicitly designed to pit network-delay models against synaptic-delay models. They found that the information needed to control behavior had disappeared from the granule cell discharges ~100 ms before the signal that initiated associative learning. Under such conditions, the network-delay model predicts no response learning – again, contrary to the data. Raymond and Lisberger inferred that there must be a synapse-specific delay of at least 100 ms – an inference since confirmed by direct observations on Purkinje cell dendritic spines [67]. In the third experiment, Svensson and Ivarsson [73] used decerebrate ferrets to eliminate any hippocampal or other cerebral contributions to adaptive timing. They then showed successful temporal conditioning using a ‘trace’ protocol, in which there is a gap between CS offset and the time the conditioned response should be generated. For this protocol, the network-delay model predicts no learning, because there is no stimulus to drive granule cell activations once the CS goes off. By contrast, synaptic delay models predict robust learning, provided that the trace interval is not too long. Available data suggest that

If the cerebellum performs both of the major functions most commonly attributed to it – spatial pattern separation and adaptive timing – then cerebellar learning constitutes a parallel search, through masses of potentially predictive recent signal sets, for reliable leading indicators that will allow the animal to make timed anticipatory responses. According to Marr–Albus theory [68–70], the first stage of cerebellar processing uses the granule cells (Figure 1a,b) to perform spatial pattern separation. Each of millions of mossy fibers (MFs) distributes its signal, one component of a massive state/context vector, across the cerebellar cortex (Figure 1a). Billions of cerebellar granule cells each use four or five dendrites to sample a partially distinctive subset of the MF context representation (Figure 1a). This allows granule cells to detect highly specific event or state-subset combinations. The creation of these combination representations, abetted by inhibitory suppression of granule cells excited by only one or two of their potential inputs, makes the representations of any two contexts more dissimilar (‘farther apart’ whence ‘separation’) in the higher-dimensional space defined by granule cell outputs than in the lower-dimensional space defined by MF signals. Each granule cell’s parallel fiber (Figure 1a,b) sends its output (if any) to hundreds of Purkinje cells, and each Purkinje cell receives inputs from many thousands of parallel fibers. Synaptic-delay models propose that adaptive timing is achieved after the granule stage, by a spectrum of delays within the population of Purkinje dendritic spines contacted by each granule via its long parallel fiber. Thus spatial pattern separation and adaptive timing can efficiently co-exist and co-operate in each cerebellar compartment. By contrast, network-delay models propose that the spectrum of delays emerges in the population of granule cells itself. It remains to be shown in a model how the granule stage could perform both functions.

Figure 1. Circuit properties of the mammalian cerebellum. (a) Within the lattice-like cortex (‘outer rind’) of the cerebellum (‘little brain’), signals pass along mossy fibers to mossy fiber terminals (rosettes), then to granule cells, then to granule cell axons, which rise and then bifurcate (T-branch) to form parallel fibers. For many mammals, there are as many granule cells in the cerebellum as there are neurons in the remainder of the brain. Each branch of each parallel fiber can excite up to hundreds of Golgi, basket, stellate and Purkinje cells, groups of which are linked in a highly conserved connectivity pattern, one token of which is shown in (b). (b) A very highly schematic depiction of the cell types and connections found in the basic cerebellar circuit, which is replicated millions of times, albeit with different inputs and output destinations, in each cerebellum of large mammals. Also shown is the pons, source of a large proportion of mossy fibers, and the inferior olive, sole source of the climbing fibers whose discharges gate one type of cerebellar learning. The convergence onto each Purkinje cell of thousands of parallel fibers (only one shown here) but just one climbing fiber (as shown) is a striking connectional asymmetry that has figured prominently in most theories of cerebellar learning, including adaptive timing. (A) and (B) reprinted with permission from [4].

Box 2. A two-stage cerebellar adaptive engine?
the cerebellum is sufficient for trace intervals up to at least 600 ms [73,74] Learning across longer trace intervals requires hippocampal assistance during training. Thus all three experimental results are consistent with the synaptic-delay model, but inconsistent with the network-delay model.

How efficient should a biologically accurate model of adaptive timing be? The synaptic-delay model proposed by Fiala et al. [66] is much more efficient than the network-delay model (Box 2), but it is not maximally efficient. Two offspring [4,71] of the synaptic-delay model have proposed distinct ways to replace each of the original model's populations of dendritic spines (and associated pre-existing spectra of delays) with a single adaptive element. In these untested models, a granule cell's output is transformed into a separate, experience-trained time delay within each Purkinje dendritic spine contacted.

Conclusions

This review has focused on neural circuit models of fluent performance of discrete actions, and the implicit claim was that at least three kinds of temporal structuring must be acknowledged to exist as distinct factors in most episodes of skilled action. Competitive queuing theorists who endorsed Lashley's inference that some sequence planning involves parallel activation of all sequence elements can now point to compelling electrophysiological support, but this does not imply that other mechanisms play no role in sequence representations important for fluent action. Vector-integration-to-endpoint models explain many electrophysiological and behavioral patterns as signatures of a nexus of circuit adaptations that made flexible voluntary action possible, but many ideas from the literatures on equilibrium point control and feedback control are usefully incorporated in extensions of CPG models like VITE. Synaptic-delay data and models indicate that the brain has an efficient intracellular mechanism for detecting predictive relationships between event onsets separated by intervals from tens of milliseconds to at least half a second, but this may say little about interval timing on significantly longer time scales [75]. Finally, it is exciting to see emerging data and computer models that highlight the embedding of subcortical circuits, in the basal ganglia and cerebellum, within segregated loops that include the frontal cortex [23,36,37]. These subcortical circuits, heretofore often associated with operant conditioning (basal ganglia) and classical conditioning (cerebellum), have emerged as key to cognitive functions such as decision making (e.g. [38]) and serial plan preparation (e.g. [12]). Clarifying the functional architecture of these extended circuits is a priority of most brain modelers interested in the temporal structure of behavior.

Acknowledgements

Preparation of this article was partially supported by NIH R01 DC02852.

References

Hikosaka, O. et al. (2000) Role of the basal ganglia in the control of purposive saccadic eye movements. Physiol. Rev. 80, 953–978
Shinkman, P.G. et al. (1996) Classical conditioning with electrical stimulation of the cerebellum as both conditioned and unconditioned stimulus. Behav. Neurosci. 110, 914–921