

LETTER

Compensating for Neural Transmission Delay Using Extrapolatory Neural Activation in Evolutionary Neural Networks

Heejin Lim and Yoonsuck Choe

Department of Computer Science, Texas A&M University
3112 TAMU, College Station, TX 77843-3112, USA
Email: hylim@cs.tamu.edu, choe@tamu.edu

(Submitted on December 20, 2005; Accepted on March 20, 2006)

Abstract — In an environment that is temporal as well as spatial in nature, the nervous system of agents needs to deal with various forms of delay, internal or external. Neural (or internal) delay can cause serious problems because by the time the central nervous system receives an input from the periphery, the environmental state is already updated. To be in touch with reality in the present rather than in the past, such a delay has to be compensated. Our observation is that facilitatory dynamics found in synapses can effectively deal with delay by activating in an extrapolatory mode. The idea was tested in a modified 2D pole-balancing problem which included sensory delays. Within this domain, we tested the behavior of recurrent neural networks with facilitatory neural dynamics trained via neuroevolution. Analysis of the performance and the evolved network parameters showed that, under various forms of delay, networks utilizing facilitatory dynamics are at a significant competitive advantage compared to networks with other dynamics.

Keywords — Neural delay, delay compensation, extrapolation, pole balancing, evolutionary neural networks

1. Introduction

Delay is an unavoidable problem for a living organism, which has physical limits in the speed of signal transmission within its system. Such a delay can cause serious problems as shown in Fig. 1. During the time a signal travels from a peripheral sensor (such as the photoreceptor) to the central nervous system (e.g. the visual cortex), a moving object in the environment can cover a significant distance which can lead to critical errors in the motor output based on that input. For example, the neural latency from visual stimulus onset to the motor output can be no less than 100 ms up to several hundred milliseconds [1, 2, 3]: An object moving at 40 mph can cover about 9 m in 500 ms (Fig. 1b).

However, the problem can be overcome if the central nervous system can take into account the neural transmission delay (Δt) and generate action based on the estimated current state $S(t + \Delta t)$ rather than that at its periphery at time t ($S(t)$, Fig. 1c). Such a compensatory mechanism can be built into a system at birth, but such a fixed solution is not feasible because the organism grows in size during development, resulting in increased delay. For example, consider that the axons are stretched to become longer during growth. How can the developing nervous system cope with such a problem? This is the main question investigated in this paper.

Psychophysical experiments such as flash-lag effect showed that extrapolation can take place in the nervous system. In visual flash-lag effect, the position of a moving object is perceived to be ahead of a briefly flashed object when they are physically co-localized at the time of the flash [4, 5, 6, 7, 8]. One interesting hypothesis arising from flash-lag effect is that of motion extrapolation: Extrapolation of state information over time can compensate for delay, and flash-lag effect may be caused by such a mechanism [5, 9, 10, 11].

According to the motion extrapolation model, a moving object's location is extrapolated so that the perceived location of the object at a given instant is the same as the object's actual location in the environment at that precise

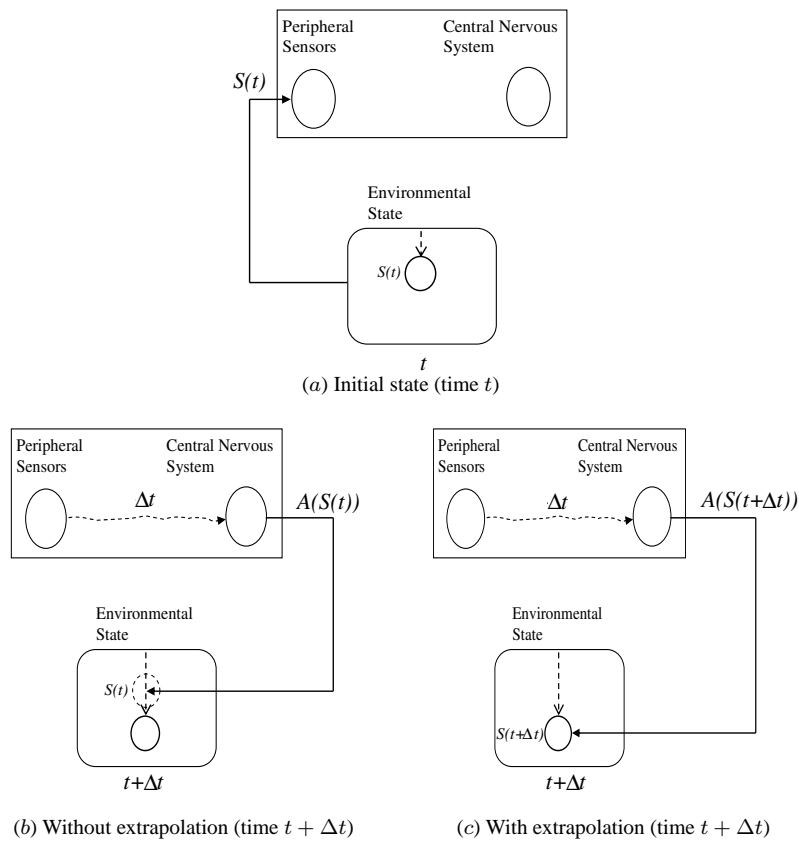


Figure 1. Interaction between agent and environment under internal delay.

moment, despite the delay. However, the abrupt flashing stimulus has no previous history to extrapolate from, thus it is perceived in its fixed location. Thus, flash-lag effect occurs due to such a discrepancy in extrapolation. Furthermore, such an extrapolatory mechanism seems to be ubiquitous in the nervous system. For example, flash-lag effect has been found in motor performance [12], auditory tasks [13], and in several visual modalities such as color, pattern entropy, and luminance [14]. The perceptual effect demonstrated in these experiments indicates that the human nervous system performs extrapolation to align its internal perceptual state with the environmental state in precise synchrony.

However, the motion extrapolation model has some limitations. For example, humans do not perceive displacement between a flashed object and a moving object when the moving object stops or reverses its direction of motion at the time of the flash [7]. Another hypothesis, postdiction, suggests that information received in the future is incorporated into the estimation of a past perceptual event [15, 16]. Postdiction helps explain anomalies near motion reversal or motion termination where extrapolation model cannot provide an accurate account. Despite these limitations, motion extrapolation model has desirable properties [17, 9, 18, 19], and we showed in [20] that the limitation can be overcome.

The question that arises at this point is, how can such an extrapolatory mechanism be implemented in the nervous system? It is possible that recurrently connected networks of neurons in the brain can provide such a function, since recurrence makes available the history of previous activations [21, 22]. However, such historical data alone may not be sufficient to effectively compensate for neural delay since the evolving dynamics of the network may not be fast enough. Our hypothesis is that for fast extrapolation, the mechanism has to be implemented at a single-neuron level.

In this paper, we developed a recurrent neural network model with facilitatory neural dynamics (Facilitating Activity Network, or FAN), where the rate of change in the activity level is used to calculate an extrapolated activity level (Fig. 2a). The network was tested in a modified nonlinear two-degree-of-freedom (2D) pole-balancing problem (Fig. 3) where various forms of input delay were introduced. To test the utility of the facilitatory dy-

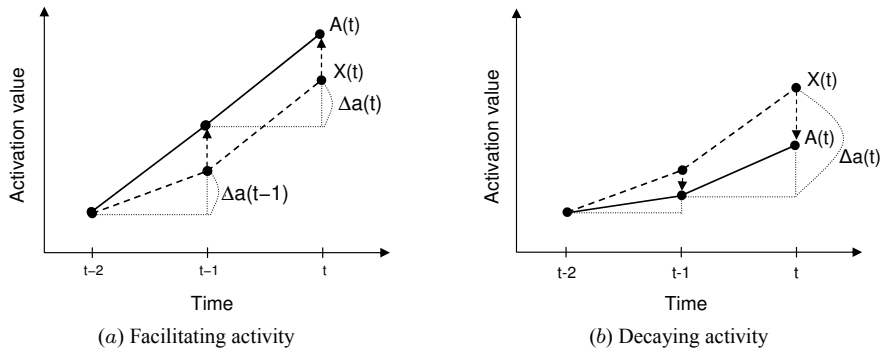


Figure 2. Facilitating or decaying neural activity.

namics, we compared the network against recurrent neural networks without any single neuron dynamics (i.e. the Control) and another with a decaying neural dynamics (Decaying Activity Network, or DAN; see Fig. 2b). The network parameters (connection weight and facilitation/decay rate) were found using the Enforced Subpopulation algorithm (ESP), a neuroevolution algorithm by Gomez and Miikkulainen [23, 24, 25]. This method allowed us to analyze the results in two ways: (1) task performance and (2) degree of utilization of particular network features such as facilitation/decay rate. In all cases, FAN’s performance was the best, and it turned out that high-tness neurons in FAN utilized the facilitation rate parameter in its chromosome more than the low-tness neurons.

Our overall results suggest that neurons with facilitatory activity can effectively compensate for neural delays, thus allowing the central nervous system to be in touch with the environment in the present, not in the past. In the following, rst, the facilitating and decaying neural dynamics will be derived (Sec. 2). Next, the modi ed 2D pole-balancing problem will be outlined (Sec. 3) and the results from the modi ed cart-pole balancing problem will be presented and analyzed (Sec. 4). Finally, discussion and conclusion will be presented (Sec. 5-6).

2. Facilitating and Decaying Neural Dynamics

There are several different ways in which temporal information can be processed in a neural network. For example, decay and delay have been used as learnable parameters in biologically motivated arti cial neural networks for temporal pattern processing [26, 27, 28]. Several computational models also include neural delay and decay as an integral part of their design [29, 30, 31, 32]. However, in these works, the focus was more on utilizing delay for a particular functional purpose such as sound localization [32], rather than recognizing neural transmission delay as a problem to be solved in itself. We introduce delay in the arrival of sensory input to hidden neurons so that each internal neuron generates its activation value based on the outdated input data (i.e., put the network in the same condition as the nervous system, which has neural delay).

Recent neurophysiological experiments have uncovered neural mechanisms that can potentially contribute to delay compensation. Researchers have shown that different dynamics exist at the synapse, as found in depressing or facilitating synapses [33, 34, 35]. In these synapses, the activation level (the membrane potential) of the postsynaptic neuron is not only based on the immediate input at a particular instant but is also dependent on the rate of change in the activation level in the near past. These synapses have been studied to nd the relationship between synaptic dynamics and temporal information processing [36, 37, 38, 39]. However, to our knowledge, these mechanisms have not yet been investigated in relation to delay compensation.

Such a mode of activation found in these experiments is quite different from conventional arti cial neural networks (ANNs) where the activation level of the neurons are solely determined by the instantaneous input and the connection weight values. For example, in conventional ANNs, activation value $X_i(t)$ of a neuron i at time t is de ned as follows:

$$X_i(t) = g \left(\sum_{j \in N_i} w_{ij} X_j(t) \right), \tag{1}$$

where $g(\cdot)$ is a nonlinear activation function (such as the sigmoid function), N_i the set of neurons sending activation to neuron i (the connectivity graph should be free of cycles), and w_{ij} the connection weight from neuron j to neuron

i. As we can see from the equation, the past activation values of X_i are not available, thus the activation value cannot be updated based on the rate of change in X_i . An exception to this is recurrent neural networks where past activation in the network can also have an effect on the present activity [21, 22]. However, in our experimental results, it turns out that such recurrent dynamics alone is not sufficient to effectively counter the effects of delay (see Sec. 4.2 for details).

There are at least two ways in which we can introduce temporal dynamics at a single-neuron level. The activity $X_i(t)$ can be either decayed or facilitated based on its past activation. Let us denote this modified activity as $A_i(t)$ to distinguish it from $X_i(t)$. With this, we can now define the decaying and facilitating dynamics in a continuous-valued neuron (i.e. a ring-rate neuron).

The activity of a neuron with facilitating synapses can be defined as follows (for the convenience of notation, we will drop the index i):

$$A(t) = X(t) + r(X(t) - A(t-1)), \quad (2)$$

where $A(t)$ is the facilitated activation level at time t , $X(t)$ the instantaneous activation solely based on the instantaneous input at time t , and r the facilitation rate ($0 \leq r \leq 1$). The basic idea is that the instantaneous activation $X(t)$ should be augmented with the rate of change $X(t) - A(t-1)$ modulated by facilitation rate r . For later use, we will call this rate of change $\Delta a(t)$:

$$\Delta a(t) = X(t) - A(t-1). \quad (3)$$

Eq. 2 is similar to extrapolation using forward Euler's method where the continuous derivative $A'(\cdot)$ is replaced with its discrete approximation $\Delta a(\cdot)$ [40] (p. 710). Fig. 2a shows how facilitatory activity is derived from the present and past neural activity. Basically, the activation level $A(t)$ at time t (where t coincides with the environmental time) is estimated using the input $X(t - \Delta t)$ that arrived with a delay of Δt . If the facilitatory rate r is close to 0, $A(t)$ reduces to $X(t)$, thus it represents old information compared to the current environmental state. If r is close to 1, maximum extrapolation is achieved. Note that the facilitating equation is different from running average which tends to smooth out the activation level.

A neuron's activity with decaying synapses can be calculated as follows:

$$A(t) = rA(t-1) + (1-r)X(t), \quad (4)$$

where $A(t)$ is the decayed activation level at time t , $X(t)$ the instantaneous activation solely based on the current input at time t , and r the decay rate ($0 \leq r \leq 1$). Thus, if r is close to 0, the equation will reduce to $X(t)$, becoming identical to Eq. 1 as in conventional neural networks. However, if r approaches 1, the activation at time t will be close to $A(t-1)$. This is similar to running average, thus it will have a smoothing effect. It is important to note that the decay rate r , as defined above, represents how much the decay dynamics is utilized, and not how fast previous activity decays over time. Fig. 2b shows an example of decaying activation value when $r = 0.5$. Note that the equation is essentially the same as Eq. 2, since $A(t) = rA(t-1) + (1-r)X(t) = X(t) + r'(X(t) - A(t-1))$, where $r' = -r$. So, both equations can be written as:

$$A(t) = X(t) + r\Delta a(t), \quad (5)$$

where $-1 \leq r \leq 1$, and thus the dynamic activation values in the facilitatory or the decaying neurons falls within the range of $X(t) - \Delta a(t) \leq A(t) \leq X(t) + \Delta a(t)$.

The basic idea behind the facilitating and decaying activity dynamics described above is very simple, but it turns out that such a small change can significantly improve the ability of the neural network in compensating for delay.

3. Experiments

3.1 2D Pole-Balancing Problem with Input Delay

The main domain we tested our idea of extrapolatory neural dynamics was the pole-balancing problem. The pole-balancing problem has been established as a standard benchmark for adaptive control systems because it is a nonlinear problem that is easy to visualize and analyze [41]. In the standard task, a cart is allowed to move along a straight line while trying to keep balanced the pole attached to it. The goal of a controller here is to produce a

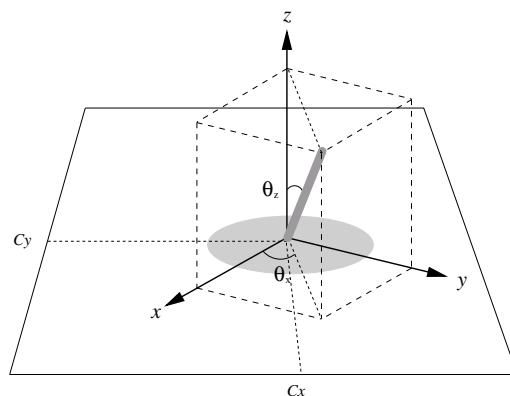


Figure 3. 2D pole-balancing problem.

sequence of force to be applied to the cart to make the pole balanced and to maintain the cart position within a 2D (x and y axes) plane for an interval.

A more difficult task than this is the 2D version, where the cart is allowed to move on a 2D plane (Fig. 3). The state of the cart-environment system at a given instant can be fully described by the cart's location (c_x, c_y) , their derivatives over time (\dot{c}_x, \dot{c}_y) , the configuration of the pole relative to the z and the x axes θ_z and θ_x , and their derivatives over time $\dot{\theta}_z$ and $\dot{\theta}_x$. The standard problem without delay can be solved by feedforward neural networks when such a full state information is available. However, if the derivatives (velocity) are not available (i.e. only c_x, c_y, θ_z , and θ_x are given), a recurrent neural network is needed: The recurrent dynamics of the network can serve as a form of memory from which the velocity information can be recovered.

For our simulations, we made the 2D pole-balancing problem even harder by introducing delay in the four state inputs c_x, c_y, θ_z , and θ_x (without the velocity information) in different combinations and with different durations. The purpose of doing this was to simulate conditions where neural conduction delay existed within a system that is interacting with the environment in real time.

There are two main branches of reinforcement learning methods that effectively solve the pole-balancing problem: methods that (1) search the space of value functions that assess the utility of behaviors (e.g. temporal difference approach [42, 43, 44, 45]) and those that (2) directly search the space of behaviors (e.g. neuroevolution approach [46, 47, 23, 48]). Without explicitly assessing their utility, neuroevolution method directly map observations to actions and gradually adapts individuals' (neurons) genotypes. One effective reinforcement learning method using neuroevolution is the Enforced Subpopulation algorithm (ESP [23, 49]), which showed successful performance in non-Markovian control problems [25, 24]. In ESP, the connection weights in a recurrent neural network are determined through evolutionary learning, and instead of full networks, single neurons are evolved so that best neurons from separate subpopulations can be put together to form a complete network. To test our hypothesis (i.e., for fast extrapolation, extrapolation has to be implemented at a single-neuron level), we used ESP as the basis of our simulations. With ESP, the neurons in each subpopulation could evolve independently, and rapidly specialize into good network sub-functions. With this setup, we can effectively observe how a single neuron, located in a particular position in the network, develops extrapolatory capability. It turns out that two output neurons which generate force to be applied to cart (i.e. f_x and f_y in Fig. 4) greatly evolve facilitation rate while the other neurons do not (cf. Sec. 4.4).

3.2 Experimental setup

For realistic physical modeling with friction, pole balancing behaviors were modeled with fourth-order Runge-Kutta integration. (See [49] for the full equations of motion). We used a recurrent neural network with five neurons to control the pole cart (Fig. 4). The neurons were fully recurrently connected, and all neurons received input from four sources: c_x, c_y, θ_z , and θ_x , as introduced earlier in Sec. 3.1 (Fig 3). Two output neurons generated the force in the x and in the y direction. The optimal values for the configurable parameters in each neuron were found through neuroevolution using the Enforced Subpopulation algorithm (ESP). Each neuron was assigned a chromosome containing the connection weights and optionally the different rate parameters (facilitation rate r_f

or decay rate r_d). The neurons were drawn from five populations, each consisting of forty neurons, to randomly construct a controller network. In each generation, 400 randomly combined networks were evaluated, and the number of generations was limited to 70. After the fitness evaluation, the neurons showing high fitness were mated using crossover, and then mutated (mutation rate = 0.7, i.e., 70% of neurons in subpopulation were mutated) to produce new offspring. The physical parameters for the cart-pole system were as follows: pole length 0.1 m, pole mass 0.02 kg, tracking area 3 m \times 3 m, and applied force limited to the range $[-10, 10]$ N. Due to the difficulty of the delay problem, a short pole length was used. Fitness was determined by the number of time steps a network could keep a pole within a specified angle from the up-right position (15°) while keeping the cart within a square boundary. The task was considered solved if a network could do the above for 10,000 time steps (100 seconds) within 70 generations.

We compared the performance of three different network types (250 experiments each): (1) Facilitating Activity Network (FAN), where facilitation rate r_f was included as an evolvable parameter as well as the standard connection weights; (2) Control, which was the baseline ESP implementation where only the weights were evolvable; and (3) Decaying Activity Network (DAN), where decay rate r_d was evolvable in the same manner as in FAN. To compare fairly the performance of the three networks, we set parameters other than those in the chromosome to be equal (e.g. number of neurons, mutation rate, etc.; see above). All weight values and the facilitation/decay rate parameter values were uniformly randomly initialized between 0.0 and 1.0. Initially, the cart is located in the center of the 2D boundary ($c_x = c_y = 0.0$) and the pole is initialized with a tilt of 0.01 radians ($\theta_z = \theta_x = 0.01$). We tested the three networks (FAN, Control, and DAN), as well as the baseline case without delay, under different internal delay conditions. The results from each experiment are reported in the following section.

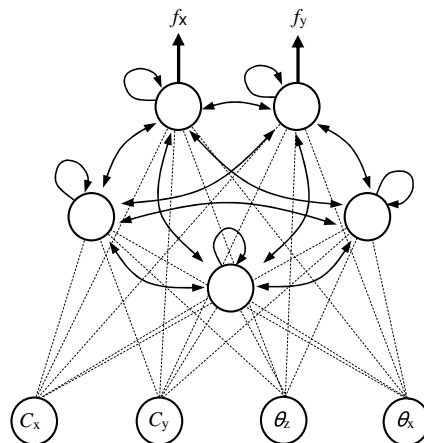


Figure 4. Recurrent neural network for pole balancing.

4. Results

4.1 Neural activation and general behavior

First, we compared the neural activity in the three networks to generally characterize the effects of adding facilitation/decay in the neural dynamics in the network. In these experiments, all four inputs were given with a 1-step (10 ms) delay beginning from 50 and lasting at 150 evaluation steps within each individual trial. (The results were similar for other delay conditions.) Fig. 5 shows the neural activities of an output neuron (top left unit in Fig. 4) from three different networks. The activities of the other output neuron (top right in Fig. 4) showed similar patterns as in Fig. 5 (data are not shown here). The results from all three networks are from successful trials. However, the activity traces are markedly different. DAN produced an on-going, noisy, and high-amplitude oscillation in its neural activity (Fig. 5a). Compared to DAN, the Control showed less noise but the oscillation had a relatively high amplitude (Fig. 5b). FAN, on the other hand, initially showed a large fluctuation (not shown), but quickly settled to a very stable low-amplitude oscillation, and maintained the stability (Fig. 5c). These results suggest that even though extrapolation is generally known to be unstable, if used in a short term, and sparingly, it

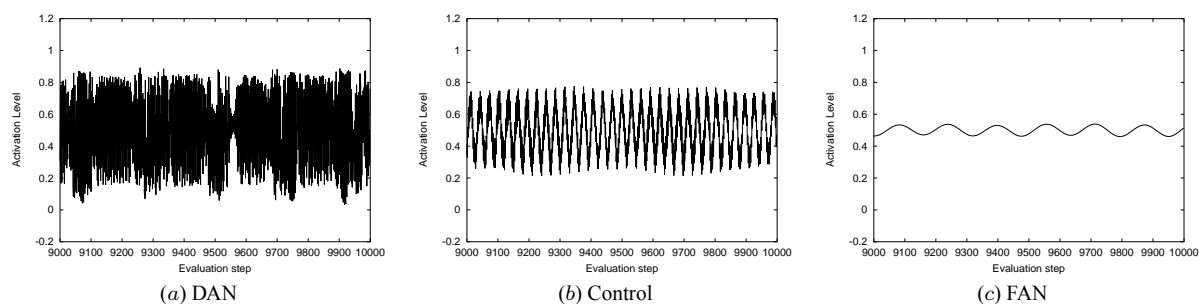


Figure 5. Activation level of output neurons.

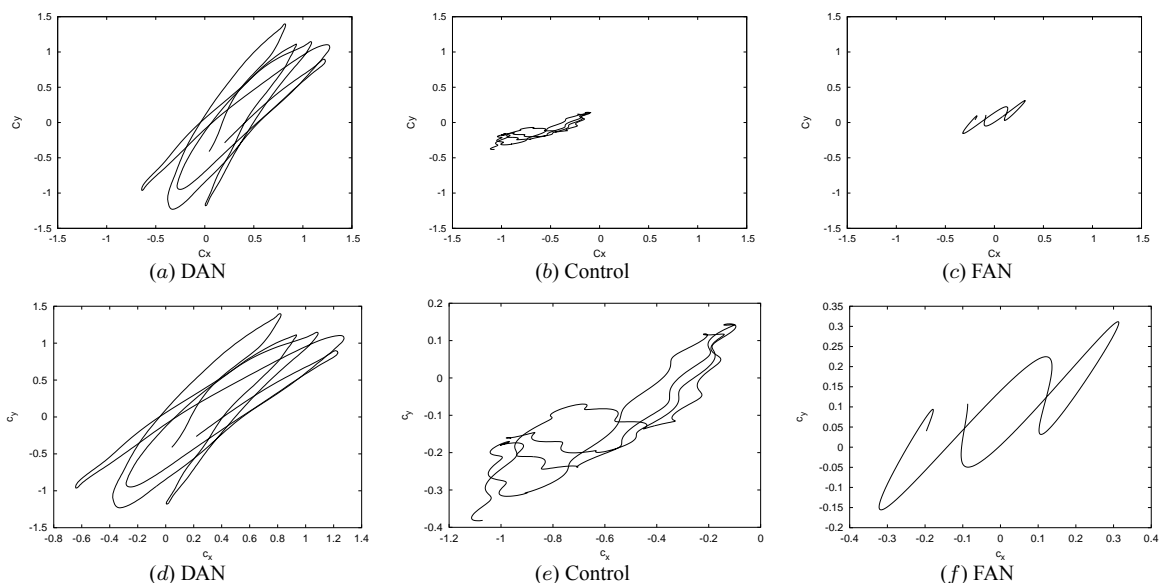


Figure 6. Cart trajectories under input delay condition.

can help faster convergence to a stable state in tasks with delay.

The next step is to compare the overall behavior of the cart under the three different controller networks. The delay condition was the same as above. We traced the cart trajectories to gain insight into how the differing neural dynamics translate into the behavior of the cart (Fig. 6). Note that the relative scale of the x - and the y -axis is same in the three plots (a) to (c) while they are different in the three plots (d) to (f) for a better view of the cart behavior (that is, (a) to (c) show the same data as (d) to (f), at a different scale). The cart trajectory in DAN was erratic and involved large motions, reflecting the noisy high-amplitude oscillation seen in its activation (Fig. 6a, d). The Control on the other hand had a wiggly trajectory (Fig. 6b, e). However, FAN had a trajectory with a very small footprint that was also very smooth (Fig. 6c, f), suggesting that the facilitating dynamics in single neurons contributed to a more accurate control of the cart. Other successful trials showed similar relative difference in behavior (data not shown).

4.2 Performance under different input delay conditions

To test the ability of the three networks in delay compensation, we conducted experiments under different delay conditions: (1) without delay, (2) with a uniform amount of delay for all input sources (c_x , c_y , θ_z , and θ_x) for a fixed, limited period of time during each run, (3) delay in θ_z , and (4) delay in θ_x throughout the entire duration of each run. Fig. 7 summarizes the results under these different experimental conditions.

In experiment 1, the base case, we tested the standard task without any delay. Under this condition, FAN had an average success rate of 0.76 (average of success rates from 5 sets, each set consisting of 50 trials), the best performance compared to the two other controllers (t -test, $p < 0.001$, $n = 5$ sets). The Control did fairly well

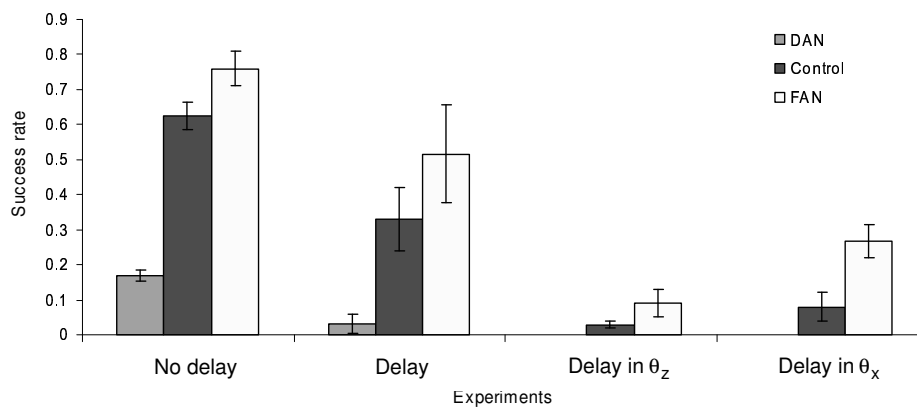


Figure 7. Success rate under different delay conditions.

(success rate 0.62), while DAN showed the lowest success rate (0.17). It is interesting to note that even without delay, FAN showed the best performance. These results establish a benchmark against which the more difficult tasks below can be assessed.

In experiment 2, all sensor inputs were delivered with one step delay (10 ms) in the middle of the evaluation, beginning from 50 iterations and lasting until 150. Note that this is a difficult task because all the inputs are delayed. If the delays were introduced from the beginning or if they lasted longer than 100 iterations, performance in all controllers significantly degraded. For this delay condition, again FAN did the best (t -test, $p < 0.005$, $n = 5$). An important observation at this point is that FAN was more robust than the two other controllers. In case of FAN, the success rate decreased by 32% from that in experiment 1, the base case. However, the Control degraded by 48%, and DAN by 82%. These results indicate that the facilitatory dynamics in FAN is effective in compensating for delay.

In experiment 3 and 4, one step delay in either θ_z or θ_x was introduced throughout each trial (Fig. 7, 3rd and 4th experiments from the left). Note that in these experiments, the delay in these two inputs persisted over the entire trial, unlike in experiment 2. Since all inputs except for one of θ_z or θ_x were received on time, the controllers were able to maintain some balance (for FAN and the Control). However, DAN totally failed in both experiments (thus the results are not reported in Fig 7). As for the successful controllers, FAN significantly outperformed the Control under both conditions (t -test, $p < 0.002$, $n = 5$). An interesting trend in these results is that the delay in θ_z had a more severe effect on the performance than the other input data did. This was somewhat expected, because θ_z is the angle from the vertical axis, and that angle was used to determine whether the pole fell or not (pole is considered down if $\theta_z > 15^\circ$).

Another interesting question is how fast these networks learn to balance the pole. For this we compared the number of generations each of the three controllers took to successfully balance the pole for the first time. For each controller, 250 evolutionary trials (5 sets of 50 trials each) were run where each trial was limited to 70 generations beyond which the controller was treated as failed. The results are summarized in Fig. 8 for experiments 1 to 4. FAN required the least number of generations before successfully balancing the pole in all cases (t -test, $p < 0.0002$, $n = 5$) except for experiment 3 (delay in θ_z) where no difference was found between FAN and the Control ($p = 0.84$). As before, DAN could not learn in the time limit of 70 generations for experiments 3 and 4, thus the results are not reported here.

In summary, facilitatory activation in single neurons significantly improved the ability of cart controllers to compensate for transmission delay within the system. Also, such a facilitatory dynamics allowed for faster learning. The failing performance of DAN is in itself an interesting phenomenon. Decay can be seen as a form of memory, where traces of activity from the past linger on in the present activity, thus we may expect it to be beneficial. However, it turns out that such a form of memory (it can also be seen as a form of smoothing) does not contribute to delay compensation, and actually make things even worse. The results also show that additional computation (as in decaying dynamics) does not always help improve performance. Our experimental results suggest that extrapolation, anticipation, and prediction, which are characteristics of facilitatory dynamics, may be serving an important role in delay compensation.

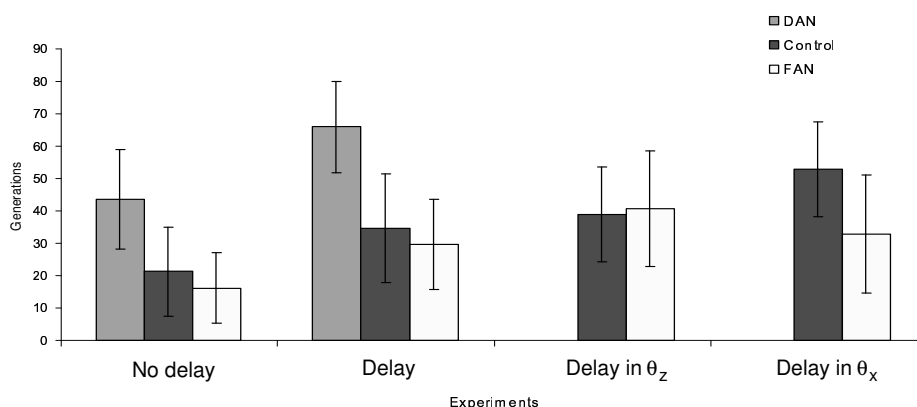


Figure 8. Learning time under different delay conditions.

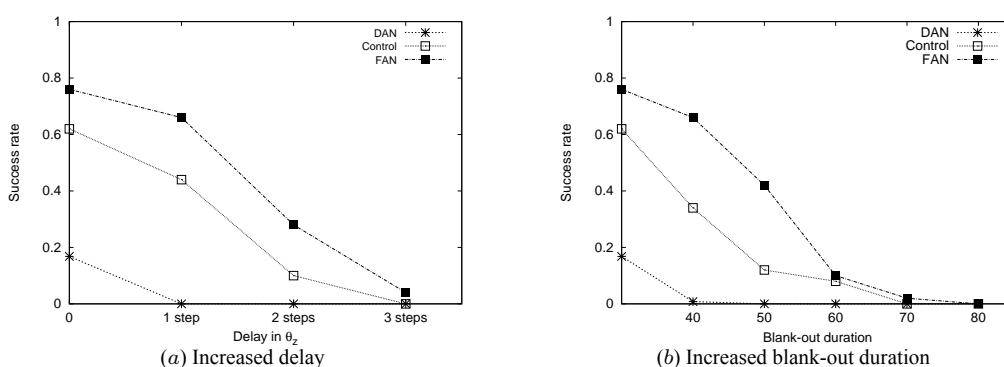


Figure 9. Effect of increased delay and blank-out duration.

4.3 Increased delay and blank-out duration

As a biological organism grows, its size increases and thus the neural processing delay increases as well. How can the nervous system cope with the increase in neural delay? Although certain tasks such as synchronization over a delay line can be achieved via Hebbian learning [50], little is known about how information content can be delivered over a delay line in a timely manner.

To test whether FAN can maintain performance under increasing delay, we increased the delay from 10 ms (1 step) to 30 ms (3 steps) across evolutionary trials. In this experiment, θ_z was delivered with delay, beginning from 50 steps and lasting until 150 steps within each trial. The performance results were compared to those of the Control and of DAN. The results are reported in Fig. 9a. As can be seen here, FAN showed a slower degradation in performance compared to other controllers as delay increased.

Finally, an interesting question is if facilitatory activity can counteract delay in the *external* environment. Suppose a moving object goes behind another object (i.e. occluding event). Until that moving object comes out again, the input may be unavailable. In fact, humans are known to be good at dealing with such a “blank out” of input in the external environment. Mehta and Schaal conducted “virtual pole” experiments where human subjects were asked to balance the pole on the computer screen where the input was blanked out for up to 600 ms at a time [51]. They proposed that internal forward model exists in the central nervous system which can extrapolate the current input into the future state based on the past input (see Sec. 5 for more discussion). It is conceivable that facilitatory dynamics can help in this kind of situation as well. To test if this is the case, we conducted another experiment where input was blanked out for a short period of time, analogous to an occlusion event as sketched above.

We assumed that the neurons would maintain steady-state ring during the blank-out so that the neurons will remain signaling their last-seen state. Thus the input data last seen immediately before the blank-out were fed into the neurons during the blank-out period (blank-out period was introduced once the network reached 500 steps

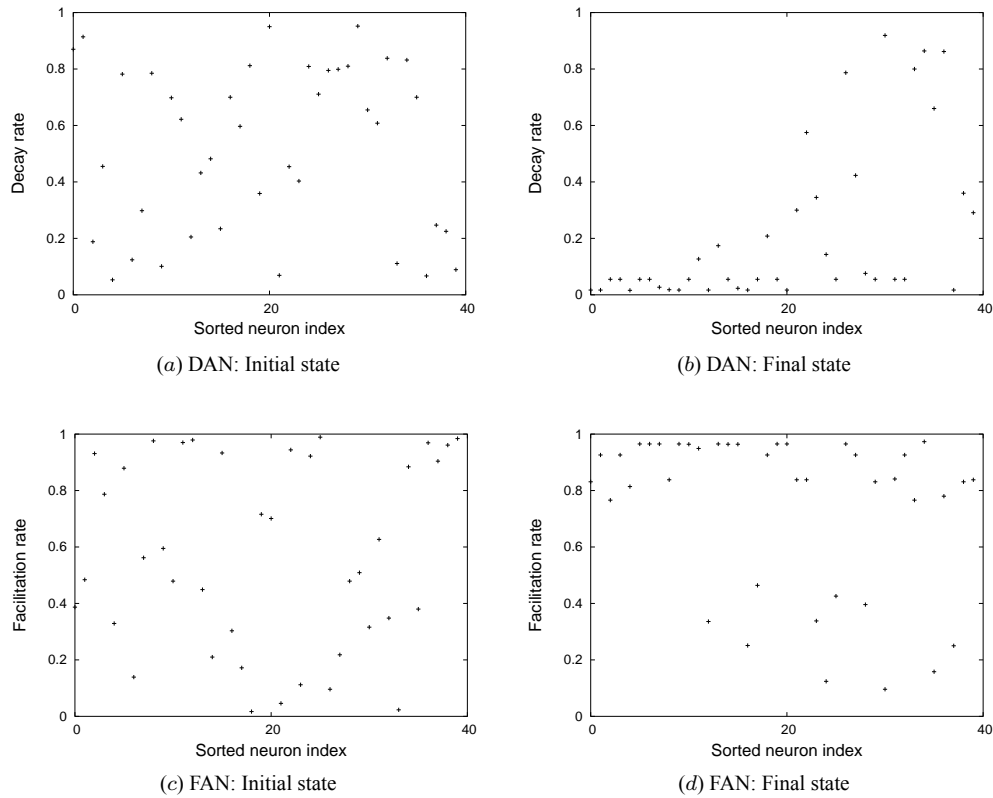


Figure 10. Evolved decay rate and facilitation rate.

of balancing.) As shown in Fig. 9b, FAN showed again higher performance than other controllers. Compared to the Control network, FAN showed slow decrease of performance by 50 steps and this trend is very similar to the sustainable blank-out period observed in humans [51]. In summary, experiments with increasing internal delay and blank-out duration have shown that FAN can effectively deal with increasing delay during growth, and also that the same mechanism can be utilized in dealing with *external* delay just as well.

4.4 Contribution of facilitation/decay rate

The performance results reported in the previous sections suggest that the facilitation rate r_f coded in the gene of the FAN controller serves a useful purpose. To verify if indeed the rate parameters are being utilized, we can look at the evolution of these parameters over generations. Fig. 10 shows an example from a single run that shows evolution of the rate parameters in FAN and DAN. The x -axis represents the index of the neurons in a subpopulation sorted by their fitness values, and the y -axis the decay or facilitation rate. Initially, the rates are uniformly randomly distributed between 0 and 1 (Fig. 10a and c). However, the rates in the final generation look markedly different in FAN vs. DAN. In case of FAN, the top-performing individuals (those on the left) have a high facilitation rate, near 1 (Fig. 10d). This means that extrapolation is pushed to the max (Eq. 2), suggesting that neuroevolution tried to utilize the facilitating dynamics as much as possible. However, for DAN, the top-performers have very low decay rate (near 0, Fig. 10b), suggesting that previous activity is not being utilized at all (Eq. 1). In other words, decay dynamics does not contribute at all in task performance, and neuroevolution tried to minimize its effect by decreasing the decay rate to 0. (Low decay rate means low utilization of historical activation values: see Sec. 2.)

Fig. 11 shows the change in the rate parameter distributions from the initial generation (initial state: the group of bars on the left in Fig. 11a and b) to the final generation (final state: the group of bars on the right in Fig. 11a and b). Fig. 11a shows the distribution of evolved decay rate r_d from successful DAN (3.5 successful trials out of 50 trials on average: 144 neurons = 3.5×40). We can observe that in the initial state, the r_d are uniformly randomly distributed across the subpopulation, but after evolution, the successful networks show an increased number of

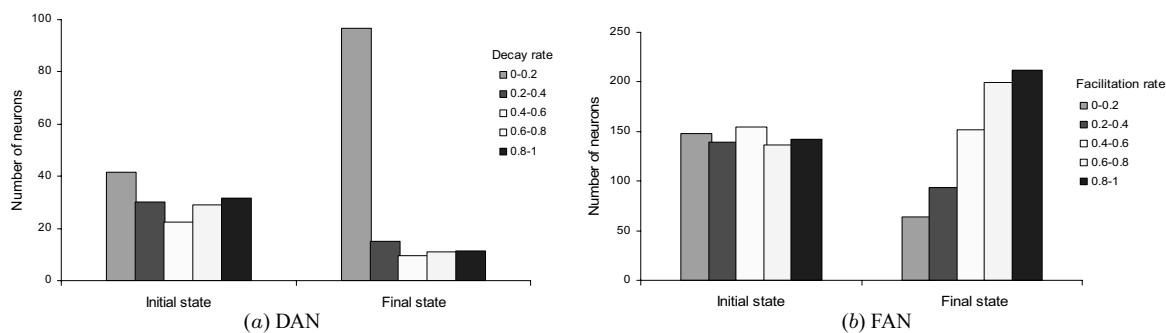


Figure 11. Distribution of evolved decay rate and facilitation rate from DAN and FAN.

neurons that have low r_d . In FAN, on the contrary, the successful networks (18 out of 50 trials on average: 720 neurons = 18×40) show an increased number of neurons that have high r_f in the final state (cf. Fig. 10c, d).

In sum, decaying activity (a form of memory alone) does not contribute in delay compensation, but proper amount of extrapolation based on the memory can significantly help overcome the delay. Thus, facilitatory neural dynamics can be an effective way of keeping an organism's internal state aligned with the environment, in the present. Finally, the adaptability of these dynamics can contribute in dealing with the growth of individual organisms as well as coping with external delay as shown in Fig. 9.

5. Discussion

The main contribution of this paper was to propose a biologically plausible neural mechanism at a single-neuron level for compensating for neural transmission delay. We developed a continuous-valued neuron model utilizing extrapolatory dynamics which was able to perform robustly under internal sensory delay in the 2D pole-balancing task. Because the extrapolation occurs at the cellular level, delay compensation is achieved faster than when it is done at a network level. Our experiments showed that a recurrent neural network with facilitatory single neuron dynamics took less time to learn to solve the delayed pole-balancing problem and showed higher performance than the networks having only recurrent network dynamics. It is possible that the superior performance of FAN is due to a momentum effect. Actually, extrapolation and momentum are closely related. According to Finke et al. [52], the brain has an internal mechanism called representational momentum (analogous to physical momentum) and with it the organism performs mental extrapolation when it perceives moving objects. (See [53, 54, 55, 17] for more about representational momentum.)

In this paper, we used continuous-valued neurons where the neural activity was represented as a single real number. However, biological neurons communicate via spikes, so the biological plausibility of the simulation results above may come under question. What could be a biologically plausible way to implement such a facilitatory dynamics introduced in Eq. 2? One potential instrument is the synaptic dynamics of facilitating synapses found in biological neurons (as we briefly mentioned in Sec. 2). These synapses generate short-term plasticity which shows activity-dependent decrease (depression) or increase (facilitation) in synaptic transmission occurring within several hundred milliseconds from the onset of activity (for reviews see [56, 37, 35]). Especially, facilitating synapses cause augmentation of postsynaptic response through increasing efficacy with successive presynaptic spikes. Preliminary results for this idea can be found in [20]. As we mentioned earlier, the facilitation rate may be an adaptable property of neurons, thus the rate may be adjusted to accommodate different delay durations. That way, the organism with delay in its nervous system can be in touch with the environment in real time.

Another question at this point relates to the extrapolatory capacity of facilitating neural dynamics. Extrapolation is usually related to prediction of the future from information from the present. However, in the nervous system, due to the neural transmission delay, extrapolation may have to be used to predict the present based on past information. Prediction or anticipation of future events is an important characteristic needed in mobile, autonomous agents [57, 58]. Also, as Llinás observed in [59] (p. 3), such projection into the future may be a fundamental property of "mindness". One prominent hypothesis regarding prediction is the internal forward model [60, 61, 51, 12]: Forward models existing in various levels in the nervous system are supposed to produce predictive behaviors which are based on sensory error correction. Internal forward models were suggested from an

engineering point of view, where the sensory motor system is regarded as a well-structured control system that can generate accurate dynamic behaviors. Even though theoretical mechanisms similar to Kalman filter methods were suggested [62, 51, 63], the precise neural basis for the forward models have not been fully investigated. General discussion of the role of prediction in brain function can be found in [64] and [59]. However, they did not relate this predictive ability to the idea of delay compensation. From our perspective, these predictive mechanisms are predicting the present, not predicting the future. An interesting question is, is it possible that neural mechanisms that initially came into use for delay compensation could have developed further to predict future events?

Recently, several brain imaging studies provided supporting evidence for the existence of internal forward models in the nervous system [65, 66, 67, 68]. However, these results did not suggest what could be the neural substrate. Thus, it may be worthwhile investigating how such abilities in autonomous agents may be related to facilitatory dynamics at the cellular level. The input blank-out experiment conducted in Sec. 4.3 is a first step in this direction, where delay compensation mechanisms evolved to deal with internal delay can be directed outward to handle environmental delay and uncertainty.

Time delay has been recognized as one of the main problems in real-time control systems, robotics and teleoperation systems. Various experimental and mathematical methods have been proposed to solve the problem of delay because process delay may cause severe degradation of stability and performance in target systems. On the other hand, neural transmission delay in the biological organisms also has been identified [69] and some researchers have investigated natural delay compensation mechanisms and tried to translate those into mathematical models (for reviews see [70, 60]).

In our research, we focused on the dynamics of single neurons only. In principle, extrapolation can be done at a different level such as the local circuit level or large-scale network level. However, our view is that to compensate for delays existing in various levels in the central nervous system and to achieve faster extrapolation, the compensation mechanism needs to be implemented at the single-neuron level [71].

6. Conclusion

In this paper, we have shown that facilitatory (extrapolatory) dynamics found in facilitating synapses can be used to compensate for delay at a single-neuron level. Experiments with a recurrent neural network controller in a modified 2D pole-balancing problem with sensory delay showed that facilitatory activation greatly helps in coping with delay. The same mechanism was also able to deal with uncertainty in the external environment, as shown in the input blank-out experiment. In summary, it was shown that facilitatory (or extrapolatory) neural activation can effectively deal with delays inside and outside the system, and it can very well be implemented at a single-neuron level, thus allowing a developing nervous system to be in touch with the present.

Acknowledgments

We would like to thank Faustino Gomez and Risto Miikkulainen for making their ESP implementation available to us.

References

- [1] L. G. Nowak and J. Bullier, "The timing of information transfer in the visual system," *Cerebral Cortex*, vol. 12, pp. 205–239, 1997.
- [2] M. T. Schmolesky, Y. Wang, D. P. Hanes, K. G. Thomas, S. Leutgeb, J. D. Schall, and A. D. Leventhal, "Signal timing across the macaque visual system," *The Journal of Neurophysiology*, vol. 79, pp. 3272–3278, 1998.
- [3] S. J. Thorpe and M. Fabre-Thorpe, "Seeking categories in the brain," *Science*, vol. 291, pp. 260–263, 2001.
- [4] D. M. MacKay, "Perceptual stability of a stroboscopically lit visual field containing self-luminous objects," *Nature*, vol. 181, pp. 507–508, 1958.
- [5] R. Nijhawan, "Motion extrapolation in catching," *Nature*, vol. 370, pp. 256–257, 1994.
- [6] M. V. Baldo, "Extrapolation or attention shift?" *Nature*, vol. 378, pp. 565–566, 1995.

- [7] D. Whitney and I. Murakami, “Latency difference, not spatial extrapolation,” *Nature Neuroscience*, vol. 1, pp. 656–657, 1998.
- [8] B. Krekelberg and M. Lappe, “A model of the perceived relative positions of moving objects based upon a slow averaging process,” *Vision Research*, vol. 40, pp. 201–215, 2000.
- [9] Y.-X. Fu, Y. Shen, and Y. Dan, “Motion-induced perceptual extrapolation of blurred visual targets,” *The Journal of Neuroscience*, vol. 21, 2001.
- [10] R. Nijhawan, “Neural delays, visual motion and the flash-lag effect,” *Trends in Cognitive Sciences*, vol. 6, pp. 387–393, 2002.
- [11] W. Erlhagen, “The role of action plans and other cognitive factors in motion extrapolation: A modelling study,” *Visual Cognition*, vol. 11, pp. 315–340, 2004.
- [12] R. Nijhawan and K. Kirschfeld, “Analogous mechanisms compensate for neural delays in the sensory and motor pathways: Evidence from motor flash-lag,” *Current Biology*, vol. 13, pp. 749–753, 2003.
- [13] D. Alais and D. Burr, “The flash-lag effect occurs in audition and cross-modally,” *Current Biology*, vol. 13, pp. 59–63, 2003.
- [14] B. Sheth, R. Nijhawan, and S. Shimojo, “Changing objects lead briefly flashed ones,” *Nature Neuroscience*, vol. 3, pp. 489–495, 2000.
- [15] D. Eagleman and T. J. Sejnowski, “Motion integration and postdiction in visual awareness,” *Science*, vol. 287, pp. 2036–2038, 2000.
- [16] R. P. Rao, D. M. Eagleman, and T. J. Sejnowski, “Optimal smoothing in visual motion perception,” *Neural Computation*, vol. 13, pp. 1243–1253, 2001.
- [17] J. Müsseler, S. Storck, and D. Kerzel, “Comparing mislocalizations with moving stimuli: the fröhlich effect, the flash-lag, and representational momentum,” *Visual Cognition*, vol. 9(1/2), pp. 120–138, 2002.
- [18] D. Kerzel and K. R. Gegenfurtner, “Neuronal processing delays are compensated in the sensorimotor branch of the visual system,” *Current Biology*, vol. 13, pp. 1975–1978, 2003.
- [19] R. Nijhawan, K. Watanabe, B. Khurana, and S. Shimojo, “Compensation of neural delays in visual-motor behaviour: No evidence for shorter afferent delays for visual motion,” *Visual Cognition*, vol. 11(2/3), pp. 275–289, 2004.
- [20] H. Lim and Y. Choe, “Facilitatory neural activity compensating for neural delays as a potential cause of the flash-lag effect,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2005, pp. 268–273.
- [21] J. L. Elman, “Distributed representations, simple recurrent networks, and grammatical structure,” *Machine Learning*, vol. 7, pp. 195–225, 1991.
- [22] —, “Finding structure in time,” *Cognitive Science*, vol. 14, pp. 179–211, 1990.
- [23] F. Gomez and R. Miikkulainen, “2-D pole balancing with recurrent evolutionary networks,” in *Proceeding of the International Conference on Artificial Neural Networks (ICANN)*. Elsevier, 1998, pp. 758–763.
- [24] —, “Solving non-markovian control tasks with neuroevolution,” in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Denver, CO: Morgan Kaufmann, 1999.
- [25] —, “Active guidance for a useless rocket through neuroevolution,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. Chicago, IL: Springer, 2003, pp. 2084–2095.
- [26] D. Wang, “Temporal pattern processing,” in *The Handbook of Brain Theory and Neural Networks (2nd edition)*, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 2003, pp. 1163–1166.
- [27] S. George, A. Dibazar, V. Desai, and T. W. Berger, “Using dynamic synapse based neural networks with wavelet preprocessing for speech applications,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2003, pp. 666–669.
- [28] R. Derakhshani, “Biologically inspired evolutionary temporal neural circuits,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2002, pp. 1357–1361.
- [29] C. W. Eurich, K. Pawelzik, U. Ernst, J. D. Cowan, and J. G. Milton, “Dynamics of self-organized delay adaptation,” *Physical Review Letters*, vol. 82, pp. 1594–1597, 1999.
- [30] A. P. Shon and R. P. Rao, “Learning temporal patterns by redistribution of synaptic efficacy,” *Neurocomputing*, vol. 52-54, pp. 13–18, 2003.

- [31] Y. Choe, "The role of temporal parameters in a thalamocortical model of analogy," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1071–1082, 2004.
- [32] W. Gerstner, "Hebbian learning of pulse timing in the barn owl auditory system," in *Pulsed Neural Networks*, W. Maass and C. M. Bishop, Eds. MIT Press, 1998, ch. 14, pp. 353–377.
- [33] M. Tsodyks, K. Pawelzik, and H. Markram, "Neural networks with dynamic synapses," *Neural Computation*, vol. 10, pp. 821–835, 1998.
- [34] H. Markram, Y. Wang, and M. Tsodyks, "Differential signaling via the same axon of neocortical pyramidal neurons," in *Proceedings of the National Academy of Sciences, USA*, vol. 95, 1998.
- [35] H. Markram, "Elementary principles of nonlinear synaptic transmission," in *Computational Models for Neuroscience: Human Cortical Information Processing*, R. Hecht-Nielsen and T. McKenna, Eds. London, UK: Springer, 2002, ch. 5, pp. 125–169.
- [36] T. Natschläger, W. Maass, and A. Zador, "Efficient temporal processing with biologically realistic dynamic synapses," *Network: Computation in Neural Systems*, vol. 12, pp. 75–87, 2001.
- [37] E. S. Fortune and G. J. Rose, "Short-term synaptic plasticity as a temporal filter," *Trends in Neurosciences*, vol. 24, pp. 381–385, 2001.
- [38] G. Fuhrmann, I. Segev, H. Markram, and M. Tsodyks, "Coding of temporal information by activity-dependent synapses," *Journal of Neurophysiology*, vol. 87, pp. 140–148, 2002.
- [39] G. Silberberg, C. Wu, and H. Markram, "Synaptic dynamics control the timing of neuronal excitation in the activated neocortical microcircuit," *The Journal of Physiology*, vol. 556, pp. 19–27, 2004.
- [40] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. Cambridge, UK: Cambridge University Press, 1992.
- [41] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Systems Magazine*, vol. 9, pp. 31–37, 1989.
- [42] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, pp. 9–44, 1988.
- [43] S. Schaal, "Learning from demonstration," in *The Proceedings of the Advances in Neural Information Processing Systems (NIPS 1997)*, M. C. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 1040–1046.
- [44] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, pp. 219–245, 2000.
- [45] J. Si and Y.-T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Networks*, vol. 12, pp. 264–276, 2001.
- [46] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, pp. 1423–1447, 1999.
- [47] K. O. Stanley and R. Miikkulainen, "Efficient reinforcement learning through evolving neural network topologies," in *Proceedings Genetic and Evolutionary Computation Conference (GECCO)*. IEEE, 2002, pp. 1757–1762.
- [48] D. E. Moriarty, "Efficient reinforcement learning through symbiotic evolution," *Machine Learning*, vol. 22, pp. 11–32, 1997.
- [49] F. Gomez, "Robust non-linear control through neuroevolution," Ph.D. dissertation, Department of Computer Science, The University of Texas at Austin, Austin, TX, 2003, Technical Report AI03-303.
- [50] C. W. Eurich, K. Pawelzik, U. Ernst, A. Thiel, J. D. Cowan, and J. G. Milton, "Delay adaptation in the nervous system," *Neurocomputing*, vol. 32-33, pp. 741–748, 2000.
- [51] B. Mehta and S. Schaal, "Forward models in visuomotor control," *Journal of Neurophysiology*, vol. 88, pp. 942–953, 2002.
- [52] R. A. Finke and G. C. Gary, "Mental extrapolation and representational momentum for complex implied motions," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 14, pp. 112–120, 1988.
- [53] J. J. Freyd and R. A. Finke, "Representational momentum," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, pp. 126–132, 1984.
- [54] J. J. Freyd and J. Q. Johnson, "Probing the time course of representational momentum," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 20, pp. 259–286, 1987.

- [55] M. P. Munger and T. R. Owens, “Representational momentum and the ash-lag effect,” *Visual Cognition*, vol. 11, pp. 81–103, 2004.
- [56] J. Liaw and T. W. Berger, “Dynamic synapse: Harnessing the computing power of synaptic dynamics,” *Neurocomputing*, vol. 26-27, pp. 199–206, 1999.
- [57] R. Möller, “Perception through anticipation: An approach to behavior-based perception,” in *Proceedings of New Trends in Cognitive Science*, 1997, pp. 184–190.
- [58] H.-M. Gross, A. Heinze, T. Seiler, and V. Stephan, “Generative character of perception: A neural architecture for sensorimotor anticipation,” *Neural Networks*, vol. 12, pp. 1101–1129, 1999.
- [59] R. R. Llinás, *I of the Vortex*. Cambridge, MA: MIT Press, 2001.
- [60] D. M. Wolpert and J. R. Flanagan, “Motor prediction,” *Current Biology*, vol. 11(18), pp. R729–R732, 2001.
- [61] R. C. Miall and D. M. Wolpert, “Forward models for physiological motor control,” *Neural Networks*, vol. 9, pp. 1265–1285, 1996.
- [62] D. M. Wolpert, “Computational approaches to motor control,” *Trends in Cognitive Sciences*, vol. 1, pp. 209–216, 1997.
- [63] E. Oztopa, D. Wolpert, and M. Kawato, “Mental state inference using visual control parameters,” *Cognitive Brain Research*, vol. 22, pp. 129–151, 2005.
- [64] J. Hawkins and S. Blakeslee, *On Intelligence*, 1st ed. Hency Holt and Company, New York: Times Books, 2004.
- [65] S. J. Blackmore, D. M. Wolpert, and C. D. Frith, “Central cancellation of self-produced tickle sensation,” *Nature neuroscience*, vol. 1, pp. 635–640, 1998.
- [66] M. Desmurget and S. Grafton, “Forward modeling allows feedback control for fast reaching movements,” *Trends in Cognitive Sciences*, vol. 4, pp. 423–431, 2000.
- [67] M. R. Mehta, “Neural dynamics of predictive coding,” *The Neuroscientist*, vol. 7, pp. 490–495, 2001.
- [68] B. Webb, “Neural mechanisms for prediction: do insects have forward models?” *Trends in Neurosciences*, vol. 27, pp. 278–282, 2004.
- [69] A. Pellionisz and R. Llinas, “Brain modeling by tensor network theory and computer simulation, the cerebellum: Distributed processor for predictive coordination,” *Neuroscience*, vol. 4, pp. 328–348, 1979.
- [70] A. Kataria, H. Özbay, and H. Hemani, “Controller design for natural and robotic systems with transmission delays,” *Journal of Robotic Systems*, vol. 19, pp. 231–244, 2002.
- [71] H. Lim and Y. Choe, “Delay compensation through facilitating synapses and stdp: A neural basis for orientation ash-lag effect,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2006, Accepted.